10th Workshop on Current Trends in Cryptology (CTCrypt 2021)

June 1-4, 2021, Dorokhovo, Ruza District,
Moscow Region, Russia

Pre-proceedings

In cooperation

CTCrypt 2021 is organized by

– Academy of Cryptography of the Russian Federation

– Steklov Mathematical Institute of Russian Academy of Science

– Technical Committee for Standardization «Cryptography and security mechanisms» (TC 026)

Steering Committee

Co-chairs

Aleksandr Shoitov	–	Academy of Cryptography of the Russian Federation, Russia
Vladimir Sachkov	–	Academy of Cryptography of the Russian Federation, Russia
Igor Kachalin	–	TC 026, Russia

Steering Committee Members

Andrey Zubkov	–	Steklov Mathematical Institute of RAS, Russia
Dmitry Matyukhin	–	TC 026, Russia
Andrey Pichkur	–	Federal Educational and Methodical Association in System of Higher Education on Information Security, Russia

Program Committee

Co-chairs

Alexander Lapshin	–	Academy of Cryptography of the Russian Federation, Russia
Dmitry Matyukhin	–	TC 026, Russia
Andrey Zubkov	–	Steklov Mathematical Institute of RAS, Russia

Program Committee Members

Sergey Agievich	–	Research Institute for Applied Problems of Mathematics and Informatics, Belarus
Sergey Aleshnikov	–	Immanuel Kant Baltic Federal University, Russia
Alexey Alexandrov	–	Vladimir State University named after Alexander and Nikolay Stoletovs, Russia
Sergey Checheta	–	Federal Educational and Methodical Association in System of Higher Education on Information Security, Russia
Ivan Chizhov	–	Lomonosov Moscow State University, Russia
Vladimir Fomichev	–	«Security Code», LLC, Russia
Yury Kharin	–	Research Institute for Applied Problems of Mathematics and Informatics, Belarus
Grigory Marshalko	–	TC 026, Russia
Mridul Nandi	–	Indian Statistical Institute, India
Eduard Primenko	–	Lomonosov Moscow State University, Russia
Boris Ryabko	–	Institute of Computational Technologies SB RAS; and Novosibirsk State University, Russia
Markku-Juhani Olavi Saarinen	–	PQShield Ltd., Finland/UK
Igor Semaev	–	The University of Bergen, Norway
Vasily Shishkin	–	"NPK Kryptonite", JSC, Russia
Stanislav Smyshlyaev	–	"Crypto-Pro", LLC, Russia
Alexey Tarasov	–	Federal Educational and Methodical Association in System of Higher Education on Information Security, Russia
Andrey Trishin	–	"Certification Research Center", LLC, Russia
Alexey Urivskiy	–	"InfoTeCS", JSC, Russia
Amr Youssef	–	Concordia University, Canada
Andrey Zyazin	–	Russian Technological University (MIREA), Russia

External Reviewers

Damian Straszak, Denis Fomin, Dmitriy Trifonov, Ernesto Dominguez Fiallo, Jean-Christophe Deneuville, Luca De Feo, Maxim Nikolaev, Nikolay Shenets, Rinat Shakirov, Sergey Grebnev, Vladislav Nozdrunov.

Invited Talks

French-Russian Scientific Cooperation in Cryptography and Information Security

Vladimir Fomichev[1,2,3] and Alisa Koreneva[2]

[1]Financial University under the Government of the Russian Federation,
[2]«Security Code», LLC,
[3]Federal Research Center "Informatics and Management" of the Russian Academy of Sciences
fomichev.2016@yandex.ru, a.koreneva@securitycode.ru

Abstract

In this talk we would like to share the experience we have gained from the collaboration with an independent French academic and technical Journal of Computer Virology and Hacking Techniques (JICV) initiated by its Editor-in-Chief, Professor Eric Filiol. The mission of this joint project is a contribution to raising awareness of the Russian research activity. This task is staple, as the publications of Russian scientists are unfortunately not sufficiently known in the Western world, due to the historically low attention to the Russian language and electronic resources of Russian journals. We contributed to the collaboration as guest editors of the journal special issue titled Russian Research in Cryptology and Information Security Systems. This issue covered state-of-the-art works of Russian researchers on fundamental problems and applications of cryptography and information security and was successfully published in December 2020, providing two articles from the editors, one invited paper and eight selected articles. We give an outline of the papers and highlight its scientific value.

Keywords: Scientific Cooperation, cryptography, information Security.

You Only Speak Once: Private Computing on Public Blockchains

Hugo Krawczyk

IBM Research, USA
hugokraw@gmail.com

Abstract

Blockchains are well-known for their consensus and integrity properties but secrecy is hard to impose, let alone general secure privacy-preserving computation. In this talk I will introduce a notion called "You Only Speak Once" (YOSO) and show how it leads to scalable secure (multi-party) computation over blockchains. In the YOSO model of computation, a small subset of parties (physical machines) are periodically assigned ephemeral roles that require the machine to send a single message after which the machine erases all its state. Thus, an attacker, that is limited on the number of machines it can control at any given time, cannot know which machines/roles to attack till they speak; but then it is too late to learn useful information from their compromise. This model can be realized in blockchains where it is unpredictable who the proposer of the next block is, such as in bitcoin, Algorand and others.

Keywords: blockchain, multi-party computation, YOSO.

Towards post-quantum cryptographic standards, focus on code-based cryptography

Jean-Christophe Deneuville

French Civil Aviation University, France
jean-christophe.deneuville@enac.fr

Abstract

Late 2017, the National Institute for Standards and Technologies (NIST) initiated a process to standardize quantum safe cryptographic primitives: public-key encryption, key-exchange and digital signature schemes. The process is currently in the 3rd round, and several candidates (among the finalists) should be selected for standardization at the end of this round, some others (among the alternates) should be selected after another round.

In this talk, I will give an overview of the finalists and alternate candidates, with a focus on code-based proposals. To do so, I will recall some fundamentals of code-based cryptography, present historical constructions that have inspired recent designs, and provide elements to understand why code-based cryptography stands as a mature possible replacement for encryption.

I will also compare the code-based finalist (Classic McEliece) with the alternate candidates (BIKE and HQC) to explain why a 4th round makes sense for a wide adoption of code-based encryption.

Finally, I will conclude the talk with challenges and open questions code-based cryptography faces.

Keywords: quantum safe cryptographic primitives, code-based cryptography.

Contents

Symmetric Cryptography

Modes of Operations

Some Properties of One Mode of Operation of Block Ciphers **12**
Dmitriy Bogdanov and Vladislav Nozdrunov

Misuse-resistant MGM2 Mode **18**
Liliya Akhmetzyanova, Evgeny Alekseev, Alexandra Babueva, Andrey Bozhko, and Stanislav Smyshlyaev

IQRA: Incremental Quadratic Re-keying friendly Authentication scheme **35**
Liliya Akhmetzyanova, Evgeny Alekseev, Alexandra Babueva, Lidiia Nikiforova, and Stanislav Smyshlyaev

The Re-keying Mechanism COM-CTR+D **64**
Daymé Almeida, Alejandro Freyre, and Adrián Alfonso

Format-Preserving Encryption: a Survey **78**
Kirill Tsaregorodtsev

Permutations and Substitutions

On Differential Uniformity of Permutations Derived Using a Generalized Construction **97**
Denis Fomin and Maria Kovrizhnykh

On the Generation of Cryptographically Strong Substitution Boxes from Small Ones and Heuristic Search **112**
Alejandro Freyre-Echevarría

Constructing Involutions Specifying Their Coordinate Functions **129**
Daniel Humberto Hernández Piloto and Oliver Coy Puente

Analysis

On the Impossibility of an Invariant Attack on Kuznyechik 150
Denis Fomin

Algebraic Cryptanalysis of Round-reduced Lightweight Ciphers Simon and Speck 162
Aleksandr Kutsenko, Natalia Atutova, Darya Zyubina, Ekaterina Maro, and Stepan Filippov

Algebraic and Probabilistic Aspects

Two Variants of Lempel-Ziv Criterion and Their Reasoning 183
Vladimir Mikhailov and Vasilii Kruglov

Streebog Compression Function as PRF in Secret-key Settings 194
Vitaly Kiryukhin

Nonlinearity of Bent Functions over Finite Fields 210
Vladimir Ryabov

On Some Properties of the Curvature and Nondegeneracy of Boolean Functions 220
Reynier Antonio de la Cruz Jiménez

On the Properties of Some Sequences Generated by Shift Registers and Latin Squares 250
Ramses Rodriguez Aulet and Adrián Alfonso Peñate

On Derivatives of Boolean Bent Functions 262
Alexander Shaporenko

The Duality Mapping and Unitary Operators Acting on the Set of All Generalized Boolean Functions 274
Aleksandr Kutsenko and Anastasiya Gorodilova

Postquantum Cryptography

Some Remarks on the Security of Isogeny-based Cryptosystems 303
Sergey Grebnev

The Hadamard Square of Concatenated Linear Codes 313
Ivan Chizhov and Alexandra Davletshina

Public key cryptography

Small Scalar Multiplication on Weierstrass Curves using Division Polynomials **329**
Sergey Agievich, Stanislav Poruchnik, and Vladislav Semenov

One "Short" Signature Scheme's Security Properties **345**
Anton Guselev

Symmetric Cryptography
Modes of Operations

Some properties of one mode of operation of block ciphers

Dmitriy Bogdanov[1] and Vladislav Nozdrunov[2]

[1]National Research Nuclear University MEPhI (Moscow Engineering Physics Institute), Russia
[2]Technical Committee on Standardisation "Cryptographic Information Security" (TC 026), Russia
bogdanov_ds@tc26, nozdrunov_vi@tc26.ru

Abstract

As part of the work of the Technical Committee for Standardization "Cryptography and Security Mechanisms" (TC 026), a draft document [6] was presented in 2019 describing the mode of operation for full disk encryption (hereinafter referred to as the DEC regime).

This mode is a modification of the CTR [4] mode, made taking into account the operating conditions of block-oriented data carriers and some of their features. In this paper, the cryptographic characteristics of the DEC mode, such as the limit on the number of partition keys generated and the probability of collision the keystream used for sector encryption, are investigated.

Keywords: full disk encryption, modes of operation, cryptographic protocols, CTR, KDF.

1 Introduction

As part of the work of the Technical Committee on Standardisation «Cryptographic Information Security» (TC26), a draft document [6] was submitted in 2019 describing a block cipher mode designed to ensure data confidentiality on block-oriented data carriers (hereinafter – DEC mode). This mode is a modification of the CTR [4] mode, made with respect to the operating conditions of block-oriented data carriers and some of their peculiarities.

Most modern storage medium read and write in whole sectors – bit strings of fixed length. Regardless of whether a whole sector or a fraction of it is occupied by "usable" information, the whole sector will be read or overwritten. Accordingly, it is assumed here and below that no empty or incomplete sectors can exist.

In DEC mode operation, it is assumed that the medium is represented as w consecutive partitions, where a partition is the set of s sectors, $w, s \in \mathbb{N}$. DEC mode uses gamification to encrypt the plaintext on the storage medium. The DEC mode itself describes the procedure for generating the initialization values $CTR(i, l_{j,i}, t)$ and the encryption keys $K_{j,i,l_{j,i}}$.

2 Brief description of DEC mode

Initialisation values are generated according to the rule:

$$CTR(i, l_{j,i}, t) = i||(l_{j,i} \cdot q) \boxplus_{n/2} t,$$

j – partition number, i – sector number in partition, $l_{j,i}$ – count of number of encryptions of i-th sector of j-th partition, n – block length of block cipher used, q – sector size in blocks[1], $\boxplus_{n/2}, \cdot$ – addition and multiplication in ring $\mathbb{Z}_{2^{\frac{n}{2}}}$, $t \in \{0, 1, \cdots, q-1\}$, e_K – mapping implementing encryption[2] using the key K and defined in [4].

Note that the DEC mode ensures that there is no overlap of $CTR(i, l_{j,i}, t)$ values when the counter values $l_{j,i}$ differ by less than $\frac{2^n}{q}$.

The keys $K_{j,i,l_{j,i}}$ (hereafter referred to as sector keys) are generated using the derived key function KDF described in [5] from the partition keys. Partition keys are generated with the KDF derivation function and the secret master key. A complete description of the procedure for generating sector keys and partition keys is beyond the scope of this paper.

Keystream blocks are generated according to the rule:

$$\Delta_t = e_{K_{j,i,l_{j,i}}}(CTR(i, l_{j,i}, t)). \tag{1}$$

To encrypt the plaintext $X_0, \cdots, X_{q-1}$ on sector number i of partition number j, the counter value $l_{j,i}$ is incriminated and keystream blocks are generated by rule 1. The ciphertext $C_0, \cdots, C_{q-1}$ is written to the sector, where $C_i = X_i \oplus \Delta_i$, $i = \{0, \cdots, q-1\}$.

3 On limiting the number of sector keys

The cryptographic properties of the KDF function are discussed in detail in work [1], in particular, the statistical properties of the key sequences produced are studied. In DEC mode, the following parameters are used in terms of work [1, 5]: f – the pseudorandom function – CMAC. L – length of output – 256, d – length of input[3] – 1536, n – function output length f – block length of the block cipher used, $\beta = \lceil \frac{L}{n} \rceil$ the intermediate key generation step follows a simplified procedure.

Then for the advantages [4] of the adversary, whose computational capability is limited by the value t, solving the problem of distinguishing q keys

[1]In most modern storage medium, bit length of sector $l \in \{4096, 32768\}$. Thus, $q = \frac{l}{n} \in \mathbb{N}$.

[2]Implying that value $CTR(i, l_{j,i}, t)$ is translated into a bit string of length n

[3]Length of output function *«format»*.

[4]Models $Adv_f^{prf^*}$ and Adv_f^{prf} are given in [1].

generated during DEC mode from random function values, can be estimated as follows:

$$Adv^{prf^*}_{kdf^2}(t,q) \le Adv^{prf}_{f}(t,\beta q) + \frac{\beta q(\beta q-1)}{2^d}. \tag{2}$$

When using the block cipher «Kuznechik», the estimate (2) will take the form:

$$\begin{aligned} Adv^{prf^*}_{kdf^2}(t,q) &\le Adv^{prf}_{CMAC}(t,2q) + \frac{2q(2q-1)}{2^{1536}} \le \\ &\le \frac{2884q^2}{2^{128}} + \frac{t'}{2^{256}} + \frac{24q+1}{2^{128}} + \frac{2q(2q-1)}{2^{1536}}, \end{aligned} \tag{3}$$

where $t' = t + O(24q)$. The estimate (3) can be used to determine limits on the number of sector keys generated from a single partition key based on the allowed values of adversary dominance. For example, with $t \le 2^{128}$, $q \le 2^{51}$ the adversary advantages will be less than 10^{-3}.

If the block cipher «Magma» is used, the estimate (2) will take the form

$$\begin{aligned} Adv^{prf^*}_{kdf^2}(t,q) &\le Adv^{prf}_{CMAC}(t,4q) + \frac{4q(4q-1)}{2^{1536}} \le \\ &\le \frac{46096q^2}{2^{64}} + \frac{t'}{2^{192}} + \frac{96q+1}{2^{64}} + \frac{4q(4q-1)}{2^{1536}}, \end{aligned} \tag{4}$$

where $t' = t + O(96q)$. The estimate (4) can be used to determine limits on the number of sector keys generated from a single partition key based on the allowed values of adversary dominance. For example, with $t \le 2^{128}$, $q \le 2^{17}$ the adversary advantages will be less than 10^{-3}.

For a typical 1TB consumer SSD drive, the write/rewrite endurance is about 1200TB or 2^{54} bits. Most storage medium have a sector size of either 4096 bits or 32768 bits. Thus, when using DEC mode with the block cipher «Kuznechik», a single partition key is enough to produce sector keys[5] for the whole life of the medium.

4 On the probability of a collision of keystream

Due to the nature of the DEC mode, a complete collision of the keystreams used to encrypt the entire sector will result in recovering a bitwise sum of plaintetxts. This section provides an estimate from above of the probability of this event.

For the purposes of this section, it is assumed that all sector keys are realizations of independent random variables, that having a uniform distribution on V_{256}.

[5]Such that the advantages of an adversary would be less than 10^{-3}.

x_1	x_2	x_3	$\cdots$	x_N
$E_{\xi_{1,1}}(x_1)$	$E_{\xi_{1,2}}(x_2)$	$E_{\xi_{1,3}}(x_3)$	$\cdots$	$E_{\xi_{1,N}}(x_N)$
$E_{\xi_{2,1}}(x_1)$	$E_{\xi_{2,2}}(x_2)$	$E_{\xi_{2,3}}(x_3)$	$\cdots$	$E_{\xi_{2,N}}(x_N)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$E_{\xi_{M,1}}(x_1)$	$E_{\xi_{M,2}}(x_2)$	$E_{\xi_{M,3}}(x_3)$	$\cdots$	$E_{\xi_{M,N}}(x_N)$

Table 1:

4.1 Mathematical model

Let $M, N \in \mathbb{N}$, $x_1, \cdots, x_N \in \mathcal{X}$, where $\mathcal{X}$ – some set, $x_i \neq x_j$ at $i \neq j$. Denote by $\overline{x}$ the set $\{x_1, \cdots, x_N\}$. Let among all injective mappings $E : \overline{x} \to \mathcal{X}$ a randomly chosen ordered set of K (not necessarily distinct) functions. Let us denote the functions from this set by $E_1, \cdots, E_K$. Let $\xi_{i,j}$, $i \in \{1, \cdots, M\}$, $j \in \{1, \cdots, N\}$ be independent random variables having a uniform distribution on $\{1, \cdots, K\}$. We need estimate the probability of event A such that there are $i, i' \in \{1, \cdots, M\}$, $j, j' \in \{1, \cdots, N\}$, $(i, j) \neq (i', j')$ such that $E_{\xi_{i,j}}(j) = E_{\xi_{i',j'}}(j')$. Such sets of pairs (i, j), (i', j') will be further called «collision».

4.2 Relationship between the model and the problem

By virtue of its construction, the sets $\{CTR(i, l_{j,i}, t),\ t = 0, \cdots, q-1\}$ at different $i, l_{j,i}$ either do not intersect or coincide. Denote the set of all different sets $\{CTR(i, l_{j,i}, t),\ t = 0, \cdots, q-1\}$ by $\overline{x}$. By N we denote the power of the set $\overline{x}$. Thus, given a fixed key $K_{j,i,l_{j,i}}$, the keystream values produced during DEC mode can be thought of as an image of some function $E_{K_{j,i,l_{j,i}}} : \overline{x} \to V_{q \cdot n}$. Note that since e_K mappings are permutations, the functions $E_{K_{k,i,l_{j,i}}}$ are injective. Based on the operational and technical characteristics of the medium protected by the DEC mode, the number M – the maximum possible number of uses of the set $\{CTR(i, l_{j,i}, t),\ t = 0, \cdots, q-1\}$ to produce the gamut can be determined.

Thus, the problem to be solved can be reduced to the model under consideration when $\mathcal{X} = V_{q \cdot n}$, $K = 2^{256}$ and $\overline{x}, N, M$ introduced above.

4.3 Evaluation probability of collision

For clarity, the values of $E_{\xi_{i,j}}(j)$ are given in table 1. Let us introduce the events $A^{k,l}$, $k \leq l$, $k, l \in \{1, \cdots, N\}$ such that $E_{\xi_{i,k}}(x_k) = E_{\xi_{i',l}}(x_l)$ exist. These events correspond to the fact that the collision occurred between the

element of k-th and l-th columns of the table 1. Then

$$A = \bigcup_{k \le l}^{N} A^{k,l}, \text{ and } Pr[A] \le \sum_{k \le l}^{N} Pr[A^{k,l}].$$

Let us introduce the events $A^{k,l}_{i,i'}$ consisting of $E_{\xi_{i,k}}(x_k) = E_{\xi_{i',l}}(x_l)$. Then

$$A^{k,k} = \bigcup_{i < i'}^{N} A^{k,k}_{i,i'}, \text{ and } Pr[A^{k,k}] \le \sum_{i < i'}^{N} Pr[A^{k,k}_{i,i'}]$$

$$A^{k,l} = \bigcup_{i,i'}^{N} A^{k,l}_{i,i'}, \text{ and } Pr[A^{k,l}] \le \sum_{i,i'}^{N} Pr[A^{k,l}_{i,i'}], k \ne l.$$

Using the law of total probability

$$Pr[A^{k,k}_{i,i'}] = Pr[A^{k,k}_{i,i'} | \xi_{i,k} = \xi_{i',k}] \cdot Pr[\xi_{i,k} = \xi_{i',k}] + Pr[A^{k,k}_{i,i'} | \xi_{i,k} \ne \xi_{i',k}] \cdot Pr[\xi_{i,k} \ne \xi_{i',k}].$$

Note that by virtue of the construction of $Pr[A^{k,k}_{i,i'} | \xi_{i,k} = \xi_{i',k}] = 1$, $Pr[A^{k,k}_{i,i'} | \xi_{i,k} \ne \xi_{i',k}] = \frac{1}{|Q|}$. Then $Pr[A^{k,k}_{i,i'}] = \frac{1}{K} + \frac{K-1}{|Q| \cdot K}$.

Using the law of total probability

$$Pr[A^{k,l}_{i,i'}] = Pr[A^{k,l}_{i,i'} | \xi_{i,k} = \xi_{i',l}] \cdot Pr[\xi_{i,k} = \xi_{i',l}] + Pr[A^{k,l}_{i,i'} | \xi_{i,k} \ne \xi_{i',l}] \cdot Pr[\xi_{i,k} \ne \xi_{i',l}].$$

Note that, due to the injectivity of the E functions, the $Pr[A^{k,l}_{i,i'} | \xi_{i,k} = \xi_{i',l}] = 0$, $Pr[A^{k,l}_{i,i'} | \xi_{i,k} \ne \xi_{i',l}] = \frac{1}{|Q|}$. Тогда $Pr[A^{k,k}_{i,i'}] = \frac{K-1}{|Q| \cdot K}$.

Thus,

$$\begin{aligned} Pr[A] \le & \frac{NM(M-1)}{2K} + \frac{NM(M-1)(K-1)}{2|Q| \cdot K} + \frac{N(N-1)M^2(K-1)}{2|Q| \cdot K} = \\ & = \frac{NM(M-1)}{2K} + \frac{NM(K-1)(NM-1)}{2|Q| \cdot K}. \end{aligned} \tag{5}$$

4.4 The consequence from an evaluation on the probability of collision

It is of particular interest to study the behaviour of the estimate (5) when the parameters N, M are changed for a fixed value of NM. The value of NM corresponds to the total number of write operations on a DEC-protected medium sector. The parameter N is directly related to the number of sectors in the partition.

We will investigate the estimation (5) using the Sturm [2] method. Let $NM = const = S$ be fixed. Let $\Delta > 1$ – some number. Let $N' = \frac{N}{\Delta}$, $M' = \Delta \cdot M$. Then the estimate (5) for the medium with parameters N', M' is

$$\frac{N'M'(M'-1)}{2K} + \frac{N'M'(K-1)(N'M'-1)}{2|Q| \cdot K} =$$
$$\frac{NM(\Delta \cdot M-1)}{2K} + \frac{NM(K-1)(NM-1)}{2|Q| \cdot K} >$$
$$\frac{NM(M-1)}{2K} + \frac{NM(K-1)(NM-1)}{2|Q| \cdot K}.$$

Thus, for a fixed value of NM, increasing the parameter M entails increasing the value of the estimate on the collision probability.

In terms of DEC mode, the consequence means that when representing a medium as one partition with $2^{\frac{n}{2}}$ sectors, the probability of complete collision of keystreams will be less [6] than when the same medium is represented as $2^{\frac{n}{4}}$ partitions with $2^{\frac{n}{4}}$ sectors.

References

[1] "Results of cryptographic research and reasoning for cryptographic qualities. Mechanisms of derivative key generation", 2017, In Russian.

[2] Gorelov M.A., "Simple optimization problems. Non-algebraic transformations", 2012, In Russian.

[3] "GOST P 34.12–2015. Information technology. Cryptographic protection of information. Block ciphers", 2015, In Russian.

[4] "GOST P 34.13–2015. Information technology. Cryptographic protection of information. Modes of operation of block ciphers", 2015, In Russian.

[5] "Recommendations for standardisation P 1323565.1.022-2018. Information technology. Cryptographic protection of information. Key derivation function", 2018, In Russian.

[6] "Block ciphers mode of operation designed to protect of data storage medium with a block-oriented structure (draft)", 2019, In Russian.

[6] Under the same operating conditions.

Misuse-resistant MGM2 mode

Liliya Akhmetzyanova, Evgeny Alekseev, Alexandra Babueva,
Andrey Bozhko, and Stanislav Smyshlyaev

CryptoPro LLC, Russia
{lah, alekseev, babueva, bozhko, svs}@cryptopro.ru

Abstract

We introduce a slight modification of the standard AEAD MGM mode – an MGM2 mode, for which a nonce is not encrypted anymore before using it as an initial counter value. For the new mode we provide security bounds regarding security notions in the nonce-misuse setting (MRAE-integrity and CPA-resilience). The obtained bounds are even better than the bounds obtained for the original MGM mode regarding standard security notions.

Keywords: MGM, AEAD mode, security notion, security bounds, nonce-misuse, misuse-resistant

1 Introduction

Authenticated Encryption with Associated Data (AEAD) schemes, which aim at providing both integrity and confidentiality of data, are recently considered to be among the most widely used cryptographic schemes. Therefore, the security of such schemes is crucial. Security analysis of AEAD-schemes is usually carried out in the provable security paradigm regarding standard notions, introduced in [6], they are IND-CCA and IND-CPA for confidentiality and INT-CTXT for integrity.

One of the examples of such schemes is an MGM block cipher mode of operation, that was adopted in Russia as a standard AEAD-mode [13]. The MGM plaintext encryption procedure is quite similar to encryption in the CTR2 [18] mode. The main element of the MGM authentication procedure is a multilinear function with secret coefficients produced in the same way as the secret masking blocks used for plaintext encryption. Integrity and confidentiality of MGM were analysed in [1] regarding standard security notions. Since MGM was not developed with provable security in mind, security proofs turned out to be cumbersome and, hence, difficult to verify.

Even though analysis of AEAD-schemes in the standard models is mandatory and enough for use in many applications, some environments require other unusual cryptographic properties, e.g. leakage resilience [5], RUP («Release of Unverified Plaintext») security [3], KDM («Key Dependent Message») security [10], misuse-resistance [19], etc. In the current paper we focus on misuse-resistance or nonce misuse property [19]. A nonce (number used only once) is an input to encryption or decryption algorithms of AEAD-schemes that has to be unique (within a fixed key), but in some applications such requirement is hard to obtain, not to mention implementation faults. Misuse-resistant schemes aim to ensure the best possible security when faulty nonce is provided.

Security notions for misuse-resistant authenticated encryption were originally proposed by Rogaway and Shrimpton in [19] and further developed in [4]. Strong variant of misuse-resistant notions, called MRAE («Misuse-Resistant AE»), was introduced in [19]. This notion is the extensions of the IND-CCA and INT-CTXT notions by allowing an adversary to repeat nonces in all of its' queries. The MRAE notion is similar to a DAE notion [19] («Deterministic AE») where confidentiality is formalised as follows: ciphertext of each *new* query (not only new nonce) has to be indistinguishable from a random string. Providing such confidentiality is rather strong, and trying to achieve it seems to lead to loss in performance. All MRAE-secure modes, known to the authors, demand sufficiently larger amount of block cipher calls [16] or lose online property [15, 22]. For the reasons above, weak notions for confidentiality called CPA-res and CCA-res («Chosen Plaintext/Ciphertext Attack-resilience») were introduced in [4], where the confidentiality should be achieved only for messages that were encrypted correctly using unique nonces.

In nonce-misuse setting the MGM mode is obviously insecure in the MRAE model regarding confidentiality since counter-based encryption is actually used. MRAE-integrity of the MGM mode was analysed in [17]: the birthday type attack was proposed. However, no lower bounds for MGM were proven. So, there is a «hope» to provide non-trivial security bounds for MGM in the MRAE-integrity and CCA-res models.

Motivating by expectation that the security proof for the MGM mode in non-standard models will be even more complex, than in standard ones, we introduce modification of MGM mode – MGM2. The main difference between two modes lies in the way how secret masking blocks and secret coefficients of the multilinear function are produced – for the MGM2 mode this process is carried out in the CTR [18] style (without preliminary nonce encryption).

Note, that the main cryptographic core of the construction, namely multilinear function, is not changed. We provide the security bounds for MGM2 in the MRAE-integrity and CPA-res models that turned out to be even better, than the bounds for the original MGM mode in the standard models. The corresponding security proofs are relatively short and, we hope, easier to verify. Among other advantages, the design of the MGM2 mode also allows to transparently integrate internal re-keying without a master key [2] (in the same way as for CTR-ACPKM [14] done) to achive new security properties like leakage-resilience and increase key lifetime.

2 Preliminaries

By $\{0,1\}^s$ we denote the set of s-component bit strings and by $\{0,1\}^*$ we denote the set of all bit strings of finite length including the empty string. Let $|a|$ be the bit length of the string $a \in \{0,1\}^*$. For a bit string a we denote by $|a|_n = \lceil |a|/n \rceil$ the length of the string a in n-bit blocks. By $\{0,1\}^{\leqslant s}$ we denote the set of bit strings which length is less or equal to s.

For a string $a \in \{0,1\}^*$ and a positive integer $l \leqslant |a|$ let $\mathrm{msb}_\ell(a)$ be the string, consisting of the leftmost l bits of a. For nonnegative integers l and i let $\mathrm{str}_l(i)$ be l-bit representation of i with the least significant bit on the right. For bit strings $a \in \{0,1\}^n$ and $b \in \{0,1\}^n$ we denote by $a \otimes b$ a string which is the result of their multiplication in $GF(2^n)$ (here strings encode polynomials in the standard way). If the value s is chosen from a set S uniformly at random, then we denote $s \xleftarrow{\mathcal{U}} S$. We define a function $\mathsf{Set1}_r\colon \{0,1\}^n \to \{0,1\}^n$, $\mathsf{Set1}_r(x) = x$ or $(\overbrace{0\ldots0}^{r}1\overbrace{0\ldots0}^{n-r-1})$, $0 \leqslant r < n$.

For any set S, define $Perm(S)$ as the set of all bijective mappings on S (permutations on S), and $Func(S)$ as the set of all mappings from S to S. A block cipher E (or just a cipher) with a block size n and a key size k is the permutation family $\big(E_K \in Perm(\{0,1\}^n) \mid K \in \{0,1\}^k\big)$, where K is a key.

3 Security models

This section introduces models for an adversary that may repeat nonces in its queries.

We define security model using the notion of «experiment» or «game» played between a challenger and an adversary. The adversary and challenger

are modelled using consistent interactive probabilistic algorithms. The challenger simulates the functioning of the analysed cryptographic scheme for the adversary and may provide him access to one or more oracles (for details see [8]).

We describe challengers and adversaries using pseudocodes with the following notations. If a variable x gets a value val then we denote $x \leftarrow val$. Similarly, if a variable x gets the value of a variable y then we denote $x \leftarrow y$. If the variable x gets the result of a probabilistic algorithm $\mathcal{A}$ we denote $x \xleftarrow{\$} \mathcal{A}$. If we need to emphasize that $\mathcal{A}$ is deterministic than we denote it by $x \leftarrow \mathcal{A}$. The event when $\mathcal{A}$ returned value val as a result is denoted by $\mathcal{A} \rightarrow val$ (or $\mathcal{A} \xrightarrow{\$} val$ if $\mathcal{A}$ is probabilistic).

Firstly, we introduce the general definition of an AEAD-scheme.

Definition 1. *Let $\boldsymbol{K}$ be a set of keys, $\boldsymbol{P}$ be a set of plaintexts, $\boldsymbol{A}$ be a set of associated data, $\boldsymbol{C}$ be a set of ciphertexts, and $\boldsymbol{T}$ be a set of tags. An AEAD-scheme with nonce is a set of algorithms $\Pi = \{\Pi.\mathsf{Gen},\ \Pi.\mathsf{Enc},\ \Pi.\mathsf{Dec}\}$, where*

- $\Pi.\mathsf{Gen}() \xrightarrow{\$} K$: *A probabilistic key generation algorithm outputting a key $K \in \boldsymbol{K}$.*
- $\Pi.\mathsf{Enc}(K, N, A, P) \rightarrow (C, T)$: *A deterministic algorithm of authenticated encryption taking a key $K \in \boldsymbol{K}$, a nonce $N \in \boldsymbol{N}$, associated data $A \in \boldsymbol{A}$, a plaintext $P \in \boldsymbol{P}$. An output of the algorithm is a ciphertext $C \in \boldsymbol{C}$ and a tag $T \in \boldsymbol{T}$.*
- $\Pi.\mathsf{Dec}(K, N, A, C, T) \rightarrow P$: *A deterministic algorithm of authenticated decryption taking a key $K \in \boldsymbol{K}$, a nonce $N \in \boldsymbol{N}$, associated data $A \in \boldsymbol{A}$, a ciphertext $C \in \boldsymbol{C}$ and a tag $T \in \boldsymbol{T}$. An output of the algorithm is a plaintext $P \in \boldsymbol{P}$ or error symbol $\perp$.*

Let define a MRAE-int («Misuse-Resistant Authenticated Encryption - integrity») security notion for integrity (the integrity part of MRAE [19]).

Definition 2 (MRAE-int)**.** *For an AEAD-scheme Π the advantage of a* MRAE-int*-adversary $\mathcal{A}$ is defined as follows:*

$$\mathsf{Adv}_{\Pi}^{\text{MRAE-int}}(\mathcal{A}) = \Pr\left[\mathbf{Exp}_{\Pi}^{\text{MRAE-int}}(\mathcal{A}) \rightarrow 1\right],$$

where experiment $\mathbf{Exp}_{\Pi}^{\text{MRAE-int}}$ is defined below:

$\mathbf{Exp}_{\Pi}^{\text{MRAE-int}}(\mathcal{A})$	*Oracle Encrypt*(N, A, P)	*Oracle Decrypt*(N, A, C, T)
$K \xleftarrow{\$} \Pi.\mathsf{Gen}(\,)$	$(C, T) \leftarrow \Pi.\mathsf{Enc}(K, N, A, P)$	$P \leftarrow \Pi.\mathsf{Dec}(K, N, A, C, T)$
$sent \leftarrow \emptyset$	$sent \leftarrow sent \cup \{(N, A, C, T)\}$	**if** $(P \neq \perp) \wedge ((N, A, C, T) \notin sent)$:
$win \leftarrow \mathsf{false}$	**return** (C, T)	$win \leftarrow \mathsf{true}$
$\mathcal{A}^{Encrypt, Decrypt}(\,)$		**return** P
return win		

Let introduce the CPA-res («Chosen Plaintext Attack - resilience») security notion for confidentiality, defined in [4].

Definition 3 (CPA-res)**.** *For an AEAD-scheme* Π *with the tag length* s *the advantage of a* CPA-res*-adversary* $\mathcal{A}$ *is defined as follows:*

$$\mathsf{Adv}_{\Pi}^{\text{CPA-res}}(\mathcal{A}) = \Pr\left[\mathbf{Exp}_{\Pi}^{\text{CPA-res-1}}(\mathcal{A}) \to 1\right] - \Pr\left[\mathbf{Exp}_{\Pi}^{\text{CPA-res-0}}(\mathcal{A}) \to 1\right],$$

where experiments $\mathbf{Exp}^{\text{CPA-res-}b}$, $b \in \{0, 1\}$, *are defined below:*

$\mathbf{Exp}_{\Pi}^{\text{CPA-res-}b}(\mathcal{A})$	*Oracle* $O_1(N, A, P)$	*Oracle* $O_2(N, A, P)$
$K \xleftarrow{\$} \Pi.\mathsf{Gen}(\,)$	**if** $N \in \mathcal{L}_1 \cup \mathcal{L}_2$:	**if** $N \in \mathcal{L}_1$:
$\mathcal{L}_1, \mathcal{L}_2 \leftarrow \emptyset$	**return** $\perp$	**return** $\perp$
$b \xleftarrow{\$} \mathcal{A}^{O_1, O_2}(\,)$	**if** $b = 1$:	$(C, T) \leftarrow \Pi.\mathsf{Enc}(K, N, A, P)$
return b	$(C, T) \leftarrow \Pi.\mathsf{Enc}(K, N, A, P)$	**if** $N \notin \mathcal{L}_2$:
	else :	$\mathcal{L}_2 \leftarrow \mathcal{L}_2 \cup \{N\}$
	$C \parallel T \xleftarrow{\mathcal{U}} \{0, 1\}^{\lvert P\rvert + s}$	**return** (C, T)
	$\mathcal{L}_1 \leftarrow \mathcal{L}_1 \cup \{N\}$	
	return (C, T)	

In [4] the CCA-res («Chosen Ciphertext Attack - resilience») security notion is also defined. This notion differs from CPA-res in that an adversary is provided with additional access to a decryption oracle. By the technique similar to one described in [20] it is easy to show that MRAE-int-security and CPA-res-security jointly imply CCA-res-security. Therefore, further we consider the CPA-res security notion only.

4 MGM2 mode

In this section we describe a new AEAD mode called MGM2 which is a slight modification of the MGM mode. By $\mathsf{MGM2}[\mathsf{E}, r, s]$ we will denote the parametrized MGM2 mode with a block cipher E (with block size n and key size k), a nonce length r, $\frac{n}{2} \leqslant r \leqslant \frac{3n}{4}$ and a tag length s, $1 \leqslant s \leqslant n$.

For $\mathsf{MGM2}[\mathsf{E}, r, s]$ the corresponding sets are as follows: $\boldsymbol{K} = \{0,1\}^k$, $\boldsymbol{N} = \{0,1\}^r$, $\boldsymbol{A} = \boldsymbol{P} = \boldsymbol{C} = \{0,1\}^{\leqslant n(2^{n-r-2}-1)}$, $\boldsymbol{T} = \{0,1\}^s$. Moreover, the following condition should be satisfied: $0 < |A| + |P| \leqslant n(2^{n-r-2} - 1)$. The key generation, encryption and decryption algorithms are defined in Figure 1.

Difference from MGM. The main difference of the new MGM2 mode from the original MGM mode is in the modification of the way to produce the mask values for encryption (Γ_i), the coefficients of the multilinear function (H_i), and the tag values T. In MGM2 block cipher inputs, used to generate values for different use cases (we have three use cases: Γ_i, H_i, T), are separated by fixing the certain bits of inputs. Such a modification allows to obtain better security bounds, since the collision among block cipher inputs may occur only among values τ.

5 Security analysis

The security of block cipher modes of operation is commonly analyzed under assumption that the underlying block cipher is PRP-CPA-secure (see [7]), i.e. E_K for a random key is computationally indistinguishable from a random permutation. We follow this approach and provide security bounds directly for the mode with a random permutation.

We write $\mathsf{MGM2}[Perm(n), r, s]$ for MGM2 that uses a random permutation π as E_K and we write $\mathsf{MGM2}[Func(n), r, s]$ for MGM2 that uses a random function ρ.

5.1 Integrity

Theorem 1. *For any* MRAE-int*-adversary $\mathcal{A}$, making at most Q_E queries to the Encrypt oracle and at most Q_D queries to the Decrypt oracle, where the total block-length of associated data in all queries is at most σ_A and the total block-length of plaintexts and ciphertexts in all queries is at most σ_P,*

$$\mathsf{Adv}^{\text{MRAE-int}}_{\mathsf{MGM2}[Perm(n),r,s]}(\mathcal{A}) \leqslant \left(\frac{Q(Q-1)}{2^n} + \frac{Q_D}{2^s}\right)\left(1 - \frac{\sigma - 1}{2^n}\right)^{-\sigma/2}, \qquad (1)$$

where $Q = Q_E + Q_D$ and $\sigma = 2\sigma_P + \sigma_A + 2Q$.

Note that for $n \geqslant 128$ and $\sigma \leqslant 2^{n/2}$, the bound (1) can be converted as follows:

$$\mathsf{Adv}^{\text{MRAE-int}}_{\mathsf{MGM2}[Perm(n),r,s]}(\mathcal{A}) \leqslant 1.7\left(\frac{Q(Q-1)}{2^n} + \frac{Q_D}{2^s}\right). \qquad (2)$$

MGM2.Gen()

$K \xleftarrow{\mathcal{U}} \{0,1\}^k$
return K

MGM2.Enc(K, N, A, P)

$h \leftarrow |A|_n, q \leftarrow |P|_n$
$\ell \leftarrow h + q + 1$

.....Encryption.....

for $i = 1 \dots q$ **do**:
  $\Gamma_i \leftarrow \mathsf{E}_K(N \| 00 \| \mathrm{str}_{n-r-2}(i-1))$
$C \leftarrow P \oplus \mathrm{msb}_{|P|}(\Gamma_1 \| \dots \| \Gamma_q)$

.....Padding.....

$a \leftarrow n|A|_n - |A|$
$c \leftarrow n|C|_n - |C|$
$len \leftarrow \mathrm{str}_{n/2}(|A|) \| \mathrm{str}_{n/2}(|C|)$
$M_1 \| \dots \| M_l \leftarrow A \| 0^a \| C \| 0^c \| len$

.....Tag generation.....

for $i = 1 \dots \ell$ **do**:
  $H_i \leftarrow \mathsf{E}_K(N \| 01 \| \mathrm{str}_{n-r-2}(i-1))$
$\tau \leftarrow \mathsf{Set1}_r \left(\bigoplus_{i=1}^{l} M_i \otimes H_i \right)$
$T \leftarrow \mathrm{msb}_s(\mathsf{E}_K(\tau))$
return (C, T)

MGM2.Dec(K, N, A, C, T)

$h \leftarrow |A|_n, q \leftarrow |C|_n$
$\ell \leftarrow h + q + 1$

.....Padding.....

$a \leftarrow n|A|_n - |A|$
$c \leftarrow n|C|_n - |C|$
$len \leftarrow \mathrm{str}_{n/2}(|A|) \| \mathrm{str}_{n/2}(|C|)$
$M_1 \| \dots \| M_l \leftarrow A \| 0^a \| C \| 0^c \| len$

..... Tag verification .....

for $i = 1 \dots \ell$ **do**:
  $H_i \leftarrow \mathsf{E}_K(N \| 01 \| \mathrm{str}_{n-r-2}(i-1))$
$\tau \leftarrow \mathsf{Set1}_r \left(\bigoplus_{i=1}^{l} M_i \otimes H_i \right)$
$T' \leftarrow \mathrm{msb}_s(\mathsf{E}_K(\tau))$
if $T' \neq T$: **return** $\perp$

.....Decryption.....

for $i = 1 \dots q$ **do**:
  $\Gamma_i \leftarrow \mathsf{E}_K(N \| 00 \| \mathrm{str}_{n-r-2}(i-1))$
$P \leftarrow C \oplus \mathrm{msb}_{|C|}(\Gamma_1 \| \dots \| \Gamma_q)$
return P

Figure 1: AEAD mode MGM2

Thus, the MGM2 mode provides integrity beyond birthday bound. Note that for the original MGM mode, if total amount of processed data achieves $2^{n/2}$, the bound, presented in [1], becomes trivial. This result also allows to use MGM2 as a MAC function (and even as a PRF, see further) by fixing N with the constant value. Further we provide proof of the Theorem 1.

Proof. The proof is carried out in two steps. In the first step we introduce an auxiliary MAC-scheme with nonce called MGM2-MAC$[r, s]$ and estimate its security (see Section 5.1.1).

In the second step we show that the UF-CMA-security of the MGM2-MAC$[r, s]$ scheme tightly implies the MRAE-int-security of the MGM2$[Func(n), r, s]$ scheme (see Section 5.1.2).

The security bound for $\mathsf{MGM2}[Perm(n), r, s]$ is obtained using Bernstein's result [9], Theorem 2.3. Due to that theorem for any distinguisher $\mathcal{D}^f$ with oracle $f\colon \{0,1\}^n \to \{0,1\}^n$, making at most q queries, the following inequality holds:

$$\Pr[\mathcal{D}^{\pi} \to 1] \leqslant \Pr[\mathcal{D}^{\rho} \to 1] \cdot \left(1 - \frac{q-1}{2^n}\right)^{-q/2},$$

where $\pi \xleftarrow{\mathcal{U}} Perm(n)$ and $\rho \xleftarrow{\mathcal{U}} Func(n)$.

If we let $\mathcal{D}$ be the algorithm $\mathbf{Exp}^{\text{MRAE-int}}_{\mathsf{MGM2}[\cdot,r,s]}(\mathcal{A})$, where it makes queries to the oracle instead of calling the underlying function (block cipher), we obtain the target bound. □

5.1.1 Security of MGM2-MAC

We introduce an auxiliary MAC-scheme with nonce called $\mathsf{MGM2\text{-}MAC}[r, s]$ based on the scheme $\mathsf{MGM2}[Func(n), r, s]$. Usually MAC-scheme is defined as a set of algorithms $\mathsf{MAC} = \{\mathsf{MAC.Gen}, \mathsf{MAC.Tag}, \mathsf{MAC.Verify}\}$, for $\mathsf{MGM2\text{-}MAC}[r, s]$ these algorithms are defined in Figure 2. This scheme is defined for the message set $\{M = M_1 \| \dots \| M_\ell \colon M_i \in \{0,1\}^n,\ M_\ell \neq 0^n,\ 1 \leqslant \ell \leqslant 2^{n-r-2}\}$ (the message length is divisible by n, the last block is non-zero).

$\underline{\mathsf{MGM2\text{-}MAC.Gen}()}$

$\rho, \rho' \xleftarrow{\mathcal{U}} Func(n)$
$K \leftarrow (\rho, \rho')$
return K

$\underline{\mathsf{MGM2\text{-}MAC.Tag}(K, N, M)}$

$\tau \leftarrow \mathsf{PreTag}(\rho', N, M)$
$T \leftarrow \mathrm{msb}_s(\rho(\tau))$
return T

$\underline{\mathsf{PreTag}(\rho', N, M)}$

$l \leftarrow |M|_n$
for $i = 1 \dots \ell$ **do**:
  $H_i \leftarrow \rho'(N \| 01 \| \mathrm{str}_{n-r-2}(i-1))$
$\tau \leftarrow \mathsf{Set1}_r\left(\bigoplus_{i=1}^{l}(M_i \otimes H_i)\right)$
return τ

$\underline{\mathsf{MGM2\text{-}MAC.Verify}(K, N, M, T)}$

$\tau \leftarrow \mathsf{PreTag}(\rho', N, M)$
$T' \leftarrow \mathrm{msb}_s(\rho(\tau))$
if $T' \neq T$: **return** false
return true

Figure 2: The scheme MGM2-MAC

Firstly, we introduce the standard PRF security notion (in nonce-misuse setting) for nonce-based MAC-schemes and obtain the PRF-security bound

for the MGM2-MAC scheme.

Definition 4 (PRF)**.** *For a MAC-scheme* MAC *the advantage of a* PRF*-adversary* $\mathcal{A}$ *is defined as follows:*

$$\mathsf{Adv}^{\mathrm{PRF}}_{\mathsf{MAC}}(\mathcal{A}) = \Pr\left[\mathbf{Exp}^{\mathrm{PRF}-1}_{\mathsf{MAC}}(\mathcal{A}) \to 1\right] - \Pr\left[\mathbf{Exp}^{\mathrm{PRF}-0}_{\mathsf{MAC}}(\mathcal{A}) \to 1\right],$$

where experiments $\mathbf{Exp}^{\mathrm{PRF}-b}_{\mathsf{MAC}}(\mathcal{A})$, $b \in \{0,1\}$ *are defined below:*

$\mathbf{Exp}^{\mathrm{PRF}-b}_{\mathsf{MAC}}(\mathcal{A})$

if $b = 1$:
  $K \xleftarrow{\$} \mathsf{MAC.Gen}(\,)$
$sent \leftarrow \emptyset$
$b' \xleftarrow{\$} \mathcal{A}^{Tag^b}(\,)$
return b'

Oracle $Tag^1(N, M)$

if $(N, M) \in sent$:
  return $\perp$
$\boldsymbol{T} \leftarrow \mathsf{MAC.Tag}(K, N, M)$
$sent \leftarrow sent \cup \{(N, M)\}$
return T

Oracle $Tag^0(N, M)$

if $(N, M) \in sent$:
  return $\perp$
$T \xleftarrow{\mathcal{U}} \{0,1\}^s$
$sent \leftarrow sent \cup \{(N, M)\}$
return T

Lemma 1. *For any* PRF*-adversary* $\mathcal{A}$*, making at most* Q *queries to the* Tag *oracle:*

$$\mathsf{Adv}^{\mathrm{PRF}}_{\mathsf{MGM2\text{-}MAC}[r,s]}(\mathcal{A}) \leqslant \frac{Q(Q-1)}{2^n}.$$

Proof. Let define auxiliary experiments $\mathbf{Exp}^0$ and $\mathbf{Exp}^1$ (see Figure 3), which differ from the experiment $\mathbf{Exp}^{\mathrm{PRF}-1}_{\mathsf{MGM2\text{-}MAC}[r,s]}$ as follows. During the setup phase, the *tau* set is additionally initialized to empty value, and the flag *bad* is set to false. During the experiment execution, the values τ are put in the *tau* set, and the flag *bad* is set to true iff collision among the τ values occurs. Also, in the $\mathbf{Exp}^0$ experiment the tag value is chosen from $\{0,1\}^s$ uniformly at random in the case of collision (see line in box).

It is easy to see that $\mathbf{Exp}^1$ is exactly the $\mathbf{Exp}^{\mathrm{PRF}-1}_{\mathsf{MGM2\text{-}MAC}[r,s]}$ experiment. Moreover, for any $\mathcal{A}$ the value $\Pr\left[\mathbf{Exp}^0(\mathcal{A}) \Rightarrow 1\right]$ is exactly the value $\Pr\left[\mathbf{Exp}^{\mathrm{PRF}-0}_{\mathsf{MGM2\text{-}MAC}[r,s]}(\mathcal{A}) \Rightarrow 1\right]$. Indeed, in the $\mathbf{Exp}^0$ experiment all tag values T are produced according to the uniform distribution as in $\mathbf{Exp}^{\mathrm{PRF}-0}_{\mathsf{MGM2\text{-}MAC}[r,s]}$ for the following reasons. For the queries, whose corresponding τ value is new (not in the current set *tau*), the uniform random function ρ is applied to the new input and, therefore, returns uniform output. For the other queries the T value is directly sampled uniformly at random (see the line in box, Figure 3). Therefore,

$$\mathsf{Adv}^{\mathrm{PRF}}_{\mathsf{MGM2\text{-}MAC}[r,s]}(\mathcal{A}) = \Pr\left[\mathbf{Exp}^1(\mathcal{A}) \Rightarrow 1\right] - \Pr\left[\mathbf{Exp}^0(\mathcal{A}) \Rightarrow 1\right].$$

Note that before the *bad* flag is set to true (denote this event as $bad = \mathsf{true}$) the $\mathbf{Exp}^0$ and $\mathbf{Exp}^1$ experiments are functioning identically, therefore

$\mathbf{Exp}^b(\mathcal{A}),\ b \in \{0,1\}$

$(\rho, \rho') \xleftarrow{\$} \mathsf{MGM2\text{-}MAC.Gen}(\,)$
$bad \leftarrow \mathsf{false}$
$tau, sent \leftarrow \emptyset$
$b' \xleftarrow{\$} \mathcal{A}^{Tag^b}(\,)$
return b'

Oracle $Tag^b(N, M)$

if $(N, M) \in sent$:
  return $\perp$
$\tau \leftarrow \mathsf{PreTag}(\rho', N, M)$
$T \leftarrow \mathrm{msb}_s(\rho(\tau))$
if $\tau \in tau$:
  $bad \leftarrow \mathsf{true}$
  if $b = 0$: $T \xleftarrow{\mathcal{U}} \{0,1\}^s$
$tau \leftarrow tau \cup \{\tau\}$
$sent \leftarrow sent \cup \{(N, M)\}$
return T

Figure 3: Experiments $\mathbf{Exp}^0$ and $\mathbf{Exp}^1$

(due to Lemma 2, [8]) the following inequality holds:

$$\Pr\left[\mathbf{Exp}^1(\mathcal{A}) \Rightarrow 1\right] - \Pr\left[\mathbf{Exp}^0(\mathcal{A}) \Rightarrow 1\right] \leqslant \Pr[bad = \mathsf{true}]\,.$$

Let estimate $\Pr[bad = \mathsf{true}]$. Without loss of generality, we assume the adversary to be deterministic and making Q pairwise different queries $(N_i, M^i),\ i = 1, \ldots, Q$. Denote by $\widetilde{\rho}$ and $\widetilde{\rho}'$ the uniform random variables with sample space $Func(n)$. We will also use notation coll^i, $i = 2, \ldots, Q$, to denote the event that the *bad* flag is set to true during the first i queries processing. Thus,

$$\Pr[bad = \mathsf{true}] = \sum_{i=2}^{Q} \Pr\left[\mathsf{coll}^i \cap \overline{\mathsf{coll}^{i-1}}\right],$$

where the probability is defined by the random variables $\widetilde{\rho}$ and $\widetilde{\rho}'$. Let estimate the value $\Pr\left[\mathsf{coll}^i \cap \overline{\mathsf{coll}^{i-1}}\right]$ for any $i = 2, \ldots, Q$.

Note, that each i-th query – the pair (N_i, M^i), where $M^i = M^i_1 \| \ldots \| M^i_{l_i},\ M^i_j \in \{0,1\}^n$ – is determined by the tag values $T_1, \ldots, T_{i-1}$ previously obtained from the oracle. Without loss of generality, we assume $l_1 = \ldots = l_i$. Indeed, otherwise we can pad the messages with zero blocks to the length $l := \max(l_1, \ldots, l_i)$. This does not change the tag value, and the padded messages will stay pairwise different because of $M^j_{l_j} \neq 0^n$. Therefore, the $T_1, \ldots, T_{i-1}$ values fully determine l and $(N_1, M^1), \ldots, (N_i, M^i)$.

For fixed N_j we denote by $\widetilde{H^j_k},\ j = 1, \ldots, i;\ k = 1, \ldots, l$, the random variable $\widetilde{\rho}'(N_j \| 01 \| \mathrm{str}_{n-r-2}(k-1))$. Notice that $\Pr\left[\widetilde{H^j_k} = B\right] = \dfrac{1}{2^n}$ for any

$B \in \{0,1\}^n$. Note that the random variables $\widetilde{H_k^j}$ and $\widetilde{H_k^t}$ for some $j \neq t$ and any k are dependent, namely $\Pr\left[\widetilde{H_k^j} = \widetilde{H_k^t}\right] = 1$, iff $N_k = N_j$.

For short we denote by $\widetilde{H^j}$ the random variable $(\widetilde{H_1^j}, \ldots, \widetilde{H_\ell^j})$. Also for set $H = (H_1, \ldots, H_\ell)$ and message $M = M_1 \| \ldots \| M_\ell$ let $\tau(H, M)$ be the function $\mathsf{Set1}_r\left(\bigoplus_{k=1}^{l} H_k \otimes M_k\right)$. So, we have

$$\Pr\left[\mathsf{coll}^i \cap \overline{\mathsf{coll}^{i-1}}\right] = \sum_{T_1,\ldots,T_{i-1}} \Pr\left[\mathsf{coll}^i \cap \overline{\mathsf{coll}^{i-1}} \cap \{\widetilde{T}_j = T_j\}_{j=1}^{i-1}\right],$$

where we write $\widetilde{T}_j$ for random variable $\mathrm{msb}_s(\widetilde{\rho}(\tau(\widetilde{H^j}, M^j)))$, and sum is taken over all $(T_1, \ldots, T_{i-1}) \in (\{0,1\}^s)^{i-1}$.

For fixed $(N_1, M^1), \ldots, (N_i, M^i)$ introduce the following conditions on set $H^1, \ldots, H^i$, $H^j := (H_1^j, \ldots, H_\ell^j)$, $j = 1, \ldots, i$:

Condition E_1: $\forall\ j, t,\ 1 \leqslant j < t \leqslant i-1$: $\tau(H^j, M^j) \neq \tau(H^t, M^t)$.

Condition E_2: $\exists\ j,\ 1 \leqslant j \leqslant i-1$: $\tau(H^i, M^i) = \tau(H^j, M^j)$.

For any fixed $T_1, \ldots, T_{i-1}$, and hence fixed $(N_1, M^1), \ldots, (N_i, M^i)$, the event $\mathsf{coll}^i \cap \overline{\mathsf{coll}^{i-1}}$ occurs iff random variables $\widetilde{H^1}, \ldots, \widetilde{H^i}$ take such values $H^1, \ldots, H^i$ that the conditions E_1 and E_2 are satisfied. For short we will denote the events that these conditions are satisfied by the same way, namely, by E_1 and E_2 correspondingly.

Note that fixing values H^j, $j = 1, \ldots, i$, leads to fixing values $\tau_j := \tau(H^j, M^j)$. Therefore,

$$\begin{aligned}
\Pr\left[\mathsf{coll}^i \cap \overline{\mathsf{coll}^{i-1}}\right] &= \sum_{T_1,\ldots,T_{i-1}} \Pr\left[\mathsf{E}_1 \cap \mathsf{E}_2 \cap \{\widetilde{T}_j = T_j\}_{j=1}^{i-1}\right] = \\
&= \sum_{T_1,\ldots,T_{i-1}} \sum_{\substack{H^1,\ldots,H^i: \\ \mathsf{E}_1 \cap \mathsf{E}_2}} \Pr\left[\{\widetilde{H^j} = H^j\}_{j=1}^{i} \cap \{\mathrm{msb}_s(\widetilde{\rho}(\tau_j)) = T_j\}_{j=1}^{i-1}\right] = \\
&= \sum_{T_1,\ldots,T_{i-1}} \sum_{\substack{H^1,\ldots,H^i: \\ \mathsf{E}_1 \cap \mathsf{E}_2}} \Pr\left[\{\widetilde{H^j} = H^j\}_{j=1}^{i}\right] \cdot \Pr\left[\{\mathrm{msb}_s(\widetilde{\rho}(\tau_j)) = T_j\}_{j=1}^{i-1}\right].
\end{aligned}$$

Here, sum is taken over all $H^1, \ldots, H^i$, $H^j \in (\{0,1\}^n)^l$, for which the E_1 and E_2 conditions are satisfied. The last transition is due to the fact that $\widetilde{\rho}$ and $\widetilde{H^j}$, $j = 1, \ldots, i$, are independent.

Consider the value $\Pr\left[\{\mathrm{msb}_s(\widetilde{\rho}(\tau_j)) = T_j\}_{j=1}^{i-1}\right]$. For any $T_1, \ldots, T_{i-1}$ and $H^1, \ldots, H^{i-1}$ for which the condition E_1 is satisfied, this probability is exactly

the probability to sample function ρ, such that $i-1$ fixed inputs correspond to outputs with fixed s bits, i.e $\dfrac{1}{2^{s(i-1)}}$. Thus:

$$\Pr\left[\mathsf{coll}^i \cap \overline{\mathsf{coll}^{i-1}}\right] = \sum_{T_1,\ldots,T_{i-1}} \sum_{\substack{H^1,\ldots,H^i:\\ \mathsf{E}_1\cap\mathsf{E}_2}} \Pr\left[\{\widetilde{H^j} = H^j\}_{j=1}^{i}\right] \cdot \frac{1}{2^{s(i-1)}} =$$

$$= \frac{1}{2^{s(i-1)}} \sum_{T_1,\ldots,T_{i-1}} \Pr[\mathsf{E}_1 \cap \mathsf{E}_2] \leqslant \frac{1}{2^{s(i-1)}} \sum_{T_1,\ldots,T_{i-1}} \Pr[\mathsf{E}_2]\,.$$

Now consider $\Pr[\mathsf{E}_2]$ for any fixed $T_1,\ldots,T_{i-1}$, and, hence, any fixed $(N_1,M^1),\ldots,(N_i,M^i)$.

$$\Pr[\mathsf{E}_2] = \Pr\left[\exists\ j,\ 1\leqslant j\leqslant i-1\colon\ \tau(\widetilde{H^i},M^i) = \tau(\widetilde{H^j},M^j)\right] =$$

$$= \Pr\left[\bigcup_{j=1}^{i-1}\left\{\tau(\widetilde{H^i},M^i) = \tau(\widetilde{H^j},M^j)\right\}\right] \leqslant \sum_{j=1}^{i-1}\Pr\left[\tau(\widetilde{H^i},M^i) = \tau(\widetilde{H^j},M^j)\right].$$

Let estimate $p := \Pr\left[\tau(\widetilde{H^i},M^i) = \tau(\widetilde{H^j},M^j)\right]$ for any $j = 1,\ldots,i-1$. We consider two cases:

1. $N_i \neq N_j$ (in this case $\widetilde{H_k^i}$ and $\widetilde{H_k^j}$ are independent).
2. $N_i = N_j$ (in this case $\widetilde{H_k^i}$ and $\widetilde{H_k^j}$ are dependent).

The first case: $p = \dfrac{\#\{H^i,H^j\colon \tau(H^i,M^i) = \tau(H^j,M^j)\}}{2^{2nl}}$.

$$\#\{H^i,H^j\colon \tau(H^i,M^i) = \tau(H^j,M^j)\} =$$

$$= \#\left\{H^i,H^j\colon \bigoplus_{k=1}^{l} H_k^i\otimes M_k^i = \bigoplus_{k=1}^{l} H_k^j\otimes M_k^j\right\} +$$

$$+ \#\left\{H^i,H^j\colon \bigoplus_{k=1}^{l} H_k^i\otimes M_k^i = \bigoplus_{k=1}^{l} H_k^j\otimes M_k^j \oplus \mathsf{Set1}_r(0^n)\right\}.$$

Since $M_{\ell_i}^i \neq 0^n$ for any i, the cardinality is $2\cdot 2^{n(2l-1)}$. And, $p = \dfrac{2}{2^n}$.

The second case: $p = \dfrac{\#\{H^i \colon \tau(H^i, M^i) = \tau(H^i, M^j)\}}{2^{nl}}$.

$$\#\{H^i \colon \tau(H^i, M^i) = \tau(H^i, M^j)\} =$$
$$= \#\Big\{H^i \colon \bigoplus_{k=1}^{l} H_k^i \otimes (M_k^i \oplus M_k^j) = 0^n\Big\} +$$
$$+ \#\Big\{H^i \colon \bigoplus_{k=1}^{l} H_k^i \otimes (M_k^i \oplus M_k^j) = \mathsf{Set1}_r(0^n)\Big\}.$$

Since for the same nonce the messages M^i and M^j should be different, there exists k such that $M_k^i \oplus M_k^j \neq 0^n$. Therefore, the cardinality is $2 \cdot 2^{n(l-1)}$. And, $p = \dfrac{2}{2^n}$.

Summing up, we have:

$$\Pr[bad = \mathsf{true}] = \sum_{i=2}^{Q} \frac{1}{2^{s(i-1)}} \sum_{T_1,\ldots,T_{i-1}} \sum_{j=1}^{i-1} \frac{2}{2^n} = \sum_{i=2}^{Q} \frac{i-1}{2^{n-1}} = \frac{Q(Q-1)}{2^n}.$$

□

Now we introduce the standard UF-CMA security notion for nonce-based MAC-schemes and obtain the UF-CMA-security bound for the MGM2-MAC scheme.

Definition 5. *For a MAC-scheme* MAC *the advantage of a* UF-CMA*-adversary* $\mathcal{A}$ *is defined as follows:*

$$\mathsf{Adv}^{\text{UF-CMA}}_{\mathsf{MAC}}(\mathcal{A}) = \Pr\left[\mathbf{Exp}^{\text{UF-CMA}}_{\mathsf{MAC}}(\mathcal{A}) \to 1\right],$$

where experiment $\mathbf{Exp}^{\text{UF-CMA}}_{\mathsf{MAC}}(\mathcal{A})$ *is defined below:*

$\mathbf{Exp}^{\text{UF-CMA}}_{\mathsf{MAC}}(\mathcal{A})$	*Oracle* $Tag(N, M)$	*Oracle* $Verify(N, M, T)$
$K \xleftarrow{\$} \mathsf{MAC.Gen}(\,)$	**if** $(N, M) \in sent$:	$res \leftarrow \mathsf{MAC.Vf}(K, N, M, T)$
$sent \leftarrow \emptyset$	**return** $\perp$	**if** $res \wedge ((N, M) \notin sent)$:
$win \leftarrow \mathsf{false}$	$T \leftarrow \mathsf{MAC.Tag}(K, N, M)$	$win \leftarrow \mathsf{true}$
$\mathcal{A}^{Tag, Verify}(\,)$	$sent \leftarrow sent \cup \{(N, M)\}$	**return** res
return win	**return** T	

Using Proposition 7.3 [7] and Lemma 1 we obtain the following result.

Corollary 1. *For any* UF-CMA*-adversary* $\mathcal{A}$*, making at most* Q_T *queries to the* Tag *oracle and at most* Q_V *queries to the* $Verify$ *oracle:*

$$\mathsf{Adv}^{\text{UF-CMA}}_{\mathsf{MGM2\text{-}MAC}[r,s]}(\mathcal{A}) \leqslant \frac{Q(Q-1)}{2^n} + \frac{Q_V}{2^s},$$

where $Q = Q_T + Q_V$.

5.1.2 Security of MGM2 with random function

Lemma 2. *For any* MRAE-int*-adversary* $\mathcal{A}$*, making at most* Q_E *queries to the Encrypt oracle and at most* Q_D *queries to the Decrypt oracle, there exists a* UF-CMA*-adversary* $\mathcal{B}$*, making at most* Q_E *queries to the* Tag *oracle and at most* Q_D *queries to the* $Verify$ *oracle, such that*

$$\mathsf{Adv}^{\text{MRAE-int}}_{\mathsf{MGM2}[Func(n),r,s]}(\mathcal{A}) \leqslant \mathsf{Adv}^{\text{UF-CMA}}_{\mathsf{MGM2\text{-}MAC}[r,s]}(\mathcal{B})$$

Proof. Let construct an adversary $\mathcal{B}$, that uses the adversary $\mathcal{A}$ as a black box. The adversary $\mathcal{B}$ (see Figure 4) intercepts the queries of the adversary $\mathcal{A}$ and process them by itself using its own oracles. For encryption/decryption $\mathcal{B}$ implements lazy sampling for ρ''. For tag generation/tag verification the adversary $\mathcal{B}$ implements the padding procedure and send the appropriate query to its oracles.

$\mathcal{B}^{Tag,Verify}_{\mathcal{A}}$

$\rho'' \xleftarrow{\mathcal{U}} Func(n)$ // lazy sampling
return $\mathcal{A}^{SEncrypt,SDecrypt}(\,)$

$SEncrypt(N, A, P)$

$h \leftarrow |A|_n, q \leftarrow |P|_n$

. . . . . Encryption . . . . . .

for $i = 1 \ldots q$ **do**:
  $\Gamma_i \leftarrow \rho''(N \| 00 \| \mathrm{str}_{n-r-2}(i-1))$
$C \leftarrow P \oplus \mathrm{msb}_{|P|}(\Gamma_1 \| \ldots \| \Gamma_q)$

. . . . . Padding . . . . . .

$a \leftarrow n|A|_n - |A|$
$c \leftarrow n|C|_n - |C|$
$len \leftarrow \mathrm{str}_{n/2}(|A|) \| \mathrm{str}_{n/2}(|C|)$
$M \leftarrow A \| 0^a \| C \| 0^c \| len$

. . . . . Tag Genetation . . . . . .

$T \leftarrow Tag(N, M)$
return (C, T)

Oracle $SDecrypt(N, A, C, T)$

$h \leftarrow |A|_n, q \leftarrow |C|_n$

. . . . . Padding . . . . . .

$a \leftarrow n|A|_n - |A|$
$c \leftarrow n|C|_n - |C|$
$len \leftarrow \mathrm{str}_{n/2}(|A|) \| \mathrm{str}_{n/2}(|C|)$
$M \leftarrow A \| 0^a \| C \| 0^c \| len$

. . . . . Tag Verification . . . . . .

if $Verify(N, M, T) = \mathsf{false}$:
  return $\perp$

. . . . . Decryption . . . . . .

for $i = 1 \ldots q$ **do**:
  $\Gamma_i \leftarrow \rho''(N \| 00 \| \mathrm{str}_{n-r-2}(i-1))$
$P \leftarrow C \oplus \mathrm{msb}_{|C|}(\Gamma_1 \| \ldots \| \Gamma_q)$
return P

Figure 4: Adversary $\mathcal{B}$

Note that the adversary $\mathcal{B}$ simulates for $\mathcal{A}$ exactly the experiment $\mathbf{Exp}^{\text{MRAE-int}}_{\mathsf{MGM2}[Func(n),r,s]}$. Indeed, since for $\mathsf{MGM2}[Func(n), r, s]$ the inputs to the random function in case of 1) tag generation, 2) computing values H_i and

3) computing values Γ_i are different (because of fixed bits in inputs), using one random function is indistinguishable from using three independent random functions ρ, ρ', ρ'' for these three cases. Also, note that messages M, formed by $\mathcal{B}$, satisfy conditions for message set of $\mathsf{MGM2\text{-}MAC}[r, s]$.

If the adversary $\mathcal{A}$ forges, then the adversary $\mathcal{B}$ also forges in $\mathbf{Exp}^{\text{UF-CMA}}_{\mathsf{MGM2\text{-}MAC}[r,s]}$. Indeed, if $\mathcal{A}$ makes non-trivial valid query (N, A, C, T) to the *Decrypt* oracle, then the adversary makes $\mathcal{B}$ corresponding non-trivial query $(N, M = A\|0^a\|C\|0^c\|len, T)$ to the $Verify$ oracle. □

5.2 Confidentiality

Theorem 2. *For any* CPA-res-*adversary* $\mathcal{A}$*, making at most* Q_1 *queries to the* O_1 *oracle and at most* Q_2 *queries to the* O_2 *oracle, where the total block-length of associated data in all queries is at most* σ_A *and the total block-length of plaintext and ciphertexts in all queries is at most* σ_P*,*

$$\mathsf{Adv}^{\text{CPA-res}}_{\mathsf{MGM2}[Perm(n),r,s]}(\mathcal{A}) \leqslant \frac{\sigma^2}{2^{n+1}} + \frac{Q(Q-1)}{2^{n-1}}, \tag{3}$$

where $Q = Q_1 + Q_2$ *and* $\sigma = 2\sigma_P + \sigma_A + 2Q$*.*

Proof. Firstly, we apply PRP-PRF switching lemma [12] to replace $Perm(n)$ by $Func(n)$ (this gives us the term $\frac{\sigma^2}{2^{n+1}}$ in the bound), and then we obtain the CPA-res-security bound for $\mathsf{MGM2}[Func(n), r, s]$.

The security bound for $\mathsf{MGM2}[Func(n), r, s]$ is obtained in the same way as in the proof of Theorem 1. Indeed, ciphertexts C, received from the O_1 oracle, are absolutely indistinguishable from uniform random strings since the inputs to the uniform random function ρ used to produce Γ_i are unique. The indistinguishability of the tags T, received from the O_1 oracle, from uniform random strings is estimated by constructing two PRF-adversaries for MGM2-MAC that uses CPA-res-adversary as a black box. Therefore, $\mathsf{Adv}^{\text{CPA-res}}_{\mathsf{MGM2}[Func(n),r,s]}(\mathcal{A}) \leqslant \frac{Q(Q-1)}{2^{n-1}}$. □

6 Conclusion

In the current paper we introduce the modification of the MGM mode — the MGM2 mode. For this mode we obtain the security bounds for non-standard notions MRAE-int and CPA-res, allowing the adversary to repeat nonces. In comparison with the original mode, the security proof appears to be rather simple and short.

In the future work we are going to develop a SIV-construction (see [15]) of the MGM2 mode to achieve MRAE-conf-security. Also we are going to incorporate re-keying mechanisms in the MGM2 mode to achive new security properties like leakage-resilience and increase key lifetime.

References

[1] Akhmetzyanova L., Alekseev E., Karpunin G., Nozdrunov V. *Security of Multilinear Galois Mode (MGM).*, IACR Cryptology ePrint Archive 2019, p. 123, 2019.

[2] Akhmetzyanova L., Alekseev E., Smyshlyaev S., Oshkin I. (2020) *On Internal Re-keying.* In: van der Merwe T., Mitchell C., Mehrnezhad M. (eds) Security Standardisation Research. SSR 2020. Lecture Notes in Computer Science, vol 12529. Springer, Cham. https://doi.org/10.1007/978-3-030-64357-7_2

[3] Andreeva E., Bogdanov A., Luykx A., Mennink B., Mouha N., Yasuda K. (2014) *How to Securely Release Unverified Plaintext in Authenticated Encryption.* In: Sarkar P., Iwata T. (eds) Advances in Cryptology – ASIACRYPT 2014. ASIACRYPT 2014. Lecture Notes in Computer Science, vol 8873. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-45611-8_6

[4] Ashur T., Dunkelman O., Luykx A. *Boosting authenticated encryption robustness with minimal modifications* //Annual International Cryptology Conference. – Springer, Cham, 2017. – С. 3-33.

[5] Davide Bellizia and Olivier Bronchain and Gaëtan Cassiers and Vincent Grosso and Chun Guo and Charles Momin and Olivier Pereira and Thomas Peters and François-Xavier Standaert, *Mode-Level vs. Implementation-Level Physical Security in Symmetric Cryptography: A Practical Guide Through the Leakage-Resistance Jungle*, Cryptology ePrint Archive, Report 2020/211, 2020, https://eprint.iacr.org/2020/211

[6] Bellare M., Namprempre C. *Authenticated encryption: Relations among notions and analysis of the generic composition paradigm* //International Conference on the Theory and Application of Cryptology and Information Security. – Springer, Berlin, Heidelberg, 2000. – С. 531-545.

[7] Bellare M., Rogaway P. *Introduction to modern cryptography* //Ucsd Cse. – 2005. – Т. 207. – С. 207.

[8] Bellare M., Rogaway P. *The Security of Triple Encryption and a Framework for Code- Based Game-Playing Proofs* // LNCS, Advances in Cryptology - EUROCRYPT 2006, 4004, ed. Vaudenay S., Springer, Berlin, Heidelberg, 2006.

[9] Bernstein, D.J.: *Stronger Security Bounds for Permutations* (2005), http://cr.yp.to/papers.html (accessed on May 31, 2012)

[10] John Black, Phillip Rogaway, and Thomas Shrimpton. 2002.*Encryption-Scheme Security in the Presence of Key-Dependent Messages.* In Revised Papers from the 9th Annual International Workshop on Selected Areas in Cryptography (SAC '02). Springer-Verlag, Berlin, Heidelberg, 62–75.

[11] CAESAR competion. `https://competitions.cr.yp.to/caesar.html`

[12] D. Chang and M. Nandi, *A Short Proof of the PRP/PRF Switching Lemma* // IACR ePrint Archive, 2008, Report 2008/078, https://eprint.iacr.org/2008/078.

[13] Federal Agency on Technical Regulating and Metrology, *Information technology. Cryptographic data security. Authenticated encryption block cipher operation modes*, R 1323565.1.026-2019, 2019.

[14] Federal Agency on Technical Regulating and Metrology, *Information technology. Cryptographic data security. Cryptographic algorithms accompanying the use of block ciphers*, R 1323565.1.017—2018, 2018.

[15] Gueron S., Lindell Y. *GCM-SIV: full nonce misuse-resistant authenticated encryption at under one cycle per byte* //Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. – 2015. – С. 109-119.

[16] Hoang V.T., Krovetz T., Rogaway P. (2015) *Robust Authenticated-Encryption AEZ and the Problem That It Solves*. In: Oswald E., Fischlin M. (eds) Advances in Cryptology – EUROCRYPT 2015. EUROCRYPT 2015. Lecture Notes in Computer Science, vol 9056. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-46800-5_2

[17] Kurochkin A., Fomin D. *MGM Beyond the Birthday Bound* // 8th Workshop on Current Trends in Cryptology (CTCrypt 2019).

[18] Rogaway P. (2004) *Nonce-Based Symmetric Encryption*. In: Roy B., Meier W. (eds) Fast Software Encryption. FSE 2004. Lecture Notes in Computer Science, vol 3017. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-25937-4_22

[19] Rogaway P., Shrimpton T. *A provable-security treatment of the key-wrap problem* //Annual International Conference on the Theory and Applications of Cryptographic Techniques. – Springer, Berlin, Heidelberg, 2006. – С. 373-390.

[20] Shrimpton T.: *A Characterization of Authenticated-Encryption as a Form of Chosen-Ciphertext Security*. In: Cryptology ePrint Archive, Report 2004/272 (2004).

[21] Smyshlyaev, S., Nozdrunov, V., Shishkin, V., and E. Smyshlyaeva *Multilinear Galois Mode (MGM)* // 2019, <https://tools.ietf.org/html/draft-smyshlyaev-mgm-17>

[22] Shrimpton T., Terashima R. S. *A modular framework for building variable-input-length tweakable ciphers* //International Conference on the Theory and Application of Cryptology and Information Security. – Springer, Berlin, Heidelberg, 2013. – С. 405-423.

IQRA: Incremental Quadratic Re-keying friendly Authentication scheme

Liliya Akhmetzyanova, Evgeny Alekseev, Alexandra Babueva, Lidiia Nikiforova, and Stanislav Smyshlyaev

CryptoPro LLC, Russia
{lah, alekseev, babueva, nikiforova, svs}@cryptopro.ru

Abstract

The notion of incremental cryptography was introduced by Bellare, Goldwasser and Goldreich in 1994 and becomes more and more relevant in the big data world. Incremental mechanisms allow to quickly update the result of the algorithm for a modified data, rather than having to re-compute it from scratch. A significant flaw of the existing incremental schemes, specifically incremental MACs, is that they lose incremental property in case of key update. In the current paper we propose new incremental re-keying friendly MAC scheme, called IQRA, based on quadratic multivariate polynomial and PRF. We define the way how to use the IQRA scheme with re-keying mechanism based on KDF and introduce SUF-CSMA notion to analyze the security of this composition. We provide the security bound for the proposed scheme and improve it for the special case when a block cipher is used as the underlying PRF.

Keywords: incremental cryptography, incremental MAC, re-keying, IQRA scheme, provable security

1 Introduction

Incremental cryptography is a powerful tool for working with dynamically changing data. The idea of incremental constructions is to provide efficient updates compared to classical algorithms. Traditionally, the result of a cryptographic operation is re-computed from scratch after each message modification regardless of the number of modifications. Incremental schemes allow to update the result in a time proportional to the number of modified blocks (usually insert, delete and replace operations are considered). This concept was introduced by Bellare, Goldreich and Goldwasser in 1994 [12] and has evolved since then, leading to creation of many cryptographic primitives such as encryption schemes [13, 6, 7], signatures [12, 26, 32], MACs [13, 26, 33, 24],

hash functions [27, 15, 31], PRFs [5] and authenticated encryption constructions [21, 6, 36].

In the current paper we focus on incremental MAC schemes. Originally proposed for the virus protection [12], they have much wider range of application now: from sensor networks [28, 31] to securing storage in mobile cloud computing [22]. A prime example is incremental MAC usage in the full disk encryption constructions. Usually the disk space is represented as a set of sectors processed separately via read and write operations. Computing the standard MAC over all disk content to ensure data integrity is too much time-consuming since MAC should be re-generated after each sector modification. One of the solutions proposed in [23] is to use a regular MAC scheme to compute a local tag for each sector and an incremental MAC to ensure the authenticity of the local tags. Then for each sector modification the corresponding local tag is re-computed from scratch and the global tag is updated quickly according to the local tag.

Clearly incremental MAC schemes are especially important when dealing with big data (e.g. up to 2^{40} modifications could be done for each sector during disk lifetime). At the same time processing large amount of data goes hand in hand with key lifetime control. The restrictions on the maximal amount of data processed with one key come either from combinatorial properties of the used construction (most block cipher based modes of operation are secure up to the birthday paradox bound [10]), or from side-channel attacks [38]. The methods for increasing the lifetime of symmetric keys are introduced in [37] and are discussed in detail in [1, 2, 3]. The most obvious way to overcome the limitations is to change key regularly. The problem is that idea of incrementality runs counter to the re-keying approach by default. Indeed, when the key lifetime limitation is reached at time of the update operation and the key is renewed, the tag should be re-computed from scratch with the new key and thus we can no longer talk about update time proportional to the number of modifications.

Related Works. There is a variety of incremental MAC schemes proposed in the literature: XOR-MACs [14], PMAC [20], GMAC [34], PWC [35], ZMAC [29]. All of them lose their incremental property in case of key changing. The core problem is that the same key is used for processing all blocks and thus all interim results refered to separate blocks should be re-computed as a consequence of key changing. As well, the iHtE (incremental Hash-then-Encrypt) construction proposed in [5] for building incremental PRF (and thus incremental MAC) does not take into account the necessity of key lifetime

control. Thus, the designing of incremental re-keying friendly MAC scheme is the relevant task.

The security notions for incremental MACs were introduced in [12] at the idea level and were formally defined in [23]. Moreover, [5] introduces incremental unforgeability notion (iUF) that is some mix of notions defined in [23]. The relation among known notions is discussed in [23].

Our Contribution. In the current paper we propose new incremental MAC scheme compatible with re-keying techniques. Our scheme is nonce-based, uses quadratic multivariate polynomial and PRF as the underlying primitives and supports replace operation. We call it IQRA (Incremental Quadratic Re-keying friendly Authentication) scheme. The main idea behind the IQRA design is to separate the keys using for different purposes (for processing each block and nonce). Thus it is possible to change each key independently if necessary without violating the incrementality.

We propose the way how to use the IQRA scheme with external re-keying mechanism realized with KDF function. We also introduce SUF-CSMA (Strong Unforgeability under Chosen Settings and Message Attack) notion to examine the IQRA security with the proposed key derivation technique. This model allows the adversary to control not only nonce and message, but also the parameters for key derivation. We show that despite the absence of *Update* oracle access, the proposed model is the extension of the basic security model for incremental MAC schemes defined in [12].

Finally, we provide the security bound for composition of the IQRA and KDF schemes in the SUF-CSMA model in the general case and in the special case when block cipher is used as the underlying PRF.

Organization of the paper. The remainder of the paper is organized as follows. In Section 2 basic definitions and notations are introduced. Section 3 introduces the IQRA scheme and describes how it can be combined with re-keying mechanisms. In Section 4 we introduce SUF-CSMA notion and define basic security notions for the used primitives. Section 5 is devoted to the security analysis of the proposed scheme. We draw our conclusions in Section 6. Detailed proofs of our results are relegated to the appendices because of space limitations.

2 Basic notations and definitions

For any set A, by A^s we denote the set of s-component strings with elements from A. By $\{0,1\}^*$ we denote the set of all bit strings of finite length including the empty string. For bit strings a and b we denote by $a\|b$ their concatenation. For bit string $a \in \{0,1\}^n$ and $b \in \{0,1\}^n$ we denote by $a \cdot b$ a string which is the result of their multiplication in $GF(2^n)$ (here strings encode polynomials in the standard way). For nonnegative integers u and x let $\mathrm{str}_u(x)$ be u-bit representation of x with the least significant bit on the right. For integer $u \geqslant 0$ by $L(u)$ we denote minimal bit length that is multiple of byte needed for u representation: $L(u) = 8 \cdot \left\lceil \frac{\log_2(u)}{8} \right\rceil$. We denote as $0\mathrm{x}b$ a bit representation of the hex number b.

For any set A and B, define $Perm(A)$ as the set of all bijective mappings on A (permutations on A) and $Func(A,B)$ as the set of all mappings from A to B. A block cipher E with a block size n and a key size k is the permutation family $\left\{\mathrm{E}(K,\cdot) \in Perm(\{0,1\}^n) \mid K \in \{0,1\}^k\right\}$, where K is a key.

If the value x is chosen from a set S uniformly at random, then we denote $x \xleftarrow{\mathcal{U}} S$. If the variable x gets the value val then we denote $x \leftarrow val$. If the variable x gets the result of a probabilistic algorithm A we denote $x \xleftarrow{\$} A$. The event when A returned value val as a result is denoted by $A \to val$.

We define security properties using the notion of «experiment» played between a challenger and an adversary. This approach is introduced in [17] and is thoroughly discussed in [4]. The adversary and challenger are modelled using consistent interactive probabilistic algorithms. The challenger simulates the functioning of the analysed cryptographic scheme for the adversary and may provide him access to one or more oracles. Notation $\mathcal{A}^{\mathcal{O}_1,\mathcal{O}_2,\cdots} \to 1$ means that the adversary $\mathcal{A}$, after interacting with oracles $\mathcal{O}_1, \mathcal{O}_2, \ldots$, outputs 1. The parameters of an adversary $\mathcal{A}$ are its computational resources (for a fixed model of computation and a method of encoding) and oracles query complexity. The query complexity usually includes the number and the length of queries. Denote by $\mathsf{Adv}_{\mathrm{S}}^{M}(\mathcal{A})$ the measure of the success rate of the adversary $\mathcal{A}$ in realizing a certain threat, defined by the security notion M for the cryptographic scheme S. The formal definition of this measure will be given in each specific case.

3 IQRA scheme description

We follow the line of papers oriented on incremental MACs [25, 23, 5] by using the corresponding term «document» instead of «message».

When designing the scheme, we focus on the applications with the fixed length data (e. g. full disk encryption mechanisms). Therefore, we define the update algorithm only for replace operation. We believe that slight modifications are required to support more modification operations such as appending of a block to a document, but we do not analyze it in the current paper and leave this for further research.

Nonce-based IQRA (Incremental Quadratic Re-keying friendly Authentication) scheme is specified as follows:

$$\text{IQRA} = (\text{F}, \mathsf{BS}, \mathsf{DS}, \mathsf{KS}, \mathsf{NS}, \mathsf{TS}, \text{Tag}, \text{Replace}),$$

where

- $\text{F} : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ — underlying PRF;
- a block space $\mathsf{BS} = \{0,1\}^n$ and a document space $\mathsf{DS} = \mathsf{BS}^w$, meaning a document has the form $\mathsf{D} = (D_0, \ldots, D_{w-1})$ and $D_i \in \mathsf{BS}$ for all $i \in \{0, \ldots, w-1\}$;
- a key space $\mathsf{KS} = \mathsf{FKS} \times \mathsf{BKS}^w$, where a finalization key space $\mathsf{FKS} = \{0,1\}^k$ and a block key space $\mathsf{BKS} = \{0,1\}^n$; we will denote key as a pair of finalization key $Key \in \mathsf{FKS}$ and vector of block keys $\mathsf{K} \in \mathsf{BKS}^w$, such that $\mathsf{K} = (K_0, \ldots, K_{w-1})$, $K_i \in \mathsf{BKS}$ for all $i \in \{0, \ldots, w-1\}$;
- a nonce space NS and a tag space TS are equal to $\{0,1\}^n$;
- a tagging algorithm Tag, that takes a key $(Key, \mathsf{K}) \in \mathsf{KS}$, a nonce $N \in \mathsf{NS}$ and document $\mathsf{D} \in \mathsf{DS}$ and deterministically returns a tag $T \in \mathsf{TS}$: $T \leftarrow \text{Tag}(Key, \mathsf{K}, N, \mathsf{D})$;
- a tag updating algorithm Replace, that takes current and new finalization keys $Key, Key' \in \mathsf{FKS}$, current and new block keys for i-th block $K_i, K_i' \in \mathsf{BKS}$, current and new nonce values $N, N' \in \mathsf{NS}$, current and new block values for i-th block $D_i, D_i' \in \mathsf{BS}$ and a current tag T and deterministically returns an updated tag $T' \in \mathsf{TS}$: $T' \leftarrow \text{Replace}(Key, Key', K_i, K_i', N, N', D_i, D_i', T)$.

Algorithms IQRA.Tag and IQRA.Replace are defined as follows:

IQRA.Tag$(Key, \mathsf{K}, N, \mathsf{D})$

1 : $T \leftarrow (D_0 \oplus K_0) \cdot K_0 \oplus \ldots \oplus (D_{w-1} \oplus K_{w-1}) \cdot K_{w-1} \oplus \mathrm{F}(Key, N)$
2 : **return** T

IQRA.Replace$(Key, Key', K_i, K'_i, N, N', D_i, D'_i, T)$

1 : $T' \leftarrow T \oplus (D_i \oplus K_i) \cdot K_i \oplus (D'_i \oplus K'_i) \cdot K'_i \oplus \mathrm{F}(Key, N) \oplus \mathrm{F}(Key', N')$
2 : **return** T'

We do not define IQRA.Verify algorithm since it can be implemented via Tag invocation and comparing the result with the candidate tag. An arbitrary PRF could be used as F such as HMAC [9] or block cipher. We will denote the IQRA scheme based on particular function F as $\mathrm{IQRA}_{\mathrm{F}}$.

If block key is not changed (i.e. $K'_i = K_i$), Replace algorithm can be simplified. In this case it requires only one field multiplication instead of two because $(D_i \oplus K_i) \cdot K_i \oplus (D'_i \oplus K'_i) \cdot K'_i$ can be calculated as $(D_i \oplus D'_i) \cdot K_i$.

Let us discuss how the IQRA definition relates to the standard incremental family interface defined in [23, 5]. The core difference is the complex structure of the key space. To save henceforth the incrementality in case of key update, we separate the keys used for different purposes. Specifically, we identified two types of keys: keys for each block processing $\mathsf{K} = (K_0, \ldots, K_{w-1})$ and finalization key Key used for generating the mask value from nonce. Such separation allows to control the lifetime of each key, and thus update it independently from other keys right during the Replace algorithm work. The Replace interface supports such opportunity by taking as inputs not only current finalization key and key for the modifiable block, but also new key values. Current and new values could either coincide, or differ, as appropriate.

Other differences are not so principal and are justified by the interface simplification. Tag and Replace algorithms do not take as input a document identifier, since we consider our scheme only in the single-document setting. Moreover, the Replace algorithm needs only current value for modifiable block to update the tag and do not require the whole current document and the number of the modifiable block as inputs.

In the full disk encryption constructions blocks may be equal to the local tags of each sector. In this case the IQRA scheme provides the authenticity of the local tags.

Key derivation. The IQRA scheme uses $(w + 1)$ keys by design, however usually only one master key is available. We propose the way how to derive these keys from one $K_{master} \in \{0, 1\}^{\kappa}$ using the standard KDF function that

maps $\{0,1\}^\kappa \times \{0,1\}^* \times \mathbb{N}$ to $\{0,1\}^*$. Namely,

$$Key = \mathrm{KDF}(K_{master}, 0\mathrm{x}00 \| seed, k),$$

$$K_i = \mathrm{KDF}(K_{master}, 0\mathrm{x}01 \| \mathrm{str}_{L(w)}(i) \| S_i, n),\ i = 0 \ldots w-1.$$

Here $seed$ and $\mathsf{S} = (S_0, \ldots, S_{w-1})$ are key derivation parameters. If the key update should be performed, we could simply choose the new value for $seed$ or S_i parameter and compute the corresponding key value with KDF. Such key update mechanism is the instantiation of the external re-keying approach [37, 2]. Note that providing incremental properties requires K_{master} not to be changed during life-cycle of the IQRA scheme. Therefore, KDF parameters should be chosen in a such way that there would be no need to change K_{master} value either from combinatorial restrictions or from side-channel attacks.

We will denote the IQRA scheme with keys derived with such algorithm as [IQRA, KDF] scheme (see Figure 1).

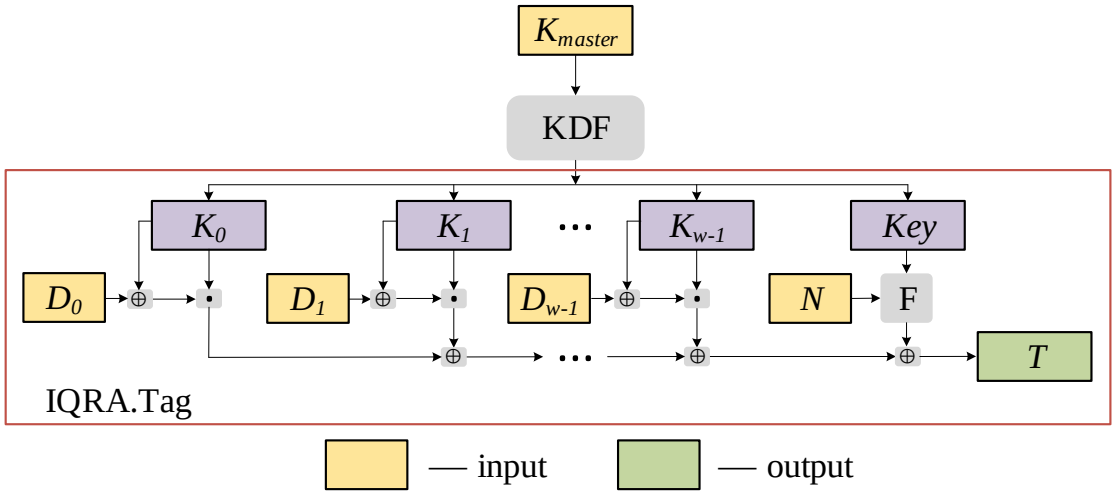

Figure 1: [IQRA, KDF] scheme

4 Security notions

4.1 Standard security notions

In this section we formally define basic security notions for cryptographic mechanisms used in the [IQRA_F, KDF] scheme.

Definition 1. *For function* $\mathrm{F} : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$:

$$\mathsf{Adv}_{\mathrm{F}}^{PRF}(\mathcal{A}) = \Pr\left[\mathbf{Exp}_{\mathrm{F}}^{PRF\text{-}1}(\mathcal{A}) \to 1\right] - \Pr\left[\mathbf{Exp}_{\mathrm{F}}^{PRF\text{-}0}(\mathcal{A}) \to 1\right],$$

where the experiments $\mathbf{Exp}_{\mathrm{F}}^{PRF\text{-}b}(\mathcal{A}), b \in \{0,1\}$*, are defined in the following way:*

$\mathbf{Exp}_{\mathrm{F}}^{PRF\text{-}b}(\mathcal{A})$

1 : **if** $b = 1$:
2 : $\quad K \xleftarrow{\mathcal{U}} \{0,1\}^k$
3 : **else** :
4 : $\quad \rho \xleftarrow{\mathcal{U}} Func(\{0,1\}^n, \{0,1\}^n)$
5 : $b' \xleftarrow{\$} \mathcal{A}^{F^b}(\,)$
6 : **return** b'

Oracle $F^b(m)$

1 : **if** $b = 1$:
2 : $\quad$ **return** $\mathrm{F}(K, m)$
3 : **else** :
4 : $\quad$ **return** $\rho(m)$

The PRP security notion is defined in the same way as PRF except that the uniform random function ρ is replaced by a uniform random permutation.

Definition 2. *For a variable-length output function* $\mathrm{KDF} : \{0,1\}^\kappa \times \{0,1\}^* \times \mathbb{N} \to \{0,1\}^*$:

$$\mathsf{Adv}_{\mathrm{KDF}}^{PRF^*}(\mathcal{A}) = \Pr\left[\mathbf{Exp}_{\mathrm{KDF}}^{PRF^*\text{-}1}(\mathcal{A}) \to 1\right] - \Pr\left[\mathbf{Exp}_{\mathrm{KDF}}^{PRF^*\text{-}0}(\mathcal{A}) \to 1\right],$$

where the experiments $\mathbf{Exp}_{\mathrm{KDF}}^{PRF^*\text{-}b}(\mathcal{A}), b \in \{0,1\}$*, are defined in the following way:*

$\mathbf{Exp}_{\mathrm{KDF}}^{PRF^*\text{-}b}(\mathcal{A})$

1 : **if** $b = 1$:
2 : $\quad K \xleftarrow{\mathcal{U}} \{0,1\}^\kappa$
3 : $Set \leftarrow \emptyset$
4 : $b' \xleftarrow{\$} \mathcal{A}^{KDF^b}(\,)$
5 : **return** b'

Oracle $KDF^b(s, l)$

1 : **if** $b = 1$:
2 : $\quad$ **return** $\mathrm{KDF}(K, s, l)$
3 : **else** :
4 : $\quad$ **if** $(s, l, \cdot) \in Set$ **then**
5 : $\quad\quad$ **return** $Set(s, l)$
6 : $\quad$ **else**
7 : $\quad\quad d \xleftarrow{\mathcal{U}} \{0,1\}^l$
8 : $\quad\quad Set \leftarrow Set \cup \{(s, l, d)\}$
9 : $\quad\quad$ **return** d

4.2 SUF-CSMA notion

We introduce SUF-CSMA notion (Strong Unforgeability under Chosen Settings and Message Attack) to analyze the security of $[\mathrm{IQRA}_{\mathrm{F}}, \mathrm{KDF}]$ scheme. It's a natural extension of standard SUF-CMA notion for nonce-based MACs [11] obtained by adding KDF calls.

Definition 3. *For* $[\mathrm{IQRA_F}, \mathrm{KDF}]$ *scheme*

$$\mathsf{Adv}^{\textit{SUF-CSMA}}_{[\mathrm{IQRA_F},\mathrm{KDF}]}(\mathcal{A}) = \Pr\left[\mathbf{Exp}^{\textit{SUF-CSMA}}_{[\mathrm{IQRA_F},\mathrm{KDF}]}(\mathcal{A}) \to 1\right],$$

where the experiment $\mathbf{Exp}^{\textit{SUF-CSMA}}_{[\mathrm{IQRA_F},\mathrm{KDF}]}(\mathcal{A})$ *is defined in the following way:*

$\mathbf{Exp}^{\textit{SUF-CSMA}}_{[\mathrm{IQRA_F},\mathrm{KDF}]}(\mathcal{A})$

1: $K_{master} \xleftarrow{\mathcal{U}} \{0,1\}^{\kappa}$
2: $GAMMA, STATES \leftarrow \emptyset,\ win \leftarrow 0$
3: $\mathcal{A}^{Tag, Verify}(\,)$
4: **return** win

$Verify(seed, \mathsf{S}, N, \mathsf{D}, T)$

1: $Key \leftarrow \mathrm{KDF}(K_{master}, 0\mathrm{x}00 \| seed, k)$
2: **for** $i = 0 \ldots w-1$ **do**
3: $\quad K_i \leftarrow \mathrm{KDF}(K_{master}, 0\mathrm{x}01 \| \mathrm{str}_{L(w)}(i) \| S_i, n)$
4: $\mathsf{K} \leftarrow (K_0, \ldots, K_{w-1})$
5: $T' \leftarrow \mathrm{IQRA.Tag}(Key, \mathsf{K}, N, \mathsf{D})$
6: $\mathsf{st} \leftarrow (seed, \mathsf{S}, N, \mathsf{D}, T)$
7: **if** $((T' = T) \wedge (\mathsf{st} \notin STATES))$ **then**
8: $\quad win \leftarrow 1$
9: **return** $(T' = T)$

$Tag(seed, \mathsf{S}, N, \mathsf{D})$

1: **if** $((seed, N) \in GAMMA)$ **then**
2: $\quad$ **return** $\perp$
3: $GAMMA \leftarrow GAMMA \cup \{(seed, N)\}$
4: $Key \leftarrow \mathrm{KDF}(K_{master}, 0\mathrm{x}00 \| seed, k)$
5: **for** $i = 0 \ldots w-1$ **do**
6: $\quad K_i \leftarrow \mathrm{KDF}(K_{master}, 0\mathrm{x}01 \| \mathrm{str}_{L(w)}(i) \| S_i, n)$
7: $\mathsf{K} \leftarrow (K_0, \ldots, K_{w-1})$
8: $T \leftarrow \mathrm{IQRA.Tag}(Key, \mathsf{K}, N, \mathsf{D})$
9: $\mathsf{st} \leftarrow (seed, \mathsf{S}, N, \mathsf{D}, T)$
10: $STATES \leftarrow STATES \cup \{\mathsf{st}\}$
11: **return** T

As usual, an adversary is given the access to *Tag* and *Verify* oracles, however it may control not only message and nonce, but also the parameters for key derivation. To win the adversary must make successful (the tag is correct) and non-trivial (the document was not previously tagged with the same parameters) query to *Verify* oracle.

Unlike standard models for nonce-based schemes we require $(seed, N)$ pair to be unique in the queries to *Tag* oracle. At the same time, $(seed, N)$ values in *Verify* queries may overlap with other queries. The uniqueness of $(seed, N)$ pairs can be provided in practice, for example, by storing them in the secure memory. Note that SUF-CMA model is a special case of SUF-CSMA model in which *seed* and S parameters are fixed.

Note that the modified IQRA scheme with the multilinear function instead of the quadratic multivariate polynomial is not SUF-CSMA-secure. Indeed, if the tag is calculated as $D_0 \cdot K_0 \oplus \ldots \oplus D_{w-1} \cdot K_{w-1} \oplus \mathrm{F}(Key, N)$ and some D_i is equal to zero block then the tag is independent of the S_i value. Therefore, an adversary can submit a valid forgery with the same document and tag, but with the different S_i value.

Incremental security. Traditionally security definitions for incremental MACs provide adversary with the access to *Update* oracle, because updated tags may differ from scratch-ones and allow forgery [13, 25, 23, 5]. However, for the Proposition 1 [5] states that sometimes update queries can be dropped if the function satisfies strong correctness. Strong correctness means that tags returned by the update algorithm are the same as if the updated document had instead been tagged directly, from scratch, via the tagging algorithm. Authors of [5] formulate their result for security model, where *Update* queries have to be made to the last copy of the document.

We extend this result for the case when *Update* queries have to be made for the authentic documents and tags (where MAC.Verify accepts). Specifically, we claim that for the $[\mathrm{IQRA}_{\mathrm{F}}, \mathrm{KDF}]$ scheme the proposed SUF-CSMA model is equivalent to iSUF-CSMA model defined similarly up to allowing the adversary access to the *Replace* oracle as defined in Figure 2.

$\mathit{Replace}(\mathit{seed}', S_i', N', D_i', \mathit{seed}, \mathsf{S}, N, \mathsf{D}, T)$

// Check the authenticity of the query
1 : $\mathit{Key} \leftarrow \mathrm{KDF}(K_{\mathit{master}}, 0\mathrm{x}00 \| \mathit{seed}, k)$
2 : **for** $i = 0 \ldots w-1$ **do**
3 : $\quad K_i \leftarrow \mathrm{KDF}(K_{\mathit{master}}, 0\mathrm{x}01 \| \mathrm{str}_{L(w)}(i) \| S_i, n)$
4 : $\mathsf{K} \leftarrow (K_0, \ldots, K_{w-1})$
5 : $T^* \leftarrow \mathrm{IQRA.Tag}(\mathit{Key}, \mathsf{K}, N, \mathsf{D})$
6 : **if** $(T^* \neq T)$ **then return** $\perp$
// Check uniqueness of (seed', N') pair
7 : **if** $((\mathit{seed}', N') \in \mathit{GAMMA})$ **then return** $\perp$
8 : $\mathit{GAMMA} \leftarrow \mathit{GAMMA} \cup \{(\mathit{seed}', N')\}$
// Update tag
9 : $\mathit{Key}' \leftarrow \mathrm{KDF}(K_{\mathit{master}}, 0\mathrm{x}00 \| \mathit{seed}', k)$
10 : $K_i' \leftarrow \mathrm{KDF}(K_{\mathit{master}}, 0\mathrm{x}01 \| \mathrm{str}_{L(w)}(i) \| S_i', n)$
11 : $T' \leftarrow \mathrm{IQRA.Replace}(\mathit{Key}, \mathit{Key}', K_i, K_i', N, N', D_i, D_i', T)$
12 : $\mathsf{S}' \leftarrow (S_0, \ldots, S_{i-1}, S_i', S_{i+1}, \ldots, S_{w-1})$
13 : $\mathsf{D}' \leftarrow (D_0, \ldots, D_{i-1}, D_i', D_{i+1}, \ldots, D_{w-1})$
14 : $\mathsf{st} \leftarrow (\mathit{seed}', \mathsf{S}', N', \mathsf{D}', T')$
15 : $\mathit{STATES} \leftarrow \mathit{STATES} \cup \{\mathsf{st}\}$
16 : **return** T'

Figure 2: The *Replace* oracle in the iSUF-CSMA notion

We consider *Replace* instead more general *Update* oracle, since the IQRA scheme is incremental regarding only replace operation. Replace operation can be applied only to authentic $(\mathit{seed}, \mathsf{S}, N, \mathsf{D}, T)$ set that is guaranteed by

checks in lines 1-6. The requirement of uniqueness of $(seed', N')$ pair remains for queries to the *Replace* oracle (see lines 7-8).

Lemma 1. *Let $\mathcal{A}$ be an adversary in the iSUF-CSMA model for* $[\mathrm{IQRA}, \mathrm{KDF}]$ *scheme, making at most q_{tag} queries to Tag oracle, q_{rep} queries to Replace oracle and q_{ver} queries to Verify oracle. Then there exists an adversary $\mathcal{B}$ with the same time complexity for* $[\mathrm{IQRA}, \mathrm{KDF}]$ *scheme in the SUF-CSMA model that makes at most $(q_{tag} + q_{rep})$ queries to Tag oracle and $(q_{rep} + q_{ver})$ queries to Verify oracle, such that:*

$$\mathsf{Adv}^{iSUF\text{-}CSMA}_{[\mathrm{IQRA},\mathrm{KDF}]}(\mathcal{A}) \leqslant \mathsf{Adv}^{SUF\text{-}CSMA}_{[\mathrm{IQRA},\mathrm{KDF}]}(\mathcal{B}).$$

We provide the formal proof of this lemma in Appendix B.1. The proof is based on the fact that the IQRA scheme is strongly correct and thus, the *Replace* oracle can be simulated via *Tag* and *Verify* oracles.

Lemma 1 allows us to analyze the security of $[\mathrm{IQRA}, \mathrm{KDF}]$ scheme in the standard setting and easily move the results to the incremental setting.

Relation to other notions. It is easy to see that iSUF-CSMA model is the extension of the IUF-BS model [23] up to KDF embedding and associated adversary control of key derivation parameters. Moreover, these models are exactly the same assuming *seed* and S parameters fixed in the queries. Thus, the security in SUF-CSMA model implies the security in IUF-BS model and the results of 5.1.3 [23] can be applied directly to identify the place of SUF-CSMA model among known security notions for incremental MACs.

5 Security bounds

In this section we provide the security bound for the $[\mathrm{IQRA}, \mathrm{KDF}]$ scheme in the SUF-CSMA model.

Theorem 1. *Let $\mathcal{A}$ be an adversary in the SUF-CSMA model for the* $[\mathrm{IQRA}_\mathrm{F}, \mathrm{KDF}]$ *scheme, making at most q_{tag} and q_{ver} queries to the Tag and Verify oracles respectively. Let assume that the number of distinct seed values in $\mathcal{A}$ queries is at most d and the number of queries with the same seed value is at most r. Then there exists an adversary $\mathcal{B}$ that breaks* KDF *scheme in PRF* model, making at most $(w+1)(q_{tag} + q_{ver})$ queries, and adversary $\mathcal{C}$ that breaks* F *in PRF model, making at most r queries, such that:*

$$\mathsf{Adv}^{SUF\text{-}CSMA}_{[\mathrm{IQRA}_\mathrm{F},\mathrm{KDF}]}(\mathcal{A}) \leqslant \mathsf{Adv}^{PRF^*}_{\mathrm{KDF}}(\mathcal{B}) + d \cdot \mathsf{Adv}^{PRF}_{\mathrm{F}}(\mathcal{C}) + \frac{q_{ver}}{2^{n-1}}.$$

Furthermore, the additional computational resources of $\mathcal{B}$ *and* $\mathcal{C}$ *are at most* $2c \cdot (q_{tag} + q_{ver}) \cdot T_{\text{IQRA}}$*, where* T_{IQRA} *is computational resources needed to calculate* IQRA.Tag*,* c *is a constant that depends only on a model of computation and a method of encoding.*

The idea behind the proof is as follows. We sequentially replace KDF and F functions with uniform random functions and then estimate the forgery probability for such idealized scheme. The full proof can be found in Appendix B.2.

Let us discuss the obtained bound. On the one hand, it does not depend on the amount of data processed using the same block keys $K_0, \ldots, K_{w-1}$. Specifically, the scheme remains secure in the SUF-CSMA model even when S parameter is fixed (i.e. tag is always computed with the same $K_0, \ldots, K_{w-1}$). However, the requirement to change block keys may follow from side-channel attack restrictions.

On the other hand, the obtained bound demonstrates the benefits from Key updating. $\mathsf{Adv}_{\text{F}}^{\text{PRF}}(\mathcal{C})$ value is usually of order $\frac{r^2}{2^n}$, where r is the number of queries made by $\mathcal{C}$. If the same amount of data is processed using the same Key, then r is equal to $\frac{q_{tag} + q_{ver}}{d}$ in the worst case. Thus, the term $d \cdot \mathsf{Adv}_{\text{F}}^{\text{PRF}}(\mathcal{C})$ is of order $\frac{(q_{tag} + q_{ver})^2}{d \cdot 2^n}$ and decreases with d growth. As a result, the best bound is achieved if each tag is computed with the new Key.

Applying Theorem 1 straightforward to the case when block cipher E is used as the underlying PRF leads to appearance $d \cdot \mathsf{Adv}_{\text{E}}^{\text{PRF}}(\mathcal{C})$ term in the bound. This term is equal to $d \cdot \mathsf{Adv}_{\text{E}}^{\text{PRP}}(\mathcal{C}) + \frac{dr(r-1)}{2^{n+1}}$ according to the PRP/PRF switching lemma [16] and thus the bound degenerates as r is of order $2^{n/2}$. However, this bound can be improved using Bernstein results [18].

Corollary 1. *Let* $\mathcal{A}$ *be an adversary in the SUF-CSMA model for the* $[\text{IQRA}_{\text{E}}, \text{KDF}]$ *scheme, making at most* q_{tag} *and* q_{ver} *queries to the Tag and Verify oracles respectively. Let assume that the number of distinct seed values in* $\mathcal{A}$ *queries is at most* d *and the number of distinct nonces queried with each seed value is exactly* $r_1, \ldots, r_d$*,* $r = \max_{1 \leqslant i \leqslant d} r_i$*. Then there exists an adversary* $\mathcal{B}$ *that breaks* KDF *scheme in PRF* model, making at most* $(w+1)(q_{tag} + q_{ver})$ *queries, and adversary* $\mathcal{C}$ *that breaks* E *in PRP model, making at most* r *queries, such that:*

$$\mathsf{Adv}^{\textit{SUF-CSMA}}_{[\text{IQRA}_{\text{E}}, \text{KDF}]}(\mathcal{A}) \leqslant \mathsf{Adv}^{\textit{PRF}^*}_{\text{KDF}}(\mathcal{B}) + d \cdot \mathsf{Adv}^{\textit{PRP}}_{\text{E}}(\mathcal{C}) + \delta_n(r_1) \cdot \ldots \cdot \delta_n(r_d) \cdot \frac{q_{ver}}{2^{n-1}},$$

where $\delta_n(q) = \left(1 - \frac{q-1}{2^n}\right)^{-q/2}$. *Furthermore, the additional computational resources of* $\mathcal{B}$ *and* $\mathcal{C}$ *are at most* $2c \cdot (q_{tag} + q_{ver}) \cdot T_{\text{IQRA}}$, *where* T_{IQRA} *is computational resources needed to calculate* IQRA.Tag, c *is a constant that depends only on a model of computation and a method of encoding.*

The proof of the Corollary is similar to the Theorem 1 proof except that the block cipher is firstly replaced with uniform random permutation and the transition to uniform random function is done using the extension of the Bernshtein bound [18], Theorem 2.1, on case of multiple oracle access. The full proof can be found in Appendix B.3.

In some cases the following observation can be useful:

$$\delta_n(r_1) \cdot \delta_n(r_2) \cdot \dots \cdot \delta_n(r_d) \leqslant \delta_n(r_1 + r_2 + \ldots + r_d) \leqslant \delta_n(q_{tag} + q_{ver}).$$

Let's make sure that the obtained bound is better than the one derived from Theorem 1. This is most evident in the case when number of queries is beyond the birthday bound. Let $n = 128, d = 2, r = \dfrac{q_{tag} + q_{ver}}{2} = 2^{68}$. Then the $\dfrac{dr(r-1)}{2^{n+1}}$ term in the Theorem 1 bound degenerates while in fact the scheme is still secure: $\delta_{128}(2^{69}) \leqslant 513$, so

$$\delta_n(r_1)\delta_n(r_2)\frac{q_{ver}}{2^{n-1}} \leqslant \delta_n(q_{tag} + q_{ver})\frac{q_{ver}}{2^{n-1}} = \delta_{128}(2^{69})\frac{q_{ver}}{2^{127}} \leqslant \frac{q_{ver}}{2^{117}}.$$

Let us also show that the Corollary bound is sensitive to the Key update. Suppose that $n = 64$, $q_{tag} + q_{ver} = 2^{64}$ and consider two extreme cases.

If re-keying is not performed, i.e. $d = 1$, then $r_1 = 2^{64}$ in the worst case. The $\delta_n(r_1)$ value is equal to

$$\left(1 - \frac{2^{64} - 1}{2^{64}}\right)^{-2^{63}} = \left(\frac{1}{2^{64}}\right)^{-2^{63}} \gg 2^{64}$$

and thus, the bound totally degenerates.

If re-keying is performed with e.g. $d = 2^{63}, r_1 = \dots = r_d = 2$, by the Corollary 1 the scheme remains secure:

$$\delta_{64}(2)^{2^{63}} \cdot \frac{q_{ver}}{2^{63}} = \left(1 - \frac{1}{2^{64}}\right)^{-2^{63}} \cdot \frac{q_{ver}}{2^{63}} \leqslant \frac{q_{ver}}{2^{62}}.$$

6 Conclusion

This paper introduces the IQRA scheme that is new incremental re-keying friendly MAC scheme. We propose the way, how to combine the IQRA

scheme with external re-keying mechanism, and introduce SUF-CSMA security model for its analysis. The obtained security bound allow us to estimate the security of the [IQRA, KDF] scheme by the security of the used cryptographic primitives (PRF and KDF).

The direction for further research is the analysis of [IQRA, KDF] scheme in the stronger security notions: either incremental, where an adversary has a capability to apply replace operation to the non-authentic tag [23], or standard, that takes into account the adversary capability to perform side-channel attacks. Moreover, as mentioned above, we are going to examine incrementality of [IQRA, KDF] scheme regarding other modification operations.

References

[1] Akhmetzyanova L., Alekseev E., Oshkin I., Smyshlyaev S., Sonina L., "On the properties of the CTR encryption mode of the Magma and Kuznyechik block ciphers with re-keying method based on CryptoPro Key Meshing", *Matem. Vopr. Kriptogr*, **8**, 2 (2017), 39–50.

[2] Akhmetzyanova L., Alekseev E., Oshkin I., Smyshlyaev S., "On Internal Re-keying", *LNCS*, International Conference on Research in Security Standardisation, **12529**, Springer, Cham, 2020, 23–45.

[3] Akhmetzyanova L., Alekseev E., Sedov G., Smyshlyaeva E., Smyshlyaev S., "Practical significance of security bounds for standardized internally re-keyed block cipher modes", *Matem. Vopr. Kriptogr.*, **10**, 2 (2019), 31–46.

[4] Alekseev E., Akhmetzyanova L., Zubkov A., Karpunin G., Smyshlyaev S., "On one approach to formalizing cryptographic analysis tasks", to be published (in Russian), *Matem. Vopr. Kriptogr.*, 2021.

[5] Arte V., Bellare M., Khati L., "Incremental Cryptography Revisited: PRFs, Nonces and Modular Design", International Conference on Cryptology in India, 2020, 576–598.

[6] Atighehchi K., Muntean T., "Towards fully incremental cryptographic schemes", Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security, 2013, 505–510.

[7] Atighehchi K., "Space-efficient, byte-wise incremental and perfectly private encryption schemes", 2014, 104.

[8] Bellare M., Canetti R., Krawczyk H., "Pseudorandom functions revisited: The cascade construction and its concrete security", Proceedings of 37th Conference on Foundations of Computer Science, 1996, 514–523.

[9] Bellare M., Canetti R., Krawczyk H., "Keying hash functions for message authentication", Annual international cryptology conference, 1996, 1–15.

[10] Bellare, M., Desai, A., Jokipii, E., Rogaway, P., "A concrete security treatment of symmetric encryption", In Proceedings of 38th Annual Symposium on Foundations of Computer Science (FOCS'97), 1997, 394–403.

[11] Bellare M., Namprempre C., "Authenticated encryption: Relations among notions and analysis of the generic composition paradigm", International Conference on the Theory and Application of Cryptology and Information Security, 2000, 531–545.

[12] Bellare M., Goldreich O., Goldwasser S., "Incremental cryptography: The case of hashing and signing", Annual International Cryptology Conference, 1994, 216–233.

[13] Bellare M., Goldreich O., Goldwasser S., "Incremental cryptography and application to virus protection", Proceedings of the twenty-seventh annual ACM symposium on Theory of computing, 1995, 45–56.

[14] Bellare M., Guérin R., Rogaway P., "XOR MACs: New methods for message authentication using finite pseudorandom functions", Annual International Cryptology Conference, 1995, 15–28.

[15] Bellare M., Micciancio D., "A new paradigm for collision-free hashing: Incrementality at reduced cost", International Conference on the Theory and Applications of Cryptographic Techniques, 1997, 163–192.

[16] Bellare M., Rogaway P., "Introduction to modern cryptography", Ucsd Cse, 2005, 207.

[17] Bellare M., Rogaway P., "The security of triple encryption and a framework for code-based game-playing proofs", Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2006, 409–426..

[18] Bernstein D., "Stronger security bounds for permutations", 2005.

[19] Bernstein D., "Stronger security bounds for Wegman-Carter-Shoup authenticators", Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2005, 164–180.

[20] Black J., Rogaway P., "A block-cipher mode of operation for parallelizable message authentication", International Conference on the Theory and Applications of Cryptographic Techniques, 2002, 384–397.

[21] Buonanno E., Katz J., Yung M., "Incremental unforgeable encryption.", *LNCS*, Fast Software Encryption – FSE 2001, **2355**, ed. Matsui M., Springer, Berlin, Heidelberg, 2002, 109–124.

[22] Itani W., Kayssi A., Chehab A., "Energy-efficient incremental integrity for securing storage in mobile cloud computing", In 2010 International Conference on Energy Aware Computing, 2010, 1–2.

[23] Khati L., *Full disk encryption and beyond*, Diss. Université Paris sciences et lettres, 2019.

[24] Khati L., Vergnaud D., "Analysis and improvement of an authentication scheme in incremental cryptography", *LNCS*, International Conference on Selected Areas in Cryptography, ed. In Carlos Cid and Michael J. Jacobson Jr, Springer, Heidelberg, August, 2019, 50–70.

[25] Fischlin M., "Incremental cryptography and memory checkers", *LNCS*, Advances in Cryptology – EUROCRYPT'97, **1233**, ed. Walter Fumy, Springer, Berlin, Heidelberg, 1997, 293–408.

[26] Fischlin M., "Lower bounds for the signature size of incremental schemes", 38th Annual Symposium on Foundations of Computer Science, 1997, 438–447.

[27] Bok-Min G., Siddiqi M. U., Hean-Teik C., "Incremental hash function based on pair chaining & modular arithmetic combining", International Conference in Cryptology in India, 2001, 50–61.

[28] Hart J. K., Martinez K., "Environmental sensor networks: A revolution in the earth system science?", Earth-Science Reviews, 2006, 177-–191.

[29] Iwata T., Minematsu K., Peyrin T., and Seurin Y., "ZMAC: A fast tweakable block cipher mode for highly secure message authentication", *LNCS*, ed. In Jonathan Katz and Hovav Shacham, CRYPTO 2017, Springer, Heidelberg, 2017, 34-–65.

[30] Krawczyk H., "Cryptographic extraction and key derivation: The HKDF scheme", Annual Cryptology Conference, 2010, 631–648.

[31] Mihajloska H., Gligoroski D., Samardjiska S., "Reviving the idea of incremental cryptography for the zettabyte era use case: Incremental hash functions based on SHA-3", International Workshop on Open Problems in Network Security, 2015, 97–111.

[32] Mironov, Ilya and Pandey, Omkant and Reingold, Omer and Segev, Gil, "Incremental deterministic public-key encryption", Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2012, 628–644.

[33] Micciancio D., "Oblivious data structures: Applications to cryptography", In 29th Annual ACM Symposium on Theory of Computing, 1997, 456–464.

[34] David A. McGrew and John Viega., "The security and performance of the Galois/counter mode (GCM) of operation", *LNCS*, INDOCRYPT 2004, **3348**, ed. Canteaut A., Viswanathan K., Springer, Heidelberg, 2004, 343–355.

[35] Peyrin T., and Seurin Y., "Counter-in-tweak: Authenticated encryption modes for tweakable block ciphers", *LNCS*, CRYPTO 2016, **9814**, ed. Robshaw M., Katz J., Springer, Heidelberg, 2016, 33–63.

[36] Sasaki Y., Yasuda K., "A new mode of operation for incremental authenticated encryption with associated data", *LNCS*, SAC 2015: 22nd Annual International Workshop on Selected Areas in Cryptography, **9566**, ed. Dunkelman O., Keliher L., Springer, Berlin, Heidelberg, 2016, 397—416.

[37] Smyshlyaev S., *Re-keying Mechanisms for Symmetric Keys*, RFC 8645, DOI 10.17487/RFC8645, 2019, https://www.rfc-editor.org/info/rfc8645.

[38] Standaert FX., "Introduction to Side-Channel Attacks", Secure Integrated Circuits and Systems. Integrated Circuits and Systems, 2010, 27–42.

A Additional security notions

Let us define mu-PRF and mu-PRP security notions with parameter d which will be used in the follow-up proofs.

Definition 4. *For a function* $\mathrm{F} : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$

$$\mathsf{Adv}_{\mathrm{F}}^{mu\text{-}PRF}(\mathcal{A}) = \Pr\left[\mathbf{Exp}_{\mathrm{F}}^{mu\text{-}PRF\text{-}1}(\mathcal{A}) \to 1\right] - \Pr\left[\mathbf{Exp}_{\mathrm{F}}^{mu\text{-}PRF\text{-}0}(\mathcal{A}) \to 1\right],$$

where experiments $\mathbf{Exp}_{\mathrm{F}}^{mu\text{-}PRF\text{-}b}(\mathcal{A}), b \in \{0,1\}$, *are defined in the following way:*

$\mathbf{Exp}_{\mathrm{F}}^{mu\text{-}PRF\text{-}b}(\mathcal{A})$	*Oracle* $F^b(i,m)$
1: **for** $i = 1 \dots d$ **do**	1: **if** $b = 1$ **then**
2: **if** $b = 1$ **then**	2: **return** $\mathrm{F}(Key_i, m)$
3: $Key_i \xleftarrow{\mathcal{U}} \{0,1\}^k$	3: **else**
4: **else**	4: **return** $\rho_i(m)$
5: $\rho_i \xleftarrow{\mathcal{U}} Func(\{0,1\}^n, \{0,1\}^n)$	
6: $b' \xleftarrow{\$} \mathcal{A}^{F^b}(\,)$	
7: **return** b'	

The mu-PRP security notion is defined in the same way as mu-PRF except that the random functions ρ_i are replaced by random permutations.

The following relation between mu-PRF and PRF security notions takes place [8]. The relation between mu-PRP and PRP notions is exactly the same.

Lemma 2. *Let* $\mathcal{D}$ *be an adversary for* F *scheme in the mu-PRF model with parameter* d*, making at most* q *queries to the* F^b *oracle with at most* r *different* m *values for fixed* i *value. Then there exists an adversary* $\mathcal{C}$ *for* F *scheme in the PRF model that makes at most* r *queries to its own oracle, such that:*

$$\mathsf{Adv}_{F}^{mu\text{-}PRF}(\mathcal{D}) \leqslant d \cdot \mathsf{Adv}_{F}^{PRF}(\mathcal{C}).$$

Furthermore, $\mathcal{C}$ *needs at most* $c \cdot (dk + q \cdot (T_{\mathrm{F}} + 2n \log q))$ *additional computational resources, where* T_{F} *is computational resources needed to calculate function* F*,* c *is a constant that depends only on a model of computation and a method of encoding.*

This lemma is presented in [8] (lemma 3.3), but with the difference that adversary $\mathcal{C}$ in [8] makes at most q queries. See the next sentence in proof of lemma 3.3: «In the worst case D_1 still has to make $q_1 = q_2$ queries to its oracle g (although on the average it is $\frac{q_2}{m}$)». Here we provide more accurate estimation of the number of queries made by $\mathcal{C}$ bounded it by the maximal number of $\mathcal{D}$ queries with the same i value.

B Security proofs

B.1 Lemma 1 proof

Proof. We fix $\mathcal{A}$ – the adversary that makes forgery for the [IQRA, KDF] scheme in the iSUF-CSMA model. The adversary $\mathcal{A}$ has the access to the oracles *Tag*, *Replace* and *Verify* making at most q_{tag}, q_{rep} and q_{ver} queries respectively. Let construct an adversary $\mathcal{B}$ that breaks [IQRA, KDF] scheme in the SUF-CSMA model and uses adversary $\mathcal{A}$ as a «black box». $\mathcal{B}$ has access to its own oracles Tag^* and $Verify^*$.

The adversary $\mathcal{B}$ works at follows. It runs $\mathcal{A}$ and simulates *Tag* and *Verify* oracles just translating the queries to its own oracles. It simulates *Replace* oracle by the following procedure.

$SimReplace(seed', S'_i, N', D'_i, seed, \mathsf{S}, N, \mathsf{D}, T)$

1: $res \leftarrow Verify^*(seed, \mathsf{S}, N, \mathsf{D}, T)$
2: **if** $(res = 0)$ **then return** $\perp$
3: $\mathsf{S}' \leftarrow (S_0, \ldots, S_{i-1}, S'_i, S_{i+1}, \ldots, S_{w-1})$
4: $\mathsf{D}' \leftarrow (D_0, \ldots, D_{i-1}, D'_i, D_{i+1}, \ldots, D_{w-1})$
5: $T' \leftarrow Tag^*(seed', \mathsf{S}', N', \mathsf{D}')$
6: **return** T'

SimReplace works exactly the same as original *Replace* oracle since the IQRA scheme is strongly correct: replace operation leads to the same result as if the tag was computed directly, from scratch.

By construction the adversary $\mathcal{B}$ makes at most $(q_{tag} + q_{rep})$ queries to Tag^* oracle and at most $(q_{ver} + q_{rep})$ queries to $Verify^*$ oracle. Let estimate the probability of $\mathcal{B}$ success.

$\mathcal{B}$ wins if at least one of two events takes place: $\mathcal{A}$ makes query to *Verify* oracle that is correct and non-trivial (event_1) or $\mathcal{A}$ makes query to *Replace* oracle such that $(seed, \mathsf{S}, N, \mathsf{D}, T)$ set forms non-trivial forgery (event_2). Furthermore, the probability of event_1 is equal to $\Pr\left[\mathbf{Exp}^{\text{iSUF-CSMA}}_{[\text{IQRA,KDF}]}(\mathcal{A}) \to 1\right]$

by definition of iSUF-CSMA model. Thus

$$\Pr\left[\mathbf{Exp}^{\text{SUF-CSMA}}_{[\text{IQRA,KDF}]}(\mathcal{B}) \to 1\right] = \Pr[\mathsf{event}_1 \vee \mathsf{event}_2] \geq$$
$$\geq \Pr[\mathsf{event}_1] = \Pr\left[\mathbf{Exp}^{\text{iSUF-CSMA}}_{[\text{IQRA,KDF}]}(\mathcal{A}) \to 1\right].$$

□

B.2 Theorem 1 proof

Proof. Let $\mathbf{Exp}^0$ denote the original security experiment as defined in the SUF-CSMA security model definition (see Definition 3). We fix $\mathcal{A}$ – the adversary that makes forgery for the [IQRA, KDF] scheme in the SUF-CSMA model. The adversary has the access to the oracles *Tag* and *Verify*. We assume that adversary can make at most q_{tag} queries to the oracle *Tag* and q_{ver} queries to the oracle *Verify*.

Our goal is to upper-bound $\Pr\left[\mathbf{Exp}^{\text{SUF-CSMA}}_{\text{IQRA}}(\mathcal{A}) \to 1\right] = \Pr\left[\mathbf{Exp}^0(\mathcal{A}) \to 1\right]$.

Construction of adversary $\mathcal{B}$. $\mathbf{Exp}^1$ is the modification of the $\mathbf{Exp}^0$ obtained by implementing function KDF as a uniform random function using «lazy sampling» (see Figure 3). Here and after we denote the difference between experiments by color in pseudocode.

The idea is to «open» new pairs $(seed, Key)$ and triplets (i, S_i, K_i) as soon as the adversary makes the corresponding queries. We store $(seed, Key)$ pairs in Π_1 set and (i, S_i, K_i) triplets in Π_2 set. We can choose and store these values independently since there are distinct prefixes 0x00 and 0x01 for Key and K_i derivation. Moreover, for each block we choose block keys independently from each over, because $\text{str}_{L(w)}(i)$ guarantees that KDF inputs for different block numbers do not intersect.

If $(\alpha, \beta) \in \Pi_1$, we denote β as $\Pi_1(\alpha)$, we write $(\alpha, \cdot) \in \Pi_1$ shorthand for the condition that there exists β such that $(\alpha, \beta) \in \Pi_1$. If $(\alpha, \gamma, \beta) \in \Pi_2$, we denote β as $\Pi_2(\alpha, \gamma)$, we write $(\alpha, \gamma, \cdot) \in \Pi_2$ shorthand for the condition that there exists β such that $(\alpha, \gamma, \beta) \in \Pi_2$.

Let estimate the difference between $\mathbf{Exp}^0$ and $\mathbf{Exp}^1$. We construct the adversary $\mathcal{B}$ that breaks the KDF in PRF* security model. The adversary $\mathcal{B}$ has the access to its own oracle KDF^b, $b \in \{0, 1\}$. $\mathcal{B}$ invokes the adversary $\mathcal{A}$ as a subroutine and simulates *Tag* and *Verify* oracles for $\mathcal{A}$ as in $\mathbf{Exp}^0$ replacing the KDF calls by calls to his own oracle KDF^b. The adversary $\mathcal{B}$ returns 1 to its own challenger, if adversary $\mathcal{A}$ makes a successful forgery. If

$b = 1$ then $\mathcal{B}$ implements for $\mathcal{A}$ exactly $\mathbf{Exp}^0$. If $b = 0$ then $\mathcal{B}$ implements exactly $\mathbf{Exp}^1$. Thus,

$$\Pr\left[\mathbf{Exp}^0(\mathcal{A}) \to 1\right] - \Pr\left[\mathbf{Exp}^1(\mathcal{A}) \to 1\right] =$$
$$= \Pr\left[\mathbf{Exp}_{\mathrm{KDF}}^{\mathrm{PRF}^*\text{-}1}(\mathcal{B}) \to 1\right] - \Pr\left[\mathbf{Exp}_{\mathrm{KDF}}^{\mathrm{PRF}^*\text{-}0}(\mathcal{B}) \to 1\right] = \mathsf{Adv}_{\mathrm{KDF}}^{\mathrm{PRF}^*}(\mathcal{B}).$$

Processing each $\mathcal{A}$ query leads to $(w+1)$ query to KDF^b oracle. Thus, the adversary $\mathcal{B}$ makes at most $(w+1)(q_{tag}+q_{ver})$ queries to its own oracle. By construction, $\mathcal{B}$ needs $O((q_{tag}+q_{ver})T_{\mathrm{IQRA}})$ additional computational resources for simulating *Tag* and *Verify* work for $\mathcal{A}$. Here T_{IQRA} is computational resources needed to calculate function IQRA.Tag.

$\mathbf{Exp}^1(\mathcal{A})$

$\Pi_1, \Pi_2 \leftarrow \emptyset$
$GAMMA, STATES \leftarrow \emptyset,\ win \leftarrow 0$
$\mathcal{A}^{Tag, Verify}(\,)$
return win

Verify$(seed, \mathsf{S}, N, \mathsf{D}, T)$

if $((seed, \cdot) \in \Pi_1)$ **then**
  $Key \leftarrow \Pi_1(seed)$
else
  $Key \xleftarrow{\mathcal{U}} \{0,1\}^k,\ \Pi_1 \leftarrow \Pi_1 \cup \{(seed, Key)\}$
for $i = 0 \ldots w-1$ **do**
  if $((i, S_i, \cdot) \in \Pi_2)$ **then**
    $K_i \leftarrow \Pi_2(i, S_i)$
  else
    $K_i \xleftarrow{\mathcal{U}} \{0,1\}^n,\ \Pi_2 \leftarrow \Pi_2 \cup \{(i, S_i, K_i)\}$
$\mathsf{K} \leftarrow (K_0, \ldots, K_{w-1})$
$T' \leftarrow \mathrm{IQRA.Tag}(Key, \mathsf{K}, N, \mathsf{D})$
$\mathsf{st} \leftarrow (seed, \mathsf{S}, N, \mathsf{D}, T)$
if $((T' = T) \wedge (\mathsf{st} \notin STATES))$ **then**
  $win \leftarrow 1$
return $(T' = T)$

Tag$(seed, \mathsf{S}, N, \mathsf{D})$

if $((seed, N) \in GAMMA)$ **then**
  return $\perp$
$GAMMA \leftarrow GAMMA \cup \{(seed, N)\}$
if $((seed, \cdot) \in \Pi_1)$ **then**
  $Key \leftarrow \Pi_1(seed)$
else
  $Key \xleftarrow{\mathcal{U}} \{0,1\}^k,\ \Pi_1 \leftarrow \Pi_1 \cup \{(seed, Key)\}$
for $i = 0 \ldots w-1$ **do**
  if $((i, S_i, \cdot) \in \Pi_2)$ **then**
    $K_i \leftarrow \Pi_2(i, S_i)$
  else
    $K_i \xleftarrow{\mathcal{U}} \{0,1\}^n,\ \Pi_2 \leftarrow \Pi_2 \cup \{(i, S_i, K_i)\}$
$\mathsf{K} \leftarrow (K_0, \ldots, K_{w-1})$
$T \leftarrow \mathrm{IQRA.Tag}(Key, \mathsf{K}, N, \mathsf{D})$
$\mathsf{st} \leftarrow (seed, \mathsf{S}, N, \mathsf{D}, T)$
$STATES \leftarrow STATES \cup \{\mathsf{st}\}$
return T

Figure 3: The $\mathbf{Exp}^1$ for the adversary $\mathcal{A}$.

Construction of adversary $\mathcal{C}$. Consider the experiment $\mathbf{Exp}^{1'}$ (see Figure 4). It is the modification of the $\mathbf{Exp}^1$ in the following way: d Key values are

sampled during experiment initializing phase and *Tag* and *Verify* oracles just use them one by one responding to the queries. The Π_1 set in $\mathbf{Exp}^{1'}$ contains the pairs $(seed, ctr)$, where ctr is an index of Key that corresponds to the value $seed$. By theorem condition, maximal number of distinct $seed$ values in $\mathcal{A}$ queries is at most d, therefore such modification will be inobservable for $\mathcal{A}$:

$$\Pr\left[\mathbf{Exp}^{1'}(\mathcal{A}) \to 1\right] = \Pr\left[\mathbf{Exp}^{1}(\mathcal{A}) \to 1\right].$$

$\mathbf{Exp}^{1'}(\mathcal{A})$

$\Pi_1, \Pi_2 \leftarrow \emptyset$
$GAMMA, STATES \leftarrow \emptyset,\ win \leftarrow 0$
$ctr \leftarrow 0$
for $i = 1...d$: **do**
  $Key_i \xleftarrow{\mathcal{U}} \{0,1\}^k$
$\mathcal{A}^{Tag, Verify}(\,)$
return win

Verify$(seed, \mathsf{S}, N, \mathsf{D}, T)$

if $((seed, \cdot) \in \Pi_1)$ **then**
  $Key \leftarrow Key_{\Pi_1(seed)}$
else
  $ctr \leftarrow ctr + 1$
  $Key \leftarrow Key_{ctr}$
  $\Pi_1 \leftarrow \Pi_1 \cup \{(seed, ctr)\}$
for $i = 0 \ldots w-1$ **do**
  if $((i, S_i, \cdot) \in \Pi_2)$ **then**
    $K_i \leftarrow \Pi_2(i, S_i)$
  else
    $K_i \xleftarrow{\mathcal{U}} \{0,1\}^n,\ \Pi_2 \leftarrow \Pi_2 \cup \{(i, S_i, K_i)\}$
$\mathsf{K} \leftarrow (K_0, \ldots, K_{w-1})$
$T' \leftarrow \text{IQRA.Tag}(Key, \mathsf{K}, N, \mathsf{D})$
$\mathsf{st} \leftarrow (seed, \mathsf{S}, N, \mathsf{D}, T)$
if $((T' = T) \wedge (\mathsf{st} \notin STATES))$ **then**
  $win \leftarrow 1$
return $(T' = T)$

Tag$(seed, \mathsf{S}, N, \mathsf{D})$

if $((seed, N) \in GAMMA)$ **then**
  return $\perp$
$GAMMA \leftarrow GAMMA \cup \{(seed, N)\}$
if $((seed, \cdot) \in \Pi_1)$ **then**
  $Key \leftarrow Key_{\Pi_1(seed)}$
else
  $ctr \leftarrow ctr + 1$
  $Key \leftarrow Key_{ctr}$
  $\Pi_1 \leftarrow \Pi_1 \cup \{(seed, ctr)\}$
for $i = 0 \ldots w-1$ **do**
  if $((i, S_i, \cdot) \in \Pi_2)$ **then**
    $K_i \leftarrow \Pi_2(i, S_i)$
  else
    $K_i \xleftarrow{\mathcal{U}} \{0,1\}^n,\ \Pi_2 \leftarrow \Pi_2 \cup \{(i, S_i, K_i)\}$
$\mathsf{K} \leftarrow (K_0, \ldots, K_{w-1})$
$T \leftarrow \text{IQRA.Tag}(Key, \mathsf{K}, N, \mathsf{D})$
$\mathsf{st} \leftarrow (seed, \mathsf{S}, N, \mathsf{D}, T)$
$STATES \leftarrow \{\mathsf{st}\}$
return tag

Figure 4: The $\mathbf{Exp}^{1'}$ for the adversary $\mathcal{A}$.

$\mathbf{Exp}^{2}$ (see Figure 5) is modification of the $\mathbf{Exp}^{1'}$ in the following way. We replace function F with different finalization keys Key_i by the set of uniform

random functions ρ_i. Just like the keys Key_i in $\mathbf{Exp}^{1'}$, d functions ρ_i in $\mathbf{Exp}^2$ refered to distinct *seed* values are sampled during experiment initializing phase and *Tag* and *Verify* oracles just use them one by one responding to the queries.

Let estimate the difference between $\mathbf{Exp}^{1'}$ and $\mathbf{Exp}^2$. We construct an adversary $\mathcal{D}$ that breaks F in the mu-PRF model with parameter d. The adversary $\mathcal{D}$ has the access to its own oracle $F^b,\ b \in \{0,1\}$. $\mathcal{D}$ invokes the adversary $\mathcal{A}$ as a subroutine and simulates *Tag* and *Verify* oracles for $\mathcal{A}$ as in $\mathbf{Exp}^{1'}$ replacing F calls by calls to his own oracle F^b with seed serial number $\Pi_1(seed)$ and N as the arguments. The adversary $\mathcal{D}$ returns 1 to its own challenger, if adversary $\mathcal{A}$ makes a successful forgery. If $b = 1$ then $\mathcal{D}$ implements exactly $\mathbf{Exp}^{1'}$ for $\mathcal{A}$. If $b = 0$ then $\mathcal{D}$ implements exactly $\mathbf{Exp}^2$. Thus,

$$\Pr\Big[\mathbf{Exp}^{1'}(\mathcal{A}) \to 1\Big] - \Pr\big[\mathbf{Exp}^{2}(\mathcal{A}) \to 1\big] =$$
$$= \Pr\big[\mathbf{Exp}_{\mathrm{F}}^{\text{mu-PRF-1}}(\mathcal{D}) \to 1\big] - \Pr\big[\mathbf{Exp}_{\mathrm{F}}^{\text{mu-PRF-0}}(\mathcal{D}) \to 1\big] = \mathsf{Adv}_{\mathrm{F}}^{\text{mu-PRF}}(\mathcal{D}).$$

By construction, the adversary $\mathcal{D}$ makes at most $(q_{tag} + q_{ver})$ queries to its own F^b oracle, moreover the number of queries with the same $i = \Pi_1(seed)$ value is at most r. Adversary $\mathcal{D}$ needs at most $O((q_{tag}+q_{ver})T_{\mathrm{IQRA}})$ additional computional resources for simulating *Tag* and *Verify* work for $\mathcal{A}$. We apply Lemma 2 to estimate the $\mathsf{Adv}_{\mathrm{F}}^{\text{mu-PRF}}(\mathcal{D})$. By this lemma there exists an adversary $\mathcal{C}$ for function F in PRF model, such that

$$\mathsf{Adv}_{\mathrm{F}}^{\text{mu-PRF}}(\mathcal{D}) \leqslant d \cdot \mathsf{Adv}_{\mathrm{F}}^{\text{PRF}}(\mathcal{C}),$$

where d is the number of distinct *seed* values across $\mathcal{A}$ queries. Furthermore, adversary $\mathcal{C}$ makes at most r queries to his own oracle and needs at most $O(dk + (q_{tag} + q_{ver}) \cdot (T_{\mathrm{F}} + 2n\log(q_{tag} + q_{ver})))$ additional computational resources compared to $\mathcal{D}$, where T_{F} is computational resources needed to calculate function F, that can be upper bounded by $O((q_{tag} + q_{ver})T_{\mathrm{IQRA}})$. So the time complexity of $\mathcal{C}$ is at most $T + O((q_{tag} + q_{ver}) \cdot 2T_{\mathrm{IQRA}})$.

Summing up,

$$\Pr\Big[\mathbf{Exp}^{1'}(\mathcal{A}) \to 1\Big] - \Pr\big[\mathbf{Exp}^{2}(\mathcal{A}) \to 1\big] \leqslant d \cdot \mathsf{Adv}_{\mathrm{F}}^{\text{PRF}}(\mathcal{C}).$$

Probability of winning in $\mathbf{Exp}^2$. Let estimate the probability of $\mathcal{A}$ success in $\mathbf{Exp}^2$.

Firstly, let consider $q_{ver} = 1$ case. This means that $\mathcal{A}$ has only one attempt to make a forgery. We denote the z-th query to *Tag* oracle as

Exp$^2(\mathcal{A})$

$\Pi_1, \Pi_2 \leftarrow \emptyset$
$GAMMA, STATES \leftarrow \emptyset,\ win \leftarrow 0$
$ctr \leftarrow 0$
for $i = 1...d$ **do** :
  $\rho_i \xleftarrow{\mathcal{U}} Func(\{0,1\}^n; \{0,1\}^n)$
$\mathcal{A}^{Tag, Verify}(\ )$
return win

Verify$(seed, \mathsf{S}, N, \mathsf{D}, T)$

if $((seed, \cdot) \in \Pi_1)$ **then**
  $\rho \leftarrow \rho_{\Pi_1(seed)}$
else
  $ctr \leftarrow ctr + 1$
  $\rho \leftarrow \rho_{ctr}$
  $\Pi_1 \leftarrow \Pi_1 \cup \{(seed, ctr)\}$
for $i = 0 \dots w-1$ **do**
  if $((i, S_i, \cdot) \in \Pi_2)$ **then**
    $K_i \leftarrow \Pi_2(i, S_i)$
  else
    $K_i \xleftarrow{\mathcal{U}} \{0,1\}^n,\ \Pi_2 \leftarrow \Pi_2 \cup \{(i, S_i, K_i)\}$
$T' \leftarrow (D_0 \oplus K_0) \cdot K_0 \oplus \dots \oplus$
  $\oplus (D_{w-1} \oplus K_{w-1}) \cdot K_{w-1} \oplus \rho(nonce)$
$\mathsf{st} \leftarrow (seed, \mathsf{S}, N, \mathsf{D}, T)$
if $((T' = T) \wedge (\mathsf{st} \notin STATES))$ **then**
  $win \leftarrow 1$
return $(T' = T)$

Tag$(seed, \mathsf{S}, N, \mathsf{D})$

if $((seed, N) \in GAMMA)$ **then**
  return $\perp$
$GAMMA \leftarrow GAMMA \cup \{(seed, N)\}$
if $((seed, \cdot) \in \Pi_1)$ **then**
  $\rho \leftarrow \rho_{\Pi_1(seed)}$
else
  $ctr \leftarrow ctr + 1$
  $\rho \leftarrow \rho_{ctr}$
  $\Pi_1 \leftarrow \Pi_1 \cup \{(seed, ctr)\}$
for $i = 0 \dots w-1$ **do**
  if $((i, S_i, \cdot) \in \Pi_2)$ **then**
    $K_i \leftarrow \Pi_2(i \| S_i)$
  else
    $K_i \xleftarrow{\mathcal{U}} \{0,1\}^n,\ \Pi_2 \leftarrow \Pi_2 \cup \{(i, S_i, K_i)\}$
$T \leftarrow (D_0 \oplus K_0) \cdot K_0 \oplus \dots \oplus$
  $\oplus (D_{w-1} \oplus K_{w-1}) \cdot K_{w-1} \oplus \rho(nonce)$
$\mathsf{st} \leftarrow (seed, \mathsf{S}, N, \mathsf{D}, T)$
$STATES \leftarrow \{\mathsf{st}\}$
return T

Figure 5: The **Exp**^{2} for the adversary $\mathcal{A}$

$(seed^z, \mathsf{S}^z, N^z, \mathsf{D}^z)$. The *Tag* oracle returns the value $T_z \in \{0,1\}^n$ as an answer to the z-th query.

The adversary $\mathcal{A}$ is deterministic. Consequently, its actions are determined by answers from *Tag* oracle, i.e. by random variables $\widetilde{T_1}, \dots, \widetilde{T_{q_{tag}}}$. If the values of these random variables are fixed by $T_1, \dots, T_{q_{tag}}$, then the all $\mathcal{A}$ queries are fixed including the forgery $(seed^*, N^*, \mathsf{S}^*, \mathsf{D}^*, T^*)$.

Let denote the vector of random variables whose values correspond to $\mathsf{S}^z = (S_0^z, \dots, S_{w-1}^z)$ values (or $\mathsf{S}^* = (S_0^*, \dots, S_{w-1}^*)$) as $\widetilde{\mathsf{K}^z} = (\widetilde{K_0^z}, \dots, \widetilde{K_{w-1}^z})$ $(\widetilde{\mathsf{K}^*} = (\widetilde{K_0^*}, \dots, \widetilde{K_{w-1}^*})$ respectively). Note, that random variables $\widetilde{K_i^j}$ and $\widetilde{K_i^p}$ are independent by [IQRA, KDF] definition for fixed different queries j и p

if $S_i^j \neq S_i^p$. If $S_i^j = S_i^p$, random variables $\widetilde{K_i^j}$ and $\widetilde{K_i^p}$ are dependent and $\Pr\left[\widetilde{K_i^j} = \widetilde{K_i^p}\right] = 1$. Similarly, random variables $\widetilde{K_i^*}$ and $\widetilde{K_i^j}$ are equal if $S_i^j = S_i^*$ and independent if $S_i^j \neq S_i^*$. Thus,

$$\Pr\left[\mathbf{Exp}^2(\mathcal{A}) \to 1\right] = \sum_{T_1,\ldots,T_{q_{tag}}} \Pr\left[\{\mathbf{Exp}^2(\mathcal{A}) \to 1\} \cap \{\widetilde{T}_i = T_i\}_{i=1}^{q_{tag}}\right] =$$

$$= \sum_{T_1,\ldots,T_{q_{tag}}} \sum_{\mathsf{K}^1,\ldots,\mathsf{K}^{q_{tag}},\mathsf{K}^*} \Pr\left[\{\mathbf{Exp}^2(\mathcal{A}) \to 1\} \cap \{\widetilde{T}_i = T_i\}_{i=1}^{q_{tag}} \cap \left\{\begin{matrix}\widetilde{\mathsf{K}^i}=\mathsf{K}^i,\\ \widetilde{\mathsf{K}^*}=\mathsf{K}^*\end{matrix}\right\}_{i=1}^{q_{tag}}\right].$$

The event $\{\mathbf{Exp}^2(\mathcal{A}) \to 1\}$ takes place if

$$\rho^*(N^*) = \underbrace{(D_0^* \oplus \widetilde{K_0^*}) \cdot \widetilde{K_0^*} \oplus \ldots \oplus (D_{w-1}^* \oplus \widetilde{K_{w-1}^*}) \cdot \widetilde{K_{w-1}^*} \oplus T^*}_{=\widetilde{c^*}},$$

where ρ^* — uniform random function corresponding to $seed^*$. If the values $\mathsf{K}^*, \mathsf{D}^*, T^*$ are fixed, then the value of random variable $\widetilde{c^*}$ is also fixed: $\widetilde{c^*} = c^*$.

The event $\{\widetilde{T}_i = T_i\}$ for i-th query, $1 \leqslant i \leqslant q_{tag}$, takes place if

$$\rho^i(N^i) = \underbrace{(D_0^i \oplus \widetilde{K_0^i}) \cdot \widetilde{K_0^i} \oplus \ldots \oplus (D_{w-1}^i \oplus \widetilde{K_{w-1}^i}) \cdot \widetilde{K_{w-1}^i} \oplus T^i}_{=\widetilde{c_i}},$$

where ρ^i — uniform random function corresponding to $seed^i$. If the values $\mathsf{K}^i, \mathsf{D}^i, T^i$ are fixed, then the value of random variable $\widetilde{c_i}$ is also fixed: $\widetilde{c_i} = c_i$. Note, that random variables $\rho^i(N^i)$, $1 \leqslant i \leqslant q_{tag}$, are independent, because the $(seed, N)$ pairs are unique across *Tag* oracle queries. Therefore, the required probability can be represented as:

$$\Pr\left[\mathbf{Exp}^2(\mathcal{A}) \to 1\right] = \sum_{T_1,\ldots,T_{q_{tag}}} \sum_{\mathsf{K}^1,\ldots,\mathsf{K}^{q_{tag}},\mathsf{K}^*} \Pr\left[\{\rho^*(nonce^*) = c^*\} \cap \{\rho^i(nonce^i) = c_i\}_{i=1}^{q_{tag}} \cap \left\{\begin{matrix}\widetilde{\mathsf{K}^i}=\mathsf{K}^i,\\ \widetilde{\mathsf{K}^*}=\mathsf{K}^*\end{matrix}\right\}_{i=1}^{q_{tag}}\right].$$

As the values $\mathsf{K}^i, \mathsf{K}^*$, $1 \leqslant i \leqslant q_{tag}$, and the random functions ρ are selected independently, we have that:

$$\Pr\left[\{\mathbf{Exp}^2(\mathcal{A}) \to 1\}\right] = \sum_{T_1,\ldots,T_{q_{tag}}} \sum_{\mathsf{K}^1,\ldots,\mathsf{K}^{q_{tag}},\mathsf{K}^*} \Pr\left[\left\{\begin{matrix}\widetilde{\mathsf{K}^i}=\mathsf{K}^i,\\ \widetilde{\mathsf{K}^*}=\mathsf{K}^*\end{matrix}\right\}_{i=1}^{q_{tag}}\right] \cdot$$

$$\cdot \Pr\left[\{\rho^*(N^*) = c^*\} \cap \{\rho^i(N^i) = c_i\}_{i=1}^{q_{tag}}\right].$$

Let estimate the probability $p := \Pr\left[\{\rho^*(N^*) = c^*\} \cap \{\rho^i(N^i) = c_i\}_{i=1}^{q_{tag}}\right]$. There are two possible cases:

1. there exists query $z, 1 \leqslant z \leqslant q_{tag}$, such that $seed^* = seed^z, N^* = N^z$. Let denote this case as case;
2. $(seed^*, N^*)$ pair was not used in the queries to Tag oracle. Let denote this case as $\overline{\mathsf{case}}$.

The first case. If the case takes place, the number z is fixed uniquely because of the uniqueness of $(seed, N)$ pair.

Then the value $\rho^*(N^*)$ is equal to some value $\rho^z(N^z)$. Thus,

$$p = \begin{cases} \Pr\left[\{\rho^i(N^i) = c_i\}_{i=1}^{q_{tag}}\right], & \text{if } c^* = c_z; \\ 0, \text{ otherwise.} \end{cases}$$

The probability $\Pr\left[\{\rho^i(N^i) = c_i\}_{i=1}^{q_{tag}}\right]$ is equal to the probability to select set of uniform random functions, such that q_{tag} certain inputs correspond to the certain outputs. This probability is equal to $\dfrac{1}{2^{nq_{tag}}}$.

The second case. If the $\overline{\mathsf{case}}$ takes place, the value of uniform random function $\rho^*(N^*)$ is selected independently, so the probability p is equal to $\Pr[\rho^*(N^*) = c^*] \cdot \Pr\left[\{\rho^i(N^i) = c_i\}_{i=1}^{q_{tag}}\right]$. Similarly to the first case, this is equal to $\dfrac{1}{2^n} \cdot \dfrac{1}{2^{nq_{tag}}} = \dfrac{1}{2^{n(q_{tag}+1)}}$ for arbitrary values c_i, c^*.

Thus, we have that

$$\begin{aligned}
&\Pr\left[\mathbf{Exp}^2(\mathcal{A}) \to 1\right] = \\
&\quad = \sum_{\substack{T_1,\ldots,T_{q_{tag}}: \\ \overline{\mathsf{case}}}} \sum_{\mathsf{K}^1,\ldots,\mathsf{K}^{q_{tag}},\mathsf{K}^*} \Pr\left[\left\{\begin{smallmatrix}\widetilde{\mathsf{K}^i}=\mathsf{K}^i, \\ \widetilde{\mathsf{K}^*}=\mathsf{K}^*\end{smallmatrix}\right\}_{i=1}^{q_{tag}}\right] \cdot \frac{1}{2^{n(q_{tag}+1)}} + \\
&\quad + \sum_{\substack{T_1,\ldots,T_{q_{tag}}: \\ \mathsf{case}}} \sum_{\substack{\mathsf{K}^1,\ldots,\mathsf{K}^{q_{tag}},\mathsf{K}^*: \\ c^*=c_z}} \Pr\left[\left\{\begin{smallmatrix}\widetilde{\mathsf{K}^i}=\mathsf{K}^i, \\ \widetilde{\mathsf{K}^*}=\mathsf{K}^*\end{smallmatrix}\right\}_{i=1}^{q_{tag}}\right] \cdot \frac{1}{2^{nq_{tag}}} = \\
&\quad = \sum_{\substack{T_1,\ldots,T_{q_{tag}}: \\ \overline{\mathsf{case}}}} \frac{1}{2^{n(q_{tag}+1)}} \underbrace{\sum_{\mathsf{K}^1,\ldots,\mathsf{K}^{q_{tag}},\mathsf{K}^*} \Pr\left[\left\{\begin{smallmatrix}\widetilde{\mathsf{K}^i}=\mathsf{K}^i, \\ \widetilde{\mathsf{K}^*}=\mathsf{K}^*\end{smallmatrix}\right\}_{i=1}^{q_{tag}}\right]}_{=1} + \\
&\quad + \sum_{\substack{T_1,\ldots,T_{q_{tag}}: \\ \mathsf{case}}} \frac{1}{2^{nq_{tag}}} \sum_{\substack{\mathsf{K}^1,\ldots,\mathsf{K}^{q_{tag}},\mathsf{K}^*: \\ c^*=c_z}} \Pr\left[\left\{\begin{smallmatrix}\widetilde{\mathsf{K}^i}=\mathsf{K}^i, \\ \widetilde{\mathsf{K}^*}=\mathsf{K}^*\end{smallmatrix}\right\}_{i=1}^{q_{tag}}\right] = \\
&\quad = \sum_{\substack{T_1,\ldots,T_{q_{tag}}: \\ \overline{\mathsf{case}}}} \frac{1}{2^{n(q_{tag}+1)}} + \sum_{\substack{T_1,\ldots,T_{q_{tag}}: \\ \mathsf{case}}} \frac{1}{2^{nq_{tag}}} \Pr\left[\widetilde{c^*} = \widetilde{c_z}\right].
\end{aligned}$$

Let estimate the probability $\Pr\left[\widetilde{c^*} = \widetilde{c_z}\right]$. Event $\{\widetilde{c^*} = \widetilde{c_z}\}$ takes place if

$$(D_0^* \oplus \widetilde{K_0^*}) \cdot \widetilde{K_0^*} \oplus \ldots \oplus (D_{w-1}^* \oplus \widetilde{K_{w-1}^*}) \cdot \widetilde{K_{w-1}^*} \oplus$$
$$\oplus (D_0^z \oplus \widetilde{K_0^z}) \cdot \widetilde{K_0^z} \oplus \ldots \oplus (D_{w-1}^z \oplus \widetilde{K_{w-1}^z}) \cdot \widetilde{K_{w-1}^z} = T_z \oplus T^*.$$

The probability $\Pr\left[\widetilde{c^*} = \widetilde{c_z}\right]$ is defined by random variables $\widetilde{K_0^z}, \ldots, \widetilde{K_{w-1}^z}$ and $\widetilde{K_0^*}, \ldots, \widetilde{K_{w-1}^*}$. If the values $T_1, \ldots, T_{q_{tag}}$ are fixed, then the set of blocks with $S_i^z = S_i^*$ is fixed, let denote their number as m, $0 \leqslant m \leqslant w$. Let consider two possible cases.

At first, let $m = w$. Then $\widetilde{K_i^*} = \widetilde{K_i^z}, i \in \{0, \ldots, w-1\}$. Thus, the event $\{\widetilde{c^*} = \widetilde{c_z}\}$ takes place if the following equation becomes true:

$$(D_0^* \oplus D_0^z) \cdot \widetilde{K_0^*} \oplus \ldots \oplus (D_{w-1}^* \oplus D_{w-1}^z) \cdot \widetilde{K_{w-1}^*} = T_z \oplus T^*.$$

Let us show that there exists at least one i, such that $D_i^* \oplus D_i^z \neq 0$. Condition $m = w$ implies $\mathsf{S}^z = \mathsf{S}^*$. The values $seed^*, N^*$ are also equal to $seed^z, N^z$ by definition of case. If $D_i^* = D_i^z$ for all i, $0 \leqslant i \leqslant w-1$, then $\mathsf{D}^* = \mathsf{D}^z$. Then the equation above becomes true only if $T_z = T^*$. But this contradicts with the condition that forgery must be not trivial. Thus, there exists at least one i, such that $D_i^* \neq D_i^z$.

Then number of solutions of the equation above can be calculated in the following way. Let choose i, such that $D_i^* \oplus D_i^z \neq 0$. Let fix all K_j^*, $j \neq i$, by arbitrary values and recover K_i^* explicitly turning the equation above into true equality. Thus, the number of solutions is equal to the number of all possible sets $K_0^*, \ldots, K_{i-1}^*, K_{i+1}^*, \ldots, K_{w-1}^*$, that is $2^{n(w-1)}$. Therefore,

$$\Pr\left[\widetilde{c^*} = \widetilde{c_z}\right] =$$
$$= \frac{\#\{K_0^*, \ldots, K_{w-1}^* : (D_0^* \oplus D_0^z) \cdot K_0^* \oplus \ldots \oplus (D_{w-1}^* \oplus D_{w-1}^z) \cdot K_{w-1}^* = T_z \oplus T^*\}}{2^{nw}}$$
$$= \frac{2^{n(w-1)}}{2^{nw}} = \frac{1}{2^n}.$$

At second, let $m < w$. Without loss of generality let assume that for the first m blocks $S_i^z = S_i^*$. Then the corresponding random variables are dependent, namely $K_0^z = K_0^*, \ldots, K_{m-1}^z = K_{m-1}^*$. Thus, the event $\{\widetilde{c^*} = \widetilde{c_z}\}$ takes place if the following equation becomes true:

$$\alpha_0 \widetilde{K'_0} \oplus \ldots \oplus \alpha_{m-1} \widetilde{K'_{m-1}} \oplus \widetilde{K'_m} \cdot \widetilde{K'_m} \oplus \alpha_m \widetilde{K'_m} \oplus \ldots \oplus$$
$$\oplus \widetilde{K'_{2w-m-1}} \cdot \widetilde{K'_{2w-m-1}} \oplus \alpha_{2w-m-1} \widetilde{K'_{2w-m-1}} = T_z \oplus T^*$$

where

– $\alpha_i = D_i^z \oplus D_i^*, \widetilde{K'_i} = \widetilde{K_i^z} = \widetilde{K_i^*}$ для $0 \leqslant i \leqslant m-1$,

– $\alpha_i = D_i^z, \widetilde{K'_i} = \widetilde{K_i^z}$ для $m \leqslant i \leqslant w-1$,

– $\alpha_i = D_{i-w+m}^*, \widetilde{K'_i} = \widetilde{K_{i-w+m}^*}$ для $w \leqslant i \leqslant 2w-m-1$.

This is a quadratic equation. The number of solutions of this equation can be calculated in the following way. Let choose some i, $m \leqslant i \leqslant 2w-m-1$, and fix all K'_j, $j \neq i$, by arbitrary values. Then the equation turns into quadratic equation for $\widetilde{K'_i}$ random variable that has at most two roots. Thus, the number of solutions is obtained from 2 possible variants for K'_i and number of all possible sets $K'_0, \ldots, K'_{i-1}, K'_{i+1}, \ldots, K'_{w-1}$, that is $2^{n(w-1)+1}$. Therefore,

$$\Pr\left[\widetilde{c^*} = \widetilde{c_z}\right] \leqslant \frac{2 \cdot 2^{n(w-1)}}{2^{nw}} = \frac{2}{2^n}.$$

Summing up the estimates in two cases, we have that for all possible m $\Pr\left[\widetilde{c^*} = \widetilde{c_z}\right] \leqslant \dfrac{2}{2^n}$. Finally, we have

$$\Pr\left[\mathbf{Exp}^2(\mathcal{A}) \to 1\right] \leqslant \sum_{\substack{T_1,\ldots,T_{q_{tag}}: \\ \overline{\mathsf{case}}}} \frac{1}{2^{n(q_{tag}+1)}} + \sum_{\substack{T_1,\ldots,T_{q_{tag}}: \\ \mathsf{case}}} \frac{1}{2^{nq_{tag}}} \cdot \frac{2}{2^n} \leqslant \sum_{T_1,\ldots,T_{q_{tag}}} \frac{2}{2^{n(q_{tag}+1)}}.$$

The number of possible sets $T_1, \ldots, T_{q_{tag}}$ is $2^{nq_{tag}}$. Thus,

$$\Pr\left[\mathbf{Exp}^2(\mathcal{A}) \to 1\right] \leqslant \frac{2 \cdot 2^{nq_{tag}}}{2^{n(q_{tag}+1)}} = \frac{1}{2^{n-1}}.$$

Let consider $q_{ver} > 1$ case. Applying Theorem 5.1 [18] and assuming $n = (seed \parallel \mathsf{S} \parallel N)$, we get

$$\Pr\left[\mathbf{Exp}^2(\mathcal{A}) \to 1\right] \leqslant \frac{q_{ver}}{2^{n-1}}.$$

Final result. Summarizing all the obtained bounds, we get a theorem statement:

$$\begin{aligned}
\mathsf{Adv}^{\text{SUF-CMA}}_{[\text{IQRA}_\text{F},\text{KDF}]}(\mathcal{A}) = \Pr\left[\mathbf{Exp}^0(\mathcal{A}) \to 1\right] = \\
\left(\Pr\left[\mathbf{Exp}^0(\mathcal{A}) \to 1\right] - \Pr\left[\mathbf{Exp}^1(\mathcal{A}) \to 1\right]\right) + \\
+ \left(\Pr\left[\mathbf{Exp}^1(\mathcal{A}) \to 1\right] - \Pr\left[\mathbf{Exp}^{1'}(\mathcal{A}) \to 1\right]\right) + \\
+ \left(\Pr\left[\mathbf{Exp}^{1'}(\mathcal{A}) \to 1\right] - \Pr\left[\mathbf{Exp}^2(\mathcal{A}) \to 1\right]\right) + \Pr\left[\mathbf{Exp}^2(\mathcal{A}) \to 1\right] \leqslant \\
\leqslant \mathsf{Adv}^{\text{PRF}^*}_{\text{KDF}}(\mathcal{B}) + d \cdot \mathsf{Adv}^{\text{PRF}}_{\text{F}}(\mathcal{C}) + \frac{q_{ver}}{2^{n-1}}.
\end{aligned}$$

□

B.3 Corollary 1 proof

Before proving the corollary let us prove the extension of the Bernshtein bound (Theorem 2.1 [18]) for the case of multiple oracle access.

Lemma 3. *Let $\pi_1, \ldots, \pi_d$ be uniform random permutations over the set $\{0,1\}^n$. Let $\rho_1, \ldots, \rho_d$ be uniform random functions from $\{0,1\}^n$ to $\{0,1\}^n$. Let $\mathcal{M}$ be an algorithm that performs exactly $q_1, \ldots, q_d$ distinct queries to the corresponding oracles, $q_i \leqslant 2^n$. Then*

$$\Pr[\mathcal{M}^{\pi_1,\ldots,\pi_d} \to 1] \leqslant \delta_n(q_1) \cdot \ldots \cdot \delta_n(q_d) \Pr[\mathcal{M}^{\rho_1,\ldots,\rho_d} \to 1],$$

where $\delta_n(q) = \left(1 - \frac{q-1}{2^n}\right)^{-q/2}$.

Lemma proof. According to Theorem 4.2 [19], the uniform random permutation over the set $\{0,1\}^n$ has maximum q-interpolation probability at most $\delta_n(q)$ for each $0 \leqslant q \leqslant 2^n$. This means that for the uniform random permutation π:

$$\Pr[(\pi(s_1), \ldots, \pi(s_q)) = (t_1, \ldots, t_q)] \leqslant \frac{\delta_n(q)}{2^{nq}}$$

for all $(t_1, \ldots, t_q) \in \{0,1\}^{nq}$ and all $(s_1, \ldots, s_q) \in \{0,1\}^{nq}$ with $s_1, \ldots, s_q$ distinct.

Let denote as $\widetilde{t^i}$, $1 \leqslant i \leqslant d$, the vector of the random variables whose values correspond to the responses obtained by $\mathcal{M}$ from i-th oracle: $t^i = (t^i_1, \ldots, t^i_{q_i}), t_i \in \{0,1\}^n$. Everything that $\mathcal{M}$ does is determined by its random tape independent of $\mathcal{M}$'s input and by responses $t^1, \ldots, t^d$ to $\mathcal{M}$ distinct queries to each oracle.

Let denote as $\alpha(t^1, \ldots, t^d)$ the conditional probability that $\mathcal{M}$ returns 1 given that the responses to $\mathcal{M}$ distinct queries to each oracle are $t^1, \ldots, t^d$:

$$\alpha(t^1, \ldots, t^d) = \Pr\left[\mathcal{M} \to 1 \;\middle|\; \{\widetilde{t^i} = t^i\}_{i=1}^d\right]$$

Therefore,

$$\Pr[\mathcal{M}^{\pi_1,\ldots,\pi_d} \to 1] = \sum_{\substack{t^1=(t^1_1,\ldots,t^1_{q_1}),\\ \vdots \\ t^d=(t^d_1,\ldots,t^d_{q_d})}} \alpha(t^1, \ldots, t^d) \Pr\left[\{\widetilde{t^i} = t^i\}_{i=1}^d\right] \leqslant$$

$$\leqslant \sum_{\substack{t^1=(t^1_1,\ldots,t^1_{q_1}),\\ \vdots \\ t^d=(t^d_1,\ldots,t^d_{q_d})}} \alpha(t^1, \ldots, t^d) \cdot \frac{\delta_n(q_1) \cdot \ldots \cdot \delta_n(q_d)}{2^{n(q_1+\ldots+q_d)}} =$$

$$= \delta_n(q_1) \cdot \ldots \cdot \delta_n(q_d) \Pr[\mathcal{M}^{\rho_1,\ldots,\rho_d} \to 1].$$

□

Now we are ready to prove Corollary 1.

Corollary proof. We rely on the proof of the Theorem 1. The first steps of the proof are exactly the same: we define $\mathbf{Exp}^0$, $\mathbf{Exp}^1$ and $\mathbf{Exp}^{1'}$ similarly to the one defined in Theorem 1, assuming block cipher E is used as F. Further we are moving from block cipher to the set of uniform random functions not directly as in Theorem 1 proof, but through the set of uniform random permutations. We construct the auxiliary experiment to obtain the better security bound.

Let $\mathbf{Exp}^2$ be the same experiment as $\mathbf{Exp}^2$ in Theorem 1 proof, except that uniform random permutations $\pi_1, \ldots, \pi_d$ are selected instead of uniform random functions $\rho_1, \ldots, \rho_d$. Similarly to the Theorem 1 proof, we construct adversaries $\mathcal{D}$ and $\mathcal{C}$ for block cipher E in the mu-PRP and PRP models respectively to estimate the difference between $\mathbf{Exp}^{1'}$ and $\mathbf{Exp}^2$:

$$\begin{aligned}\Pr\left[\mathbf{Exp}^{1'}(\mathcal{A}) \to 1\right] - \Pr\left[\mathbf{Exp}^{2}(\mathcal{A}) \to 1\right] = \\ = \Pr\left[\mathbf{Exp}_{\mathrm{E}}^{\text{mu-PRP-1}}(\mathcal{D}) \to 1\right] - \Pr\left[\mathbf{Exp}_{\mathrm{E}}^{\text{mu-PRP-0}}(\mathcal{D}) \to 1\right] = \\ = \mathsf{Adv}_{\mathrm{E}}^{\text{mu-PRP}}(\mathcal{D}) \leq d \cdot \mathsf{Adv}_{\mathrm{E}}^{\text{PRP}}(\mathcal{C}).\end{aligned}$$

Let $\mathbf{Exp}^3$ be the modification of the $\mathbf{Exp}^2$ obtained by replacing the uniform random permutations $\pi_1 \ldots \pi_d$ by uniform random functions $\rho_1, \ldots, \rho_d$. Note, that such $\mathbf{Exp}^3$ is exactly the same as $\mathbf{Exp}^2$ in Theorem 1 proof. Thus, $\Pr\left[\mathbf{Exp}^3(\mathcal{A}) \to 1\right] \leqslant \dfrac{q_{ver}}{2^{n-1}}$.

Let construct an adversary $\mathcal{M}$ to estimate the difference between $\mathbf{Exp}^2$ and $\mathbf{Exp}^3$. Let $\mathcal{M}$ invokes $\mathcal{A}$ as a subroutine and implements $\mathbf{Exp}^3$ for $\mathcal{A}$ replacing uniform random functions calls by queries to its own oracles. Let assume that $\mathcal{M}$ has access to d oracles. These oracles can either implement $\pi_1, \ldots, \pi_d$, or $\rho_1, \ldots, \rho_d$. The number of $\mathcal{M}$ queries to each oracle is exactly the same as the number of queries made by $\mathcal{A}$ with the corresponding *seed* value and distinct nonce values. The adversary $\mathcal{M}$ returns 1, if adversary $\mathcal{A}$ wins. Thus,

$$\begin{aligned}\Pr\left[\mathbf{Exp}^{2}(\mathcal{A}) \to 1\right] = \Pr\left[\mathcal{M}^{\pi_1,\ldots,\pi_d} \to 1\right], \\ \Pr\left[\mathbf{Exp}^{3}(\mathcal{A}) \to 1\right] = \Pr\left[\mathcal{M}^{\rho_1,\ldots,\rho_d} \to 1\right].\end{aligned}$$

According to Lemma 3,

$$\Pr\left[\mathcal{M}^{\pi_1,\ldots,\pi_d} \to 1\right] \leqslant \delta_n(r_1) \cdot \ldots \cdot \delta_n(r_d) \Pr\left[\mathcal{M}^{\rho_1,\ldots,\rho_d} \to 1\right],$$

where $\delta_n(q) = \left(1 - \frac{q-1}{2^n}\right)^{-q/2}$ and $r_1, \ldots, r_d$ are the number of queries made by $\mathcal{A}$ with each *seed* value and distinct nonce values. Therefore,

$$\Pr\left[\mathbf{Exp}^2(\mathcal{A}) \to 1\right] \leqslant \delta_n(r_1) \cdot \ldots \cdot \delta_n(r_d) \Pr\left[\mathbf{Exp}^3(\mathcal{A}) \to 1\right] \leqslant \\ \leqslant \delta_n(r_1) \cdot \ldots \cdot \delta_n(r_d) \cdot \frac{q_{ver}}{2^{n-1}}.$$

Summing up, we obtain the following bound:

$$\mathsf{Adv}^{\text{SUF-CMA}}_{[\text{IQRA}_\text{E},\text{KDF}]}(\mathcal{A}) = \Pr\left[\mathbf{Exp}^0(\mathcal{A}) \to 1\right] = \\ \left(\Pr\left[\mathbf{Exp}^0(\mathcal{A}) \to 1\right] - \Pr\left[\mathbf{Exp}^1(\mathcal{A}) \to 1\right]\right) + \\ + \left(\Pr\left[\mathbf{Exp}^1(\mathcal{A}) \to 1\right] - \Pr\left[\mathbf{Exp}^{1'}(\mathcal{A}) \to 1\right]\right) + \\ + \left(\Pr\left[\mathbf{Exp}^{1'}(\mathcal{A}) \to 1\right] - \Pr\left[\mathbf{Exp}^2(\mathcal{A}) \to 1\right]\right) + \Pr\left[\mathbf{Exp}^2(\mathcal{A}) \to 1\right] \leqslant \\ \leqslant \mathsf{Adv}^{\text{PRF}^*}_{\text{KDF}}(\mathcal{B}) + d \cdot \mathsf{Adv}^{\text{PRP}}_{\text{E}}(\mathcal{C}) + \delta_n(r_1) \cdot \ldots \cdot \delta_n(r_d) \cdot \frac{q_{ver}}{2^{n-1}}.$$

□

The re-keying mechanism COM-CTR+D

Daymé Almeida, Alejandro Freyre, and Adrián Alfonso

Institute of Cryptography, University of Havana, Cuba.
yamilkate@infomed.sld.cu, alefreyre.43@gmail.com

Abstract

Most of the attacks on block ciphers depend on the analysis of large amounts of data encrypted under the same key. In this paper is presented COM-CTR+D, a new mode of operation for block ciphers to extend the lifetime of keys, taking advantage of the properties of external and internal re-keying mechanisms. In addition, we carry out the security analysis of the mode of operation COM-CTR+D and we discuss aspects related to the implementation details of the ECB-CTR+D mode of operation for the Advanced Encryption Standard (AES).

Keywords: encryption mode, re-keying mechanism, key lifetime cycle, advanced encryption standard.

1 Introduction

Cryptographic protocols rarely use the key shared between parties in the encryption process. Key derivation is a common procedure used to avoid cryptanalysis, where the keys derived from the master key are used in the different processes or components of the system. It has been shown that this way has proven valid to increase the security in the encryption process [10], although, in the last years, other techniques have emerged as alternatives that differ depending on the environment in which they are used, but effectively extend the lifetime of keys. Such methods are known as re-keying mechanisms and they are divided into three groups: external re-keying [10, 12], internal re-keying [5, 6] and fresh re-keying [19].

1.1 Our contribution

In this paper we introduce a new mode of operation for block ciphers called COM-CTR+D to extend the lifetime of keys. The COM function of the new mode refers to one of the following confidentiality encryption modes of block ciphers: ECB, CBC, OFB, CFB [11] or CTR [24], while the pseudo-random transformation used to update the section keys is determined by the CTR encryption mode [24]. The difference of this method over similar

mechanisms provided in literature survey is that the lifetime of a section key relies in the security of the underlying COM encryption mode. We show that the use of the CTR mode as pseudo-random function to update the section keys guarantees its non-repetition and we also provide the probabilistic security bound for COM-CTR+D encryption mode. Finally, we analyze the implementation of ECB-CTR+D with the standard AES [14].

2 Internal and external re-keying approaches

The authors in [6] introduced the formal concept of Internal and External re-keying approaches and discuss their features, advantages and disadvantages. The first one was presented like a generalization of a key diversification scheme [10] and the second one like advanced technique of the mechanism to increase the key lifetime called «CryptoPro Key Meshing» (CPKM) [22].

2.1 Internal re-keying

The internal re-keying approach modifies a base mode of operation in such a way that each message is processed starting from the same key, which is changed using certain key update technique during the processing of the current message, so that it is integrated into the base mode of operation and changes its internal structure. One of the main concepts of internal re-keying is a «section», defined like a string consisting in all message blocks processed with the same key, also called a «section key». The parameter of this modes is the section size, chosen optionally since it affects the operating properties and limits the amount of messages. Examples of re-keying internal mechanism can be seen in [1, 5, 6, 16, 21, 22].

These methods are recommended to be used in protocols that process large single messages since the maximum gain in increasing the key lifetime is achieved by increasing the length of a message, while it provides almost no increase in the number of messages that can be processed with one key [6]. The general procedure of the internal re-keying mechanism is show below.

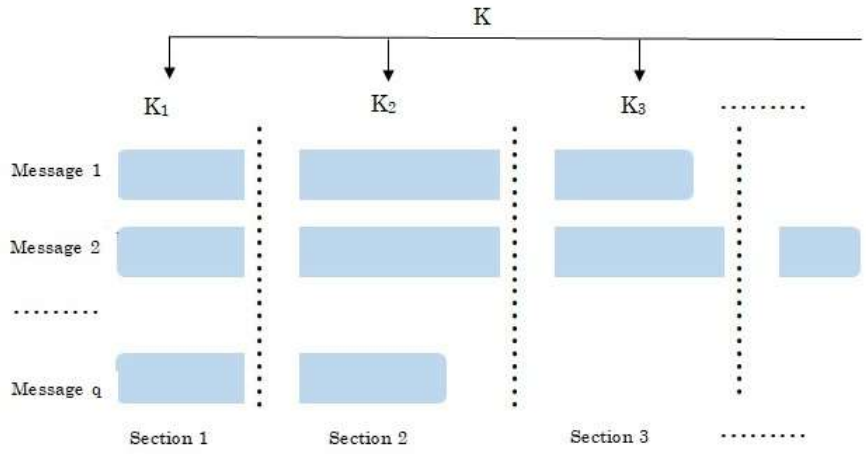

Figure 1: Internal re-keying. The value q indicate the number of processed messages and each message is processed starting from the first derived key K_1. This key is changed each time a data section of fixed length l has been processed. The lifetime L of the key K defines the total length of data processed whit this key.

Now we present some background about the internal re-keying mechanism CTR-ACPKM which helps to understand the later definition of our mode.

2.1.1 Internal re-keying CTR-ACPKM mode

This mode of operation is defined in [1] like the advanced CPKM mode proposed in [22] and works as can be seen in figure 2 for a given plaintext P of m blocks of size n.

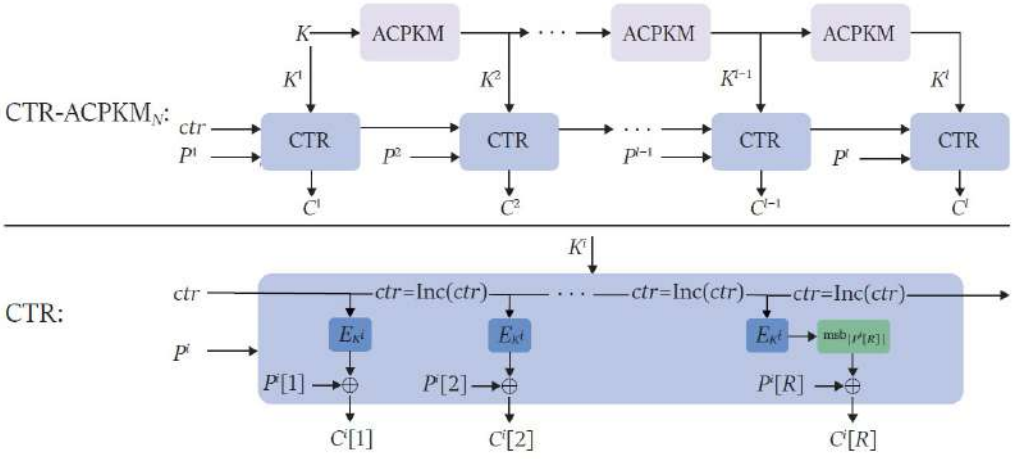

Figure 2: Internal re-keying CTR-ACPKM mode

Here the first section of each message is processed starting with the same key (the first section key) and each section is continued to be processed using the subroutine CTR (the base mode) under the respective section key,

where the key update technique after processing all the N blocks of a section consists in the transformation ACPKM defined as follows:

$$K^{i+1} = \text{ACPKM}(K^i) = msb_k(E_{k^i}(D_1))\| \ldots \|E_{k^i}(D_s)) \tag{1}$$

where $s = \lceil k/n \rceil$ and $D_1, \cdots, D_s \in \{0,1\}^n$ are arbitrary pairwise different constants, so that the $(n/2)$-th bit (counting from the right) of each of these constants is equal to 1. Note that the internal state of the CTR-ACPKM$_N$ mode, the counter, is not reset for each new section and the condition on the $D_1, D_2, \ldots . D_s$ constants allows to prevent collisions of block cipher permutation inputs in cases of key transformation and message processing.

In [1] it is demonstrated the following theorem [5, Theorem 3.1].

Theorem 1. *Let N be the parameter of CTR-ACPKM$_N$ mode. Then for any adversary A with time complexity at most t that makes queries, where the maximum message length is at most $m (m \leq 2^{n/2-1})$ blocks and the total message length is at most σ blocks, there exists an adversary B such that*

$$\boldsymbol{Adv}^{\mathsf{ind-cpna}}_{CTR-ACPKM_N}(A) \leq l \cdot \boldsymbol{Adv}^{\mathsf{prp-cpa}}_{\mathcal{E}}(B) + \frac{(\sigma_1 + s)^2 + .. + (\sigma_{l-1} + s)^2 + (\sigma_l)^2}{2^{n+1}}$$

where $s = \lceil k/n \rceil$, $l = \lceil m/N \rceil$, σ_j is the total data block length processed under the section key K^j and $\sigma_j \leq 2^{n-1}, \sigma_1 + \ldots + \sigma_l = \sigma$. The adversary B makes at most $\sigma_1 + s$ queries. Furthermore, the time complexity of B is at most $t + cn(\sigma + ls)$, where c is a constant that depends only on the model of computation and the method of encoding.

2.2 External re-keying

In this approach a key, derived according to certain key update technique, is intended to process the fixed amount of separate messages after which the key should be updated and is proposed to be performed each time a given amount of messages is processed. However, the key lifetime is defined by the total length of the processed messages and not by their amount.

External re-keying is recommended for usage in protocols that process quite small messages since the maximum gain in increasing the key lifetime is achieved by increasing the number of messages.

Doubtless advantage of external re-keying is the possibility to explicitly use the obtained security bounds for the base mode to quantify security of the corresponding externally re-keyed mode [6]. In [10] are given the bound of this mechanism accord to key derivation generator used and quantify the security as a function of the security of the primitives used. The general procedure of the internal re-keying mechanism [6] is show below.

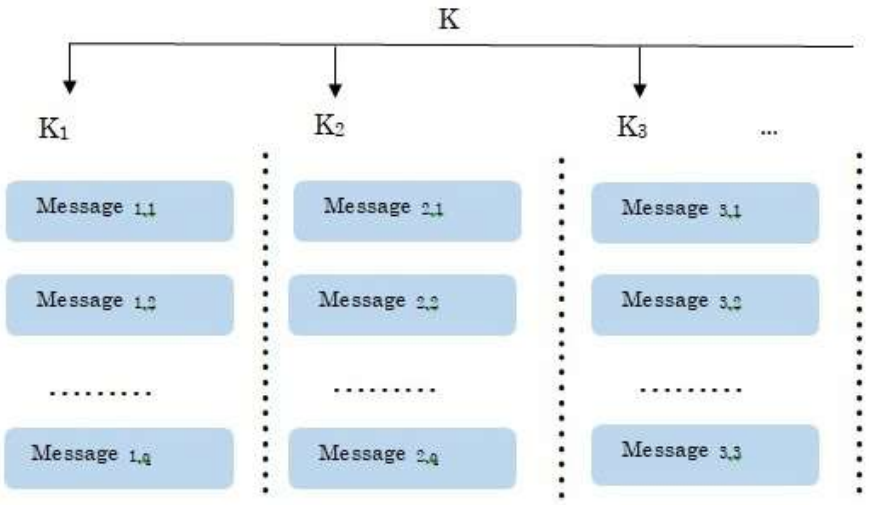

Figure 3: External re-keying. The notation $M_{i,j}$ denote that have been encrypted the j-th message whit the key K_i.

3 The mode of operation COM-CTR+D

In our proposal the message is processed starting with a first section key depending on the initialization vector IV through the CTR mode, which is updated once processed a certain number of blocks N (one section). Given the value of the parameter N, COM-CTR+D$_N$ process the input plaintext as follows.

The plaintext X, of m blocks of size n, splits into $L = \lceil m/N \rceil$ sections (denoted as $X = X^1||X^2||...||X^L$, where $X^i \in \{0,1\}^{nN}$ for $1 \leq i \leq L-1$ and $X^L \in \{0,1\}^r, r \leqslant nN$) which will be processed under the initial key. The section X^1 is encrypted using the confidentiality mode COM and the section key K^1. Then, the i-th section of message X is processed using the confidentiality mode with section key K^i, which is calculated for all $1 \leq i \leq L$ by the pseudo-random transformation CTR_i as follows:

$$\begin{aligned} K^i &= CTR_i(ctr_i) \\ &= E_K(ctr_i) \end{aligned} \tag{2}$$

where $ctr_i = IV + i$ is no other than addition of the initialization vector and the n-bit integer i. Finally, the ciphertext $Y = Y^1||Y^2||...||Y^L$ generated by COM-CTR+D$_N$ mode of operation is computed in the following way:

$$Y^i = \mathrm{COM}_{K^i}(X^i, A_i) \tag{3}$$

Here, Y^i denotes the *i-th* section of ciphertext, A_i are the initial values of section i for the COM mode that may be dependent on the values of the previous section, and K^i is obtained through equation 2. For some confidentiality modes COM, like ECB, the values of A_i are not necessary. In appendix 1, we present the encryption schemes of COM-CTR+D for mentioned confidentiality modes. Note that the internal state of COM-CTR+D$_N$, i.e. the value of ctr_i, is updated for each new section to prevent collisions. Figure 4 shows the design of COM-CTR+D$_N$ re-keying mechanism.

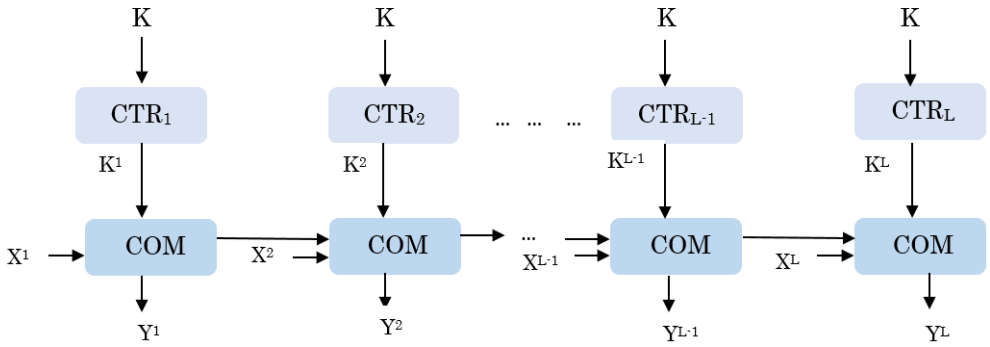

Figure 4: General scheme of COM-CTR+D$_N$ mode of operation.

Here we left some important considerations with respect to the COM-CTR+D re-keyed mechanism. Firstly, the re-keyed mechanism should reinforce the security of the base mode COM and its use must increase the lifetime of the master key. In addition, the section size is not greater than the lifetime of the section key which is bounded by the security of the base mode of operation. Furthermore, the behavior of any section key does not compromise the remaining section keys nor the sections of plaintext of the message. Finally, as long as the master key is not changed, the initial counter of the CTR mode for a new message start from the next value of the final ctr_i of the last encrypted message, as stated in [24].

It is important to notice that several cryptographic algorithms can be used as underlaying block cipher of the proposed mode COM-CTR+D taking into account that the sizes of the initialization vector, initial key, section key and plaintext blocks are not restricted. For example, we can use Rijndael algorithm [13] in the transformation CTR_i with 256 bits of key and plaintext sizes, which results in 256 bits of section key and the Magma algorithm [15] with the COM mode.

Next, we introduce the security bounds for the COM-CTR+D re-keying scheme.

3.1 Probabilistic security of COM-CTR+D

The probabilistic security analysis of our mode of operation is conducted according [23], where the following conditions are assumed:

C1 Given $N = u + v$ plaintext blocks processed under the same key such that $x_1, x_2, ..., x_u$ and $x'_1, x'_2, ..., x'_v$ are respectively known and unknown by an adversary.

C2 For a known block x_i, $1 \leqslant i \leqslant u$, we have that $y'_j = y_i$ for an unknown block x'_j, $1 \leqslant j \leqslant v$.

In this case, we find the amount of messages that can be encrypted with the same key through the simulation of a phenomenon A for the previous conditions, and we estimate the probability to obtain additional information about the unknown parts of the plaintext from a known ciphertext and some known parts of the plaintext. With conditions **C1** and **C2**, the authors of [23] present the probabilistic security bounds for the five confidentiality modes ECB, CBC, OFB, CFB and CTR, which can be used in place of the COM function in our proposal. The next theorem is the result of continuing this line of work.

Theorem 2. *Given the probabilistic security bounds $N_{max_{COM}}$ and $N_{max_{CTR}}$ for the confidentiality mode COM and the CTR mode respectively, under conditions **C1** and **C2**, the maximum amount of data that can be safely processed by COM-CTR+D_N satisfies the following inequality*

$$N_{max_{COM-CTR+D_N}} \leqslant N_{max_{CTR}} \cdot N_{max_{COM}} = 2^{\frac{n}{2}+1}\sqrt{\ln(\pi 2^n)} \cdot N_{max_{COM}} \qquad (4)$$

where π is the adversary's success probability.

The proof of this theorem is straightforward, since for each section key generated through the CTR mode, the maximum amount of data that can be safety processed by the confidentiality mode COM is upper bounded by the probabilistic security bound $N_{max_{COM}}$. For the confidentiality modes ECB, CBC, OFB, CFB and CTR the respective values of $N_{max_{COM}}$ can be seen in [23].

Remark 1. *If the section size N equals the maximum number of blocks N_{max} that can be processed by the confidentiality mode COM, then COM-CTR+D_N satisfies the equality in theorem 2.*

Remark 2. *If the section size is N=1 the maximum number of blocks that can be processed by COM-CTR+D_1 with the same key equals the probabilistic bound for the CTR mode.*

3.2 Probable security of COM-CTR+D mode

The foundations of the encryption mode COM-CTR+D rest in the choice of the confidentiality mode COM, therefore the properties of our proposal are highly dependent on the base mode, extending the lifetime of the cipher key and increasing the combinatorial properties of the block cipher. In our particular case, such properties are increased in the order of the probabilistic security bound for the CTR mode.

However, the scheme of the COM-CTR+D mechanism satisfies, but it is not restricted to the conditions presented in [10] related to external methods for key lifetime extension. For the cases where the conditions presented in [10] are satisfied, we built a re-keying method on the basis of a confidentiality mode, whose process works as an external mechanism. Such mechanism is the result of relate a generator, a base scheme and the key lifetime [10]. Hence, for any symmetric encryption scheme with well defined key generation, encryption and decryption algorithms $\mathcal{SE} = (\mathsf{K_e}, \mathcal{E}, \mathcal{D})$ and $\mathcal{G} = (\mathsf{K_g}, \mathcal{N})$, a stateful generator with block size k, being k the size of the key associated to the base mode and $l > 0$ the sub-key lifetime, we can associate an extended encryption scheme $\overline{\mathcal{SE}}[\mathcal{SE}, \mathcal{G}, l] = (\overline{\mathsf{K}}, \overline{\mathcal{E}}, \overline{\mathcal{D}})$. Furthermore, from the definitions of parallel generator and external mechanism and Corollary 1. proposed in [10], we can set the security bounds of mode COM-CTR+D for these cases trough theorem 3.

Theorem 3. *Given COM and CTR, the base encryption mode and parallel generator respectively, and $N_{COM} > 0$ the section key lifetime, if $COM - CTR + D = [COM, CTR, N_{COM}]$ is the associated re-keying scheme then*

$$\boldsymbol{Adv}^{\mathsf{ind-cpa}}_{COM-CTR+D}(t, n \cdot N_{COM}, m) \leq \boldsymbol{Adv}^{\mathsf{prf}}_{CTR,n}(t) + n \cdot \boldsymbol{Adv}^{\mathsf{ind-cpa}}_{COM}(t, N_{COM}, m)$$

where t denotes the execution time, N_{COM} is the number of allowed questions to the oracle in the form of m-bit messages pairs, and $n < N_{max_{CTR}}$ is the number of output blocks of the generator.

3.3 Security analysis for the confidentiality mode ECB

Within our proposal, one can link the CTR mode to any confidentiality mode as discussed earlier. Therefore, we present the analysis of the ECB-CTR+D mode of operation which is the combination of the confidentiality modes ECB and CTR, both selected due their simplicity w.r.t implementation and the proven security of CTR mode. In addition, this chaining of confidentiality modes guaranties the non-propagation of errors in the decryption process, parallel programming capabilities and simple design scheme,

guaranteeing the extension of the lifetime of the key employed in the encryption/decryption process. Figure 5 shows the design of ECB-CTR+D mode of operation.

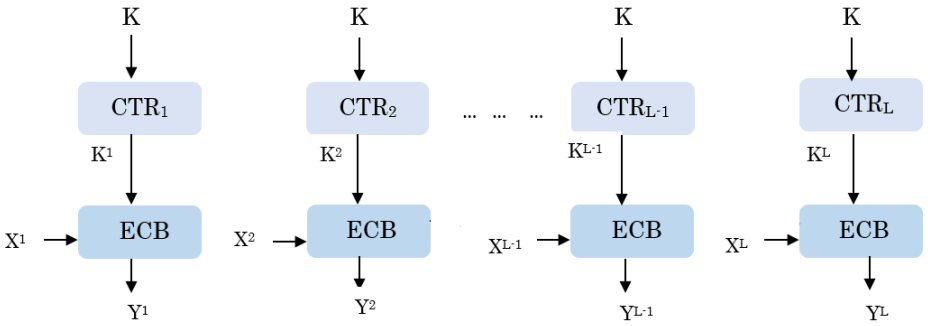

Figure 5: ECB-CTR+D general scheme.

In the security analysis conducted later, we assume that the section size N is equal to 1, i.e, each section contains a single block of data, which is consistent with the re-keying mechanism ECB-CTR+D_1 of figure 6.

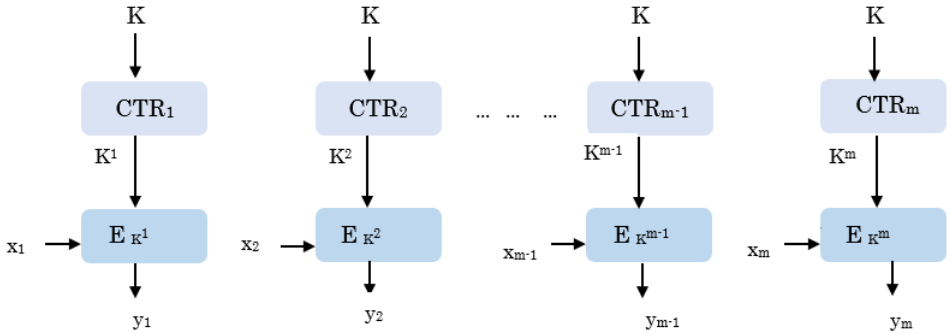

Figure 6: ECB-CTR+D_1 mode of operation.

As mentioned, the section key generated by the transformation CTR_i is used to encrypt one block of the m-block message X with block size n, in accordance to the security bound for mode ECB, recalling that the transformation CTR_i ensures that for a given key K all the section keys K^i are different and therefore the outputs of E_{K^i} are all different.

Taking into account the probabilistic security bound of theorem 2 for this particular case one has that ECB-CTR+D_1 reach the bound $2^{\frac{n}{2}+1}\sqrt{\ln(\pi 2^n)}$ since $N_{max_{ECB}} = 1$ [23]. Hence, the security bound of ECB-CTR+D_1 is equal to the probabilistic bound of the CTR mode.

3.4 Performance analysis of ECB-CTR+D with AES

This section resumes the analysis of several experiments aimed to measure the average time of processing one megabyte of data in an ASUS personal computer with Intel®Core™i7-4790 @ 3.6 GHz (8 cores) processor and 16 GB of RAM using the standard AES [14] as underlying block cipher for both ECB and CTR components of the proposed mode of operation ECB-CTR+D.

We execute 30 independent runs for each possible master key size noticing that the section key size is restricted to 128 bits due to the output size of AES. Moreover, the probabilistic bound of the confidentiality mode ECB restrict the section size to one block of data (128 bits). However, we present the performance of ECB-CTR+D with respect to two additional section sizes, 2 blocks (256 bits) and 8 blocks (1024 bits) respectively.

Mode	Master Key Size (bits)	Section Size (blocks)	Execution Time (micro seconds)	CPU Cycles
ECB	128	-	42187.5	$1.13 \cdot 10^8$
ECB-CTR+D	128	1	90104.2	$2.41 \cdot 10^8$
ECB-CTR+D		2	66666.7	$1.78 \cdot 10^8$
ECB-CTR+D		8	48953.3	$1.3 \cdot 10^8$
ECB-CTR+D	192	1	192625	$5.11 \cdot 10^8$
ECB-CTR+D		2	131250	$3.47 \cdot 10^8$
ECB-CTR+D		8	84895.8	$2.27 \cdot 10^8$
ECB-CTR+D	256	1	214062.5	$5.68 \cdot 10^8$
ECB-CTR+D		2	144791.7	$3.78 \cdot 10^8$
ECB-CTR+D		8	87500	$2.32 \cdot 10^8$

Table 1: Performance of AES with mode of operation ECB-CTR+D.

As shown in table 1, the greater the section the closer the performance of ECB-CTR+D to the base mode ECB. The major drawback w.r.t the time that the algorithm took to process the whole input data lies in the generation of the section keys, due the constant re-schedule of keys that must be carried out to process each new section.

4 Conclusion

In this paper we present a prospective mode of operation COM-CTR+D to extend the lifetime of symmetric keys, under the assumption that the underlying block cipher is secure itself. Under this condition we show that the security of COM-CTR+D is higher than the security of the base mode COM, which can be any of the confidentiality modes of operations.

The construction of the re-keying mode COM-CTR+D is similar to that of the mode CTR-ACPKM, however, the section keys used for encrypting are different for each message contrary to CTR-ACPKM, which ensures that the encryption of the same plaintext result into different ciphertext for some instances of COM-CTR+D.

Our proposal can be used as external re-keyed encryption scheme considering the base mode security restrictions. Although in [10] and [12] it is recommended the use of a PRF function for the external case, in both cases, internal and external, a PRF function can be used for the key derivation of COM-CTR+D and still its proven security remains.

Finally, we recommend the use of this mode of operation whit dynamic block cipher algorithms for applications that require high security, in such a way that the internal transformations of the underlying block cipher depend on the section keys.

References

[1] Alekseev E., Goncharenko K. and Marshalko G., "Provably Secure Counter Mode with Related Key-based Internal Re-keying", (7th Workshop on Current Trends in Cryptology CTCrypt2018, Russia), 2018, 161-180.

[2] Alekseev E., Goncharenko K. and Marshalko G., "Provably Secure Counter Mode with Related Key-based Internal Re-keying", *Journal of Computer Virology and Hacking Techniques*, **16**:4 (2020).

[3] Alfonso, A. and Almeida, D. and Castro, L., "Statistical Assessment of two Rekeying Mechanisms applied to the Generation of Random Numbers", *Journal of Science and Technology on Information Security*, **2**:12 (2020), 38–44.

[4] Akhmetzyanova L. R., Alekseev E. K., Oshkin I. B., Smyshlyaev S. V. E. and Sonina L. A., "On the properties of the CTR encryption mode of Magma and Kuznyechik block ciphers with re-keying method based on CryptoPro Key Meshing", *Mathematical Aspects of Cryptography*, **8**:2 (2017), 39-50.

[5] Akhmetzyanova L. R., Alekseev E. K. and Smyshlyaev S. V., "Security bound for CTR-ACPKM internally re-keyed encryption mode", *IACR Cryptology ePrint Archive*, **2018**:950 (2018).

[6] Ahmetzyanova L. R., Alekseev E. K., Oshkin I. B. and Smyshlyaev S. V., "Increasing the Lifetime of Symmetric Keys for the GCM Mode by Internal Re-keying", *IACR Cryptology ePrint Archive*, **2017**:697 (2017).

[7] Bassham III, L. E. and Rukhin, A. L. and Soto, J. and Nechvatal, J. R. and Smid, M. E. and Barker, E. B. and Leigh, S. D. and Levenson, M. and Vangel, M. and Banks, D. L. and others, "Sp 800-22 rev. 1a. a statistical test suite for random and pseudorandom number generators for cryptographic applications", 2010.

[8] Bellare M., Desai A., Jokipii E. and Rogaway P., "A concrete security treatment of symmetric encryption", (38th Annual Symposium on Foundations of Computer Science, USA), 1997, 394-403.

[9] Bellare M and Rogaway P., "Introduction to modern cryptography, charper 4: Symmetric encryption", 2004.

[10] Bellare M., Abdalla M., "Increasing the Lifetime of a Key: A Comparative Analysis of the Security of Re-keying Techniques", *LNCS Advances in Cryptology — ASIACRYPT 2000*, (International Conference on the Theory and Application of Cryptology and Information Security, Japan), **1976**, Springer, Berlin, Heidelberg, 2000, 546-559.

[11] National Bureau of Standards, "DES modes of operation", *Federal Information Processing Standards Publication 81 (FIPS PUB 81)*, U.S. Department of Commerce, 1980.

[12] Chen L., "Recommendation for Key Derivation Using Pseudorandom Functions", *NIST special publication*, **800** (2008), 108.

[13] Daemen J., Rijmen V., *The design of Rijndael: AES the advanced encryption standard, Second Edition*, Springer, 2020.

[14] Federal information processing Standard, "Announcing the Advanced Encryption Standard (AES)", 2001, FIPS publication 197.

[14] *GOST 28147-89. Cryptographic Protection for Information Processing Systems - Government Standard of the USSR. Government Committee of the USSR for Standards*, 1989, In English.

[15] *GOST R34.12-2015. National Standard of the Russian Federation. Federal Agency on Technical Regulation and Metrology*, 2015, In English.

[16] Jansen C.J.A., Boekee D.E., "Modes of Blockcipher Algorithms and Their Protection Against Active Eavesdropping.", *LNCS Advances in Cryptology — EUROCRYPT' 87*, (Workshop on the Theory and Application of of Cryptographic Techniques, The Netherlands), **304**, Springer, Berlin, Heidelberg, 1988.

[17] Knudsen, L. R., "Block ciphers-analysis, design and applications", 1994, Aarhus Universitet, Department of Computer Science..

[18] L'Ecuyer, P. and Simard, R., "TestU01: A C library for empirical testing of random number generators", *ACM Trans. Math. Softw.*, **33**:4 (2007).

[19] Medwed M, Standaert F, Grosschadl J and Regazzoni J., "Fresh Re-keying: Security against Side-Channel and Fault Attacks for Low-Cost Devices", *AFRICACRYPT*, 2010, 279-296.

[20] Marsaglia, G., "Diehard Battery of Tests of Randomness", 1985.

[21] Noura H., Chehab A. and Couturier R., "Efficient and secure cipher scheme with dynamic key-dependent mode of operation", *Signal Processing: Image Communication*, **78** (2019), 448–464.

[22] Popov V., Kurepkin I. and Leontiev S., "Additional cryptographic algorithms for use with GOST 28147-89, GOST R 34.10-94, GOST R 34.10-2001, and GOST R 34.11-94 algorithms", 2006, IETF RFC 4357.

[23] Lavrikov I. and Shishkin V., "How much data may be safely processed on one key in different modes?", *Mathematical Aspects of Cryptography*, **10**:2 (2019), 125-134.

[24] Rogaway P., "Evaluation of some blockcipher modes of operation", 2011, Cryptography Research and Evaluation Committees (CRYPTREC) for the Government of Japan..

Appendix 1: Values of A for different base modes

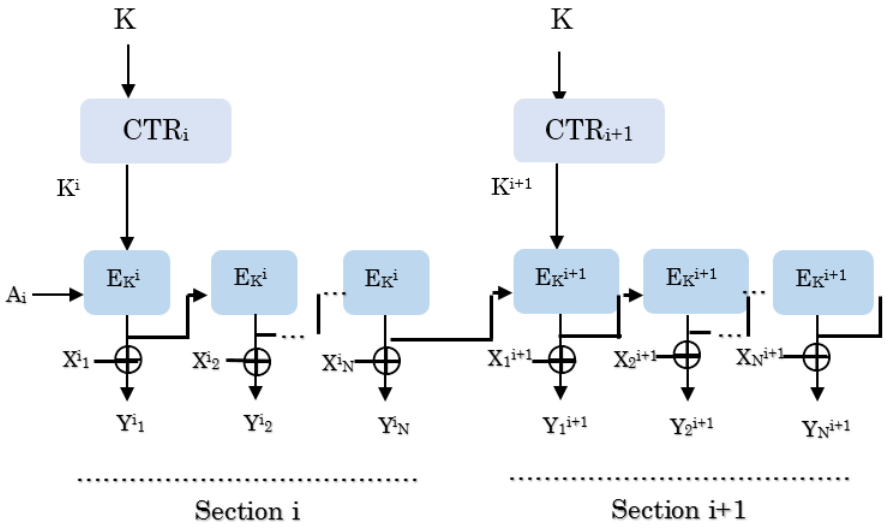

Figure 7: For the re-keyed mode of operation OFB-CTR+D the initial value of the section i+1 is calculated as $A_{i+1} = E^N_{K^i}(A_i)$

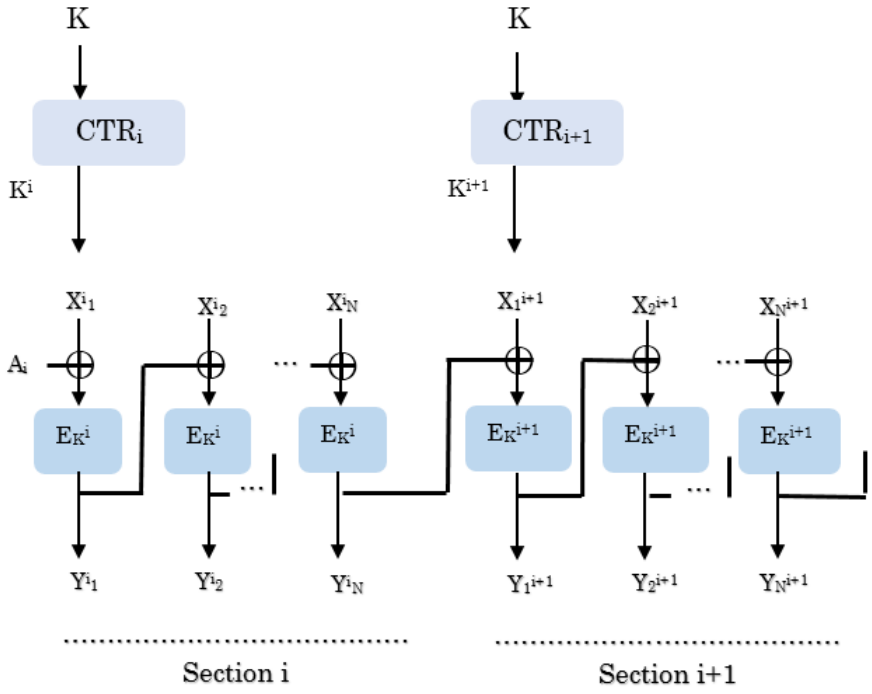

Figure 8: For the re-keyed mode of operation CBC-CTR+D the initial value of the section i+1 is calculated as $A_{i+1} = Y^i_N = E_{K^i}(X^i_N \oplus Y^i_{N-1})$

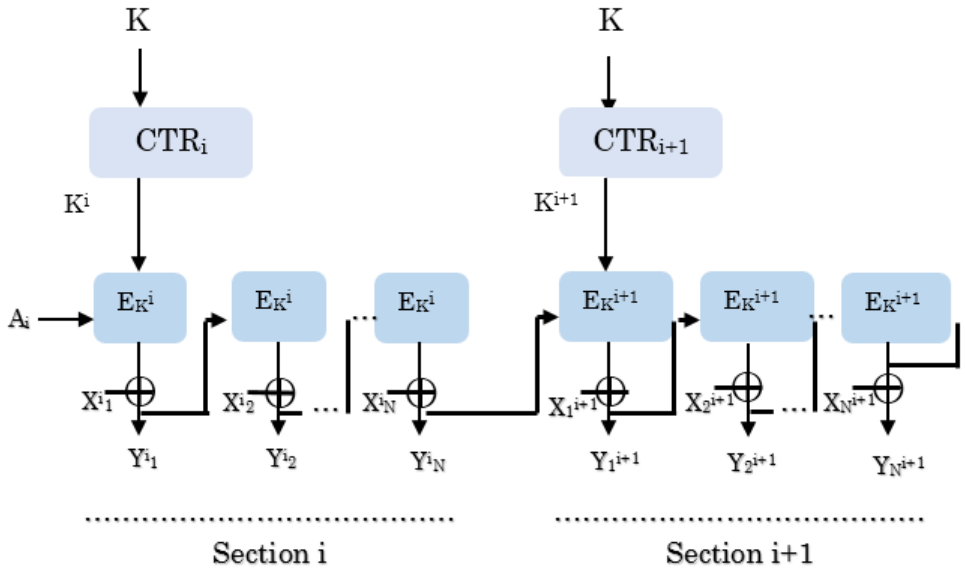

Figure 9: For the re-keyed mode of operation CFB-CTR+D the initial value of the section i+1 is calculated as $A_{i+1} = Y_N^i = X_N^i \oplus E_{K^i}(Y_{N-1}^i)$

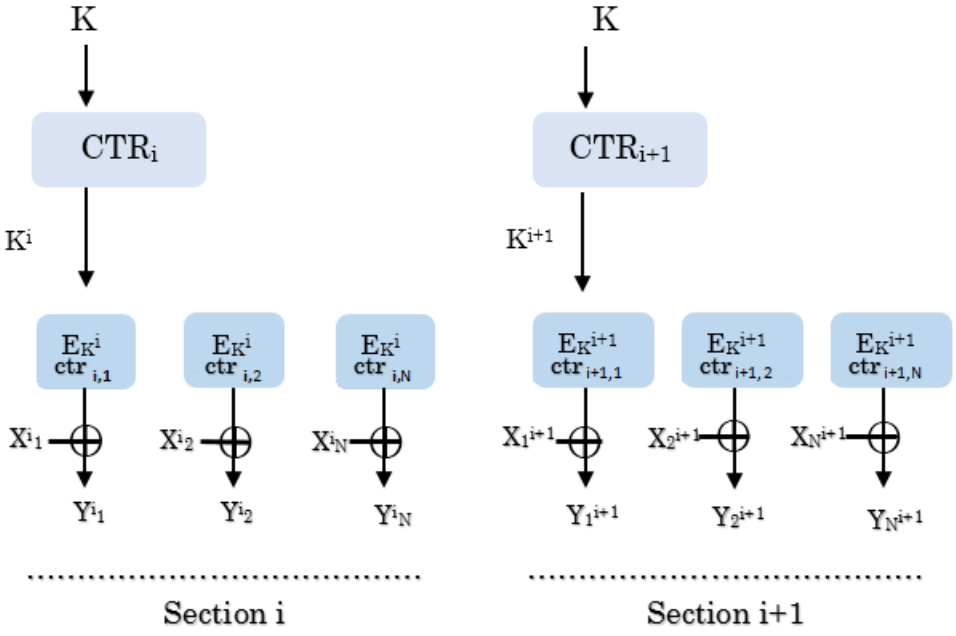

Figure 10: For the re-keyed mode of operation CTR-CTR+D the initial value of the section i+1 is calculated as $A_{i+1} = ctr_{i+1,1}$ where $ctr_{i+1,1}$ denote the initial counter of the (i+1)-th section of CTR mode.

Format-Preserving Encryption: a Survey

Kirill Tsaregorodtsev

[1]JSC "NPK Kryptonite", Russia
kirill94_12@mail.ru

Abstract

This article gives a survey on the format-preserving encryption, proposed algorithms, and attacks on them. Additionally, we propose a new format-preserving encryption scheme based on quasigroup operations.

Keywords: format-preserving encryption, quasigroup, provable security.

Introduction

Format-preserving encryption (FPE) is an encryption algorithm with the following property: the resulting ciphertext format must be the same as the format of plaintext. For instance, if we encrypt 9-digit individual insurance account number (SNILS), the result must be 9-digit ciphertext. In this case, the cipher must look like a random permutation on the given (usually small) domain. The small size of the domain makes it hard to provide both strong security properties in the presence of an adversary (which can usually obtain all ciphertexts of all points in the domain due to its size) and keep the resulting algorithm efficient as possible. Many attempts were made to build an FPE scheme using standardized solutions (such as AES) and well-studied principles (Feistel networks). The current state of the art is unsatisfactory: all proposed solutions for standardization in NIST and ISO/IEC are broken in some sense.

In this paper, we give a survey of suggested algorithms and attacks for FPE. Also, we propose a new approach for FPE based on quasigroups operation.

The structure of the paper is the following: in Section 1, we give a formal definition of format-preserving encryption, tweakable block cipher, and quasigroups. Section 2 is devoted to proposed algorithms for FPE. In section 3, we consider cryptanalysis of suggested solutions. In section 4, a new approach to the FPE algorithms is presented.

1 Preliminaries

1.1 Problem statement

The very first question is: why do we need format-preserving encryption (FPE) at all, and why is it not enough to use existing primitives such as block ciphers?

The request to preserve the format may be appropriate in the following situations:

1. Database structure may be incompatible with encrypted messages. If we want to keep the data encrypted, we either need to restructure the database or use FPE;
2. Some applications may require the data to be in a pre-defined format. In this case, we can rewrite an application from scratch or again use FPE;

Why do usual block ciphers not solve the problem? Block cipher acts as a permutation on the fixed length binary strings (for instance, $\{0,1\}^{128}$ for «Kuznyechik»). Even if the domain $\mathtt{Dom}$ is embedded in $\{0,1\}^n$, the result of encrypting $m \in \mathtt{Dom}$ is very unlikely to fall in the same subset: $E_k(m) \notin \mathtt{Dom}$ due to its relatively small size in the real-world situations and applications.

As an example, we can consider the case of credit card number (CCN):

Example 1. *CCN consist of the following numbers: 6 digits — bank number, 6 digits — account number, 3 digits — checksum, and all digits, except for account number (i.e., 9 out of 15), are publicly available. In this case the domain* $\mathtt{Dom} = \{0, \dots 9\}^6$.

Small domain size is dangerous. Due to a large number of possible input blocks for the usual block cipher (2^{64} or 2^{128}), it seems appropriate to ignore the type of attack when an adversary is able to collect the whole codebook (i.e. all pairs of the type $[m, E_k(m)]$). In example 1 we have $|\mathtt{Dom}| \approx 2^{20}$, which is perfectly feasible number to mount the dictionary attack.

Example 2 (Dictionary attack)**.** *CCNs from various banks can have the same account number. Using the bijective property of the cipher, we can be sure that the matching ciphertext blocks correspond to matching plaintext (see Table 1).*

Bank number	Account number	Checksum
012345	$E_k(000111)$	123
	$\updownarrow$ **same**	
987654	$E_k(000111)$	456

Table 1: Bank account database

1.2 Tweakable block ciphers and FPE

As shown earlier (Example 2), the small size of the domain is a severe threat. In order to prevent this sort of attacks a tweakable block cipher primitive can be used ([1]):

Definition 1. *Tweakable block cipher (TBC) is a pair of algorithms:*

$$E,\ D : \mathtt{Keys} \times \mathtt{Twk} \times \mathtt{Dom} \to \mathtt{Dom},$$

such that $D_k^t(E_k^t(m)) = m$*, where* $t \in \mathtt{Twk}$ *is a tweak (a block cipher parameter),* $k \in \mathtt{Keys}$ *is a key,* $m \in \mathtt{Dom}$ *is a message.*

Usually, the set of keys, tweaks, and messages for TBC are of the standard form $\{0,1\}^n$ for some n. The main idea of the construction is that $E_k^t(\cdot)$ are «weakly dependent» different permutations for different $t \in \mathtt{Twk}$.

Some properties of a tweak can be pointed out:

– Tweak acts like IV/nonce in the usual modes of encryption;

– The main goal of the tweak is to expand the set of possible permutations;

– Tweak may not be secret;

The property of «weak dependence» can be formalized in provable security framework (see [6, 7] for more details on provable security paradigm) as follows. Consider the experiments:

Algorithm 1 Experiment Left

function INIT
  for $t \in \mathtt{Twk}$ **do**
    $\pi^t \xleftarrow{R} Perm(\mathtt{Dom})$
function $\mathcal{O}(t, m)$
  return $\pi^t(m)$
function FIN(b')
  return b'

Algorithm 2 Experiment Right

function INIT
  $k \xleftarrow{\$} \mathtt{Keys}$
function $\mathcal{O}(t, m)$
  return $E_k^t(m)$
function FIN(b')
  return b'

Let $Adv_E^{TPRP}(\mathcal{A})$ be the advantage of the adversary $\mathcal{A}$ in the distinguishing attack, i.e.:

$$Adv_E^{TPRP}(\mathcal{A}) = \mathbb{P}[Right(\mathcal{A}) \to 1] - \mathbb{P}[Left(\mathcal{A}) \to 1].$$

The probability $\mathbb{P}[\cdot]$:

1. is taken over random choise of permutations $\pi^t \leftarrow^R Perm(\mathtt{Dom})$ for different t and random coins of $\mathcal{A}$ (if any) — in case of Left experiment;
2. is taken over random choise of key $k \leftarrow^{\$} \mathtt{Keys}$ and random coins of $\mathcal{A}$ (if any) — in case of Right experiment;

Note that the adversary queries are of the form (t, m), i.e., t is not secret and under adversary control.

Example 3 (TBC)**.** *If F is the usual PRP-strong block cipher, then the following construction is TBC:*

$$E_k^t(m) := F_k(t \oplus F_k(m)).$$

The paper [1] gives the following estimation:

$$Adv_E^{TPRP}(q_t, q_e, t) \le Adv_F^{PRP}(2q_tq_e, q_tq_e + t) + O\left(\frac{(q_tq_e)^2}{2^n}\right),$$

where q_t is the number different tweaks used by $\mathcal{A}$, q_e is the (maximal) number of encryption operations per tweak, n is the length of the block (i.e. the domain is of the form $\mathtt{Dom} = \{0,1\}^n$).

1.3 Format-preserving encryption

Definition 2. *Format-preserving encryption is a tweakable block cipher with an arbitrary set $\mathtt{Dom}$.*

The main difference between TBC and FPE is that the domain $\mathtt{Dom}$ is usually of the standard form for TBC (i.e. $\mathtt{Dom} = \{0,1\}^{128}$), but in case of FPE we are interested in «atypical» domains, such as $\mathtt{Dom} = \{0, \ldots, 9\}^6$ for CCN. Also it is possible for FPE scheme to have an empty tweak space, i.e. $\mathtt{Twk} = \emptyset$. Disk encryption with the block size of 512 bits can be viewed as FPE scheme as well ([12]) with $\mathtt{Dom} = \{0,1\}^{512}$. More formal treatment of FPE is given in [12, 13].

The quest to develop a good FPE scheme for all sizes of domains seems hard at the moment. There are subtle issues involved in the case of small-size

domains: for instance, the adversary can run a full search on the small space. As a consequence, the cryptographic strength of the scheme strongly depends on the size of the domain.

First attempts to develop an encryption algorithm for arbitrary domain are described in [10, 11]. Three papers [15, 16, 17] proposed standardized solutions for FPE algorithms. In 2016 NIST recommendations were issued based on these proposals. After the publication, a series of papers with significant advances in cryptanalytic techniques were published ([24, 25, 26, 27, 28]), which lead to theoretical and even practical threats for proposed algorithms. The current state of the art is rather unsatisfactory: for domains with the size between $\sim 2^{20}$ to $\sim 2^{64}$ there is no good provably secure and efficient algorithm. For «tiny» domains, as well as for «huge» one, there exists provably secure schemes (see [13]).

1.4 Quasigroups

Definition 3 ([29])**.** *Quasigroup is a set Q with a binary operation on it $\circ : Q \times Q \to Q$, which obeys the following property: for each $a, b \in Q$ there exist unique $x, y \in Q$ such that:*

$$a \circ x = b, \qquad y \circ a = b.$$

In other words, operations of left and right multiplication

$$L_a : Q \to Q,\ L_a(x) = a \circ x$$

$$R_a : Q \to Q,\ R_a(y) = y \circ a$$

are bijections on Q.

Some applications of quasigroup theory to cryptography can be found in [30, 31].

We use the following measure of quasigroup complexity. Given some quasigroup Q, we want to measure how close the composition of quasigroup operations (for instance, left multiplications) to the random permutation on Q. To formalize this notion, we introduce the following Experiments (the λ parameter is analogous to security parameter in classical cryptography):

Algorithm 3 Experiment Left

1: **function** INIT(λ)
2: $\quad \pi \leftarrow^R Perm(Q)$
3: **function** $\mathcal{O}(m)$
4: $\quad$ **return** $\pi(m)$
5: **function** FIN(b')
6: $\quad$ **return** b'

Algorithm 4 Experiment Right

1: **function** INIT(λ)
2: $\quad k_1, \ldots, k_\lambda \leftarrow^R Q$
3: **function** $\mathcal{O}(m)$
4: $\quad$ **return** $k_1 \circ (k_2 \circ (\ldots (k_\lambda \circ m) \ldots)$
5: **function** FIN(b')
6: $\quad$ **return** b'

Again we introduce adversary $\mathcal{A}$, who tries to distinguish between random permutation and «structured» permutation. Let $Adv_Q^{PRP}(\mathcal{A})$ be the advantage of adversary $\mathcal{A}$ in the distinguishing attack, i.e.:

$$Adv_Q^{PRP}(\mathcal{A}) = \mathbb{P}[Right(\mathcal{A}) \to 1] - \mathbb{P}[Left(\mathcal{A}) \to 1].$$

Let $InSec(t, q)$ be the maximal advantage Adv_Q^{PRP} over all adversaries $\mathcal{A}$, whose running time does not exceed t and who uses no more than q oracle queries. We want this quantity to be as small as possible for the given t and q. The quantity **directly depends** on the structure of the quasigroup Q. The inappropriate choice of quasigroup (i.e., $Q = \mathbb{Z}_N$) can make the problem trivial to solve.

Why this problem can be hard at all? The reason for that is that there exists a class of groups (called polynomially complete quasigroups), for which the problem of deciding whether or not an equation over such a quasigroup has a solution is NP-complete [32].

Definition 4. *Quasigroup Q of size k is called functionally (polynomially) complete if the system of functions consisting of binary operation $\circ$ and all constants $x \in Q$ (considered as functions of arity 0) with the operation of superposition generate all possible functions over Q, i.e.*

$$[\{\circ\} \cup \{x \in Q\}] = P_k.$$

Polynomially complete qusigroups are actively studied [33, 34, 35]. There exists an algorithm (polynomial in the size of the quasigroup) that checks whether the quasigroup is polynomially complete [35]. The NP-completeness of decision problem ([32]) is an evidence in favor of hardness of problem 1.4 in the worst case. However, the problem is not (yet) studied in the average-case scenario, and it is not clear enough what the value of parameter λ should be, as well as whether the problem is hard in average case. As it is pointed out in the conclusion 5, this is one of the main vectors of future research.

2 Proposed FPE algorithms

This section gives an overview of algorithms suggested for NIST standardization (FF1-FF3), one of the ISO standardisation candidates FEA-2 and general techniques (cycle walking, prefix encryption) helpful in designing FPE algorithms. We do not cover wide-block/disk encryption schemes with the size of the domain larger than the block size of modern ciphers. Schemes based on principles other than Feistel networks are also not touched upon [38, 37, 36, 39].

2.1 FF1, FF3

The structure of both algorithms is the same: the semi-balanced Feistel network over the group $\texttt{Dom} = \mathbb{Z}_M \times \mathbb{Z}_N$, where $M \approx N$ (see [22] for more details).

The algorithm takes the key $k \in \texttt{Keys}$, the element to be encrypted $(A, B) \in \texttt{Dom}$, and the tweak $t \in \texttt{Twk}$ (usually tweak space $\texttt{Twk}$ is of the form $\{0,1\}^{tlen}$). One round of the encryption process transforms the pair via the following rule:

$$(A, B) \to (B, A \boxplus Q),$$

where $Q \in \mathbb{Z}_M$ is derived by the following rule:

$$Q = PRF_k(B, t, i, params),$$

where i is the round number, $params$ is some (non-secret) information, PRF is a pseudorandom function (see [5] for the definition of PRF). We omit some technical details here; see [22] for a full description of algorithms.

It was suggested that 10 rounds of Feistel network are enough for FF1 security and 8 rounds for FF3. The original paper ([15]) did not provide the full security proof of the scheme. The arguments in favor of the proposed schemes include Patarin papers on the (classical) Feistel networks (see [2], [3], [4]) and paper on the Feistel networks over groups $\mathbb{Z}_M \times \mathbb{Z}_N$ [12].

Also, a wrong choice of tweak mixing in the PRF function was made in the FF3 scheme, which leads to some specific attacks on the scheme. Slight modifications were proposed to mitigate these attacks.

2.2 FF2

Along with the two algorithms mentioned above (FF1, FF3), a third one (FF2) was initially proposed for standardization at NIST. Unfortunately,

a design flaw leads to a theoretical attack on FF2. In this subsection, we briefly describe the main idea of the algorithm FF2 (VAES3, [17]) and the corresponding attack ([23]).

The algorihm FF2 consists of two steps. The input to the algorithm is three parameters: $k \in \texttt{Keys}, t \in \texttt{Twk}, m \in \texttt{Dom}$.

1. Derive the secret key for the given tweak:

$$sk = E_k(t) \in \{0,1\}^{128};$$

2. Encrypt the message with obtained key

$$c = Feistel_{sk}(m);$$

The main problem of the algorithm is that the key length $|sk| = 128$ is too short to guarantee the strong security bound.

Assume that we have n ciphertexts of the form:

$$c_j = Feistel_{E_k(t_j)}(m),$$

then we can try different keys $sk_j \in \{0,1\}^{128}$ and obtain

$$c'_j = Feistel_{sk_j}(m).$$

If some c'_j is the same as c_i for some i, then with high probability we will have $sk_j = E_k(t_j)$, i.e., we can recover the derived key for the given tweak t_j without knowing the value of master-key k.

If the number of ciphertexts $n = 2^u$, then the collision is expected to occur after 2^{128-u} steps. Even though the attack is somewhat hypothetical, it was decided to finalize standard NIST SP 800-38G without FF2 (but see also more recent work on FF2 [18]).

2.3 FEA-2 algorithm

In paper [21], a new family of tweakable block ciphers was proposed, based on Feistel networks. Some remarkable features of the proposition are the following:

1. In contrast with the previous algorithms (FF1 — FF3), this proposition suggest to embed tweak at the primitive layer, i.e., tweak is used in each encryption round.

2. Unlike simple block ciphers, FEA-2 provide the ability to encrypt messages of various lenghts, $\texttt{Dom} = \{0,1\}^n$, where $n \in \{8, 9, \ldots 128\}$.

3. The number of rounds in Feistel network depends on the block size and starts with 18.

4. The key lenght is not fixed: $klen \in \{128, 192, 256\}$.

5. Encryption on the domain $\texttt{Dom} = \{1, \ldots, N\}$ is done via embedding $\texttt{Dom} \subseteq \{0,1\}^n$ combined with the cycle walking idea (see subsection 2.5 for details).

The article ([21]) is devoted to the consideration of applications of various cryptanalytic techniques to the FEA-2 algorithm. Linear and differential analysis, as well as related-key attacks, were investigated. Additionally, threats specific to Feistel networks over small domains were analyzed. The authors claimed that for the domains of size greater than 2^8 the proposed attacks require at least 2^{64} encryptions on different parameters $t \in \texttt{Twk}$.

One round of the proposed scheme is depicted below (taken from the original paper [21]):

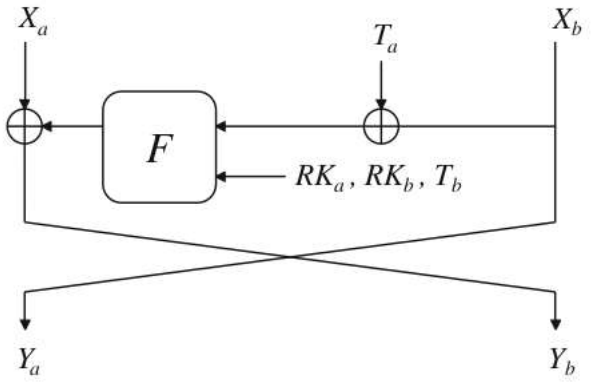

Fig 1: one round of FEA-2

The following notation is used:

– $X_a \,||\, X_b$ — left and right blocks of the message;

– $T_a \,||\, T_b$ — left and right blocks of the tweak;

– $RK_a \,||\, RK_b$ — left and right blocks of the round key;

It is assumed that $|T_a| = |X_b|, |T_a| + |T_b| = 128 = |RK_a| = |RK_b|$).

2.4 Prefix encryption

If the domain $\texttt{Dom}$ is small enough, then we can use the following idea:

1. We will compute the number list:

$$S = (E_k(0), \ldots, E_k(N-1));$$

2. To encrypt the message $m \in \mathbb{Z}_N$ we will map m to the position of $E_k(m)$ in the sorted list.

This method is provably secure but requires $O(N)$ encryption operations at the initial step and $O(N)$ memory to store the table.

2.5 Cycle walking

If the domain $\texttt{Dom}$ has the property that $\frac{|\texttt{Dom}|}{2^n} \leq 1$ is close to 1 for some standard block size n (for instance, $n = 64$ or $n = 128$), then the following approach works:

1. For $m \in \texttt{Dom}$ compute $c \leftarrow E_k(m)$.
2. If $c \in \texttt{Dom}$, then m maps to c.
3. If $c \notin \texttt{Dom}$, then $c \leftarrow E_k(c)$ and go to step 2.

It can be shown that the algorithm is provably secure ([11]). The expected number of encryption operations (before one obtains $c \in \texttt{Dom}$) is determined by the quantity $\frac{2^n}{|\texttt{Dom}|}$.

2.6 FNR and DTP algorithms

Cisco company proposed FNR algorithm (see [19]), based on the Feistel cipher with two additional permutations. Protegrity company suggested **DTP** algorithm ([20]). In [26] it was shown that:

– FNR is slightly worse than FF1 and FF3;

– DTP has serious weaknesses, which lead to a total break;

2.7 Techniques for tiny-size domains

The following methods for tiny-space domains with the size $|\texttt{Dom}| = N$ are mentioned in [13]:

1. Exhaustive permutation numbering: each key k is mapped to a number between 1 and $N!$, the point $x \in \texttt{Dom}$ is mapped to $\pi_k(x)$.

2. Knuth-Fisher-Yates shuffle: to shuffle the array of numbers one has to repeatedly choose an element from a decreasing prefix and moving it to the end. The element $x \in \texttt{Dom} = \{1, \ldots, N\}$ is mapped into the position of x in the shuffled array.

3. Prefix encryption method mentioned earlier 2.4

Each of the methods is provably secure, but encryption time (or set-up time) is proportional to the size of the domain, hence the limitation on the size.

3 Summary of cryptanalysis of proposed algorithms

This section summarizes cryptanalysis on FF1, FF3, and generic algorithms based on Feistel networks.

Currently, there are two types of attacks on FPE algorithms:

– The first type of cryptanalytic attacks exploits the wrong design of tweak mixing (specific for FF3). In these attacks, the adversary adaptively chooses plaintexts to be encrypted on two selected $t_1, t_2 \in \texttt{Twk}$.

– The second type uses the intrinsic feature of Feistel network over small domains. The general idea is that the proposed number of rounds is not enough to hide the plaintext statistics: there is a slight bias after one round of Feistel network, which can be boosted using different tweaks $t \in \texttt{Twk}$ for the same message. This asymmetry can be exploited to mount the practical distinguishing attack (or even key recovery).

The specificity of attacks on FPE algorithms is that the number of required texts formally exceeds the domain size, but in fact only a minimal (even constant) number of texts are required for each $t \in \texttt{Twk}$. This fact was not reflected in the original adversary model [12]. All the proofs were obtained in a weaker model, in which the adversary cannot make the number of requests to the oracle that exceeds the domain size.

The following notation in table 3 is used:

n — bitsize of one part of the message $m \in \texttt{Dom}$, i.e. $\texttt{Dom} = \{0,1\}^{2n}$;

$N = 2^n$ — number of different parts of the message;

r — number of rounds in Feistel network;

The most recent attack ([28]) has a significant impact on FEA-type ciphers. The results are presented in table 2.

Algorithm	Resourses	Threat
FF1, $klen = 128, r = 10$	$q = 2^{60}, t = 2^{70}$	Distinguishing attack
FF3-1, $klen = 128, r = 8$	$q = 2^{80}, t = 2^{100}$	Distinguishing attack
FEA-2, $klen = 128, r = 18$	$q = 2^{80}, t = 2^{84}$	Distinguishing attack
FEA-2, $klen = 256, r = 24$	$q = 2^{80}, t = 2^{84}$	Distinguishing attack
FEA-1, $klen = 192, r = 14$	$q = 2^{36}, t = 2^{136}$	Key recovery
FEA-1, $klen = 256, r = 16$	$q = 2^{48}, t = 2^{136}$	Key recovery
Generic Feistel network	$q = 2^{n(r-4)}, t = 2^{n(r-3)}$	Distinguishing attack

Table 2: Recent attack on FPE algorithms [28]

Work	Resources	Threat	Comments
2004, [3]	$q_t = N^{r-2}$ encryptions queries on different $t \in \texttt{Twk}$, two messages per tweak ($q_e = 2$), time complexity $t \approx q_t q_e$	Distinguisher **generic Feistel network**	Attack distinguishes Feistel network output from a random string
2015, [23]	q_t encryptions queries on different $t \in \texttt{Twk}$ for the same $m \in \texttt{Dom}$, $q_e = 1$, time complexity $t \approx \frac{2^{128}}{q}$	Subkey recovery for some tweak, **FF2**	The attack is not adaptive, the knowledge of m is required
2016, [24]	$q_t = \mathcal{O}(n \cdot N^{r-2})$ encryptions queries on different $t \in \texttt{Twk}$, 3 messages per tweak ($q_e = 3$), time complexity $t \approx q_t$	Message recovery, **generic Feistel network**	1. The adversary knows ciphertexts of three different messages (x, x', x^*) under tweaks $t_1, \ldots, t_q$, and recovers the message x. 2. The message x' is fully known to the adversary but unrelated to x. 3. x^* and x share a common right side; only the left side of x^* is known to the adversary. 4. The attack is not adaptive; only the knowledge of plaintexts is required.

2017, [25]	$q_e = \mathcal{O}(N^{\frac{11}{6}})$ encryption queries on two tweaks $t_1, t_2 \in \texttt{Twk}$ $(q_t = 2)$; time complexity $t = \mathcal{O}(N^5)$	Entire codebook recovery for t_1, t_2 for **FF3**.	1. The adaptive choice of messages is required. 2. We assume that the adversary can control the choice of $t \in \texttt{Twk}$. The attack does not work if the adversary does not have complete control over t. Partial truncation of the tweak can be applied (as shown in the [25]) to prevent this threat.
2018, [26]	$q_t = \mathcal{O}(N^{r-4}(n \cdot N + p))$ different tweaks, Number of plaintexts per tweak: $q_e = \mathcal{O}(n \cdot N)$, time complexity $t = \mathcal{O}(n \cdot N^{r-2}(n + p))$	Recovery of multiple messages $m_1, \ldots, m_p$ **generic Feistel network**	1. The attack is not adaptive; only the knowledge of plaintexts is required. 2. It is assumed that the adversary knows ciphertexts for τ known plaintexts $x_1, \ldots x_\tau$ and for p messages (plaintexts) under attack $m_1, \ldots m_p$ for q different tweaks. 3. It is assumed that right halves of $x_1, \ldots, x_\tau$ comprise all possible right halves of messages. 4. The correlation between $x_1, \ldots, x_\tau$ and $m_1, \ldots m_p$ is not required.
2019, [27]	$q_e = \mathcal{O}(N^{\frac{11}{6}})$ encryption queries on two tweaks $t_1, t_2 \in \texttt{Twk}, q_t = 2$; time complexity $t = \mathcal{O}(N^{\frac{17}{6}})$	Entire codebook recovery for t_1, t_2 for **FF3**.	1. The attack is the strengthened version of [25] 2. The adaptive choice of messages is required. 3. The attack does not work if the adversary cannot obtain full control over t.

Table 3: Attacks on FPE algorithms

4 Quasigroup based FPE

In this section, we describe one possible approach to FPE. Due to the cycle-walking technique, we can limit our consideration to domains of the particular form $\texttt{Dom} = \{0,1\}^n$ for some «small» n.

Let Q be the quasigroup over the set $\texttt{Dom}$. We will use the following method: given the key $k \in \texttt{Keys}$ and the tweak $t \in \texttt{Twk}$, we will first use some keyed pseudorandom generator PRG (see [5, 8]) to produce a sequence of «random-looking» and «independent» elements $q_i \in Q, i = 1, \ldots, \lambda$, where

λ is the parameter of the scheme and is chosen based on the quasigroup structure (it is selected in such a way that the distinguishing problem (as it is stated in 1.4) is hard to solve).

Then we will encrypt our message $m \in \texttt{Msg}$ using the quasigroup operation. Some possible variants might be:

$$m \to L_{q_\lambda}(\dots L_{q_1}(m)\dots) = q_\lambda \circ (\dots \circ q_2 \circ (q_1 \circ m)\dots), \tag{1}$$

$$m \to R_{q_\lambda}(\dots R_{q_1}(m)\dots) = (\dots (m \circ q_1) \circ q_2 \dots) \circ q_\lambda, \tag{2}$$

$$m \to D_{q_\lambda}(\dots D_{q_1}(m)\dots). \tag{3}$$

In equation (3), we are using the following agreement: the operation D_{q_i} equals L_{q_i} if i-th bit of output of some random generator (for instance, based on values k and t) is equal to 0, and R_{q_i} otherwise. In this case, equations (1) and (2) are special cases of (3). We describe several possible variants of the scheme because only left (or only right) multiplication might be vulnerable to attacks.

Now we will describe the variant of scheme (1) in more detail. We will present a series of Experiments, where Experiment 0 is the original cryptosystem, and the final Experiment is (semantically) Left Experiment from 1.2. Our goal is to show that (under assumptions on PRG and 1.4) the scheme is secure in the TPRP model (as it is stated in Experiment 1.2).

Theorem 1. *Let q_t be the maximal number of different tweaks, q_e be the maximal number of encryption queries per tweak, t is the number of operations (running time). Then:*

$$InSec^{TPRP}(t, q_t, q_e) \leq$$

$$\leq InSec^{PRG}(q_t, t + \lambda q_t q_e) + q_t InSec_Q^{PRP}(q_e, t + (1 + \lambda^2) q_e q_t).$$

Proof. Denote by $L_{q_\lambda \dots q_1}(m)$ the following operation:

$$L_{q_\lambda \dots q_1}(m) = q_\lambda \circ (\dots \circ q_2 \circ (q_1 \circ m)\dots).$$

The first experiment Exp^0 is the original cryptoalgorithm. The transition from Exp^0 to Exp^1 is done via replacement of PRG by random choice of $q_i \in Q$. The last transition from Exp^1 to Exp^2 is done by replacing operation $L_{q_\lambda \dots q_1}(m)$ with operation $\pi^t(m)$.

Algorithm 5 Exp^0

function INIT
  $k \leftarrow^R$ `Keys`
  $qs = \{\}$
function $\mathcal{O}(t, m)$
  if $t \notin qs.keys$ **then**
    $q_1, \ldots, q_\lambda \leftarrow^R PRG_k(t)$
    $qs[t] = (q_1, \ldots, q_\lambda)$
  $q_1, \ldots, q_\lambda \leftarrow qs[t]$
  $res \leftarrow L_{q_\lambda \ldots q_1}(m)$
  return res
function FIN(b')
  return b'

Algorithm 6 Exp^1

function INIT
  $k \leftarrow^R$ `Keys`
  $qs = \{\}$
function $\mathcal{O}(t, m)$
  if $t \notin qs.keys$ **then**
    $q_1, \ldots, q_\lambda \leftarrow^R Q$
    $qs[t] = (q_1, \ldots, q_\lambda)$
  $q_1, \ldots, q_\lambda \leftarrow qs[t]$
  $res \leftarrow L_{q_\lambda \ldots q_1}(m)$
  return res
function FIN(b')
  return b'

Algorithm 7 Exp^2

function INIT
  $qs = \{\}$
function $\mathcal{O}(t, m)$
  if $t \notin qs.keys$ **then**
    $\pi^t \leftarrow^R S_Q$
    $qs[t] = \pi^t$
  $\pi^t \leftarrow qs[t]$
  $res \leftarrow \pi^t(m)$
  return res
function FIN(b')
  return b'

For any adversary $\mathcal{A}$, we can write the following equality:

$$Adv_Q^{TPRP}(\mathcal{A}) = \mathbb{P}[Exp^0(\mathcal{A}) \to 1] - \mathbb{P}[Exp^2(\mathcal{A}) \to 1] =$$

$$= \Big(\mathbb{P}[Exp^0(\mathcal{A}) \to 1] - \mathbb{P}[Exp^1(\mathcal{A}) \to 1]\Big) +$$

$$+ \Big(\mathbb{P}[Exp^1(\mathcal{A}) \to 1] - \mathbb{P}[Exp^2(\mathcal{A}) \to 1]\Big).$$

Then we can bound from above each of the brackets.

The first difference is small due to the fact that PRG is a good pseudorandom generator. Namely, if $\mathcal{A}$ can distinguish between Exp^0 and Exp^1 with high probability, then it can be used to attack pseudorandomness of PRG. We can create an adversary $\mathcal{B}$, who runs $\mathcal{A}$ as a subroutine. When $\mathcal{A}$ asks for encryption of m under fresh tweak t, $\mathcal{B}$ is calling his oracle on the input t and gets values $q_1, \ldots, q_\lambda$. He saves it in the memory, computes $res \leftarrow L_{q_\lambda \ldots q_1}(m)$ and gives it to $\mathcal{A}$. At the end of the experiment $\mathcal{B}$ outputs

the resulting bit of $\mathcal{A}$. The running time of $\mathcal{B}$ is the running time of $\mathcal{A}$ plus the time needed for simulation: $t_{\mathcal{B}} \leq t_{\mathcal{A}} + \lambda q_e q_t$ (we assume here for simplicity that operation $a \circ b$ can be done in one step). The number of oracle queries $q_{\mathcal{B}} = q_t$. Thus, we obtain the following estimate:

$$\Big(\mathbb{P}[Exp^0(\mathcal{A}) \to 1] - \mathbb{P}[Exp^1(\mathcal{A}) \to 1]\Big) \leq Insec^{PRG}(q_t, t + \lambda q_t q_e)$$

The second transition can be done via standard hybrid argument technique ([9]). We have at most q_t queries for the fresh $(q_1, \ldots, q_\lambda)$, and for each tuple $(q_1, \ldots, q_\lambda)$ we ask no more than q_e encryption queries. We will replace all queries of encrypting m on i-th tweak t_i (i.e. all $L_{q_\lambda \ldots q_1}(m)$, q_i are chosen uniformly from Q) by $\pi(m)$ (where π is chosen uniformly from S_Q).

If there is an adversary $\mathcal{A}$, who can distinguish this replacement, then we can use $\mathcal{A}$ to construct $\mathcal{B}$, who will attack PRP-property of $L_{q_\lambda \ldots q_1}(\cdot)$:

$(m_1^{t_1}, t_1)\ (m_1^{t_2}, t_2) \ldots (m_1^{t_{i-1}}, t_{i-1})$	$(m_1^{t_i}, t_i)$	$(m_1^{t_{i+1}}, t_{i+1}) \ldots (m_1^{t_{q_t}}, t_{q_t})$
$\vdots$	$\vdots$	$\vdots$
Simulated via choosing $(q_1, \ldots, q_\lambda) \leftarrow^R Q$	Oracle queries	Simulated via choosing $\pi^t \leftarrow^R S_Q$
Running time $\leq \lambda^2 q_e q_t$	$q_{\mathcal{B}} \leq q_e$	Running time $\leq q_e q_t$

The running time of $\mathcal{B}$ is the time of $\mathcal{A}$ plus the time to simulate all other environment (except for the queries on tweak t_i):

$$t_{\mathcal{B}} \leq t_{\mathcal{A}} + \lambda^2 q_e q_t + q_e q_t$$

(we assume here for simplicity that operation $a \circ b$ and random choice of element $q \leftarrow^R Q$ can be done in one step). The number of oracle queries $q_{\mathcal{B}} \leq q_e$. Thus, we obtain the following estimate:

$$\mathbb{P}[Exp^1(\mathcal{A}) \to 1] - \mathbb{P}[Exp^2(\mathcal{A}) \to 1] \leq q_t \cdot InSec_Q^{PRP}(q_e, t + (1 + \lambda^2) q_e q_t).$$

Combining both estimates we get the statement of the theorem. □

We stress out that currently there are no concrete estimates of the hardness of the problem 1.4, hence no concrete bounds on $InSec^{TPRP}(t, q_t, q_e)$.

5 Conclusion

This article gives a survey on the FPE, suggested algorithms (FF1-FF3, FEA-2), and attacks on them. We propose a new cryptoalgorithm based on quasigroup operations.

Further areas of research may include:

1. Consideration of specific classes of quasigroups as a basis for proposed cryptosystem (with an emphasis on the polynomially complete quasigroups);

2. Estimating the hardness of problem formulated in 1.4 based on existing results on NP-completeness of problem of deciding whether the equation has the solution over polynomially complete quasigroup [32];

3. Implementing the cryptosystem over specific quasigroups and estimating statistical properties of resulting algorithms;

References

[1] Liskov M., Rivest R., Wagner D., "Tweakable Block Ciphers", *Journal of cryptology*, **24**:3 (2011), 588–613.

[2] Patarin J., "Luby-Rackoff: 7 Rounds Are Enough for $2^{n(1-\epsilon)}$ Security", *Advances in Cryptology*, 2003, 513–529.

[3] Patarin J., "Security of Random Feistel Schemes with 5 or More Rounds", *Advances in Cryptology*, 106–122.

[4] Nachef V., Patarin J., Volte E., *Feistel Ciphers — Security Proofs and Cryptanalysis*, Springer, 2017.

[5] Bellare M., Rogaway P., *Introduction to modern cryptography*, UCSD CSE, 2005.

[6] Katz J., Lindell Y., *Introduction to modern cryptography*, CRC press, 2020.

[7] Guo F., Susilo W., Mu Y., *Introduction to security reduction*, Springer, 2018.

[8] Goldreich O., *Foundations of cryptography: volume 1, basic tools*, Cambridge university press, 2007.

[9] Shoup V., "Sequences of games: a tool for taming complexity in security proofs", *Cryptology ePrint Archive, Report 2004/332*, 2004, `https://eprint.iacr.org/2004/332`.

[10] Brightwell M., Smith H., "Using datatype-preserving encryption to enhance data warehouse security", *20th National Information Systems Security Conference Proceedings (NISSC)*, 141–149.

[11] Black J., Rogaway P., "Ciphers with Arbitrary Finite Domains", *Proceedings of the The Cryptographer's Track at the RSA Conference on Topics in Cryptology*, 2002, 114–130.

[12] Bellare M., Ristenpart T., Rogaway P., Stegers T., "Format-preserving encryption", *International workshop on selected areas in cryptography*, 2009, 295–312.

[13] Rogaway P., "A Synopsis of Format-Preserving Encryption", 2010.

[14] Stallings W., "Format-preserving encryption: Overview and NIST specification", *Cryptologia*, **41**:2 (2017), 137–152.

[15] Bellare M., Rogaway P., Spies T., "The FFX Mode of Operation for Format-Preserving Encryption", *NIST submission*, **20** (2010).

[16] Brier E., Peyrin T., Stern J., "BPS : a Format-Preserving Encryption Proposal", *NIST submission*, 2010.

[17] Vance J., "VAES3 scheme for FFX: An addendum to The FFX mode of operation for Format Preserving Encryption", *NIST submission*, 2011.

[18] Vance J., Bellare M., "An extension of the FF2 FPE Scheme", *NIST submission*, 2014.

[19] Sashank D., Fluhrer S., "FNR: Arbitrary Length Small Domain Block Cipher Proposal", *International Conference on Security, Privacy, and Applied Cryptography Engineering*, 2014, 146–154.

[20] Mattsson U., "Format-controlling encryption using datatype-preserving encryption", *IACR Cryptology ePrint Archive*, 2009.

[21] Lee J.-K., Koo, B., Roh D. Kim W.-H., Kwon D., "Format-Preserving Encryption Algorithms Using Families of Tweakable Blockciphers", *International Conference on Information Security and Cryptology*, 2014, 132–159.

[22] Dworkin M., "Recommendation for Block Cipher Modes of Operation: Methods for Format-Preserving Encryption", *NIST Special Publication 800-38G*, 2016.

[23] Dworkin M., Perlner R., "Analysis of VAES3 (FF2)", *Cryptology ePrint Archive, Report 2015/306*, 2015, `https://eprint.iacr.org/2015/306`.

[24] Bellare M., Hoang V. T., Tessaro S., "Message-Recovery Attacks on Feistel-Based Format Preserving Encryption", *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, 444–455.

[25] Durak F. B., Vaudenay S., "Breaking the FF3 Format-Preserving Encryption Standard over Small Domains", *Advances in Cryptology*, 2017, 679–707.

[26] Hoang V. T., Tessaro S., Trieu N., "The Curse of Small Domains: New Attacks on Format-Preserving Encryption", *Advances in Cryptology*, 2018, 221–251.

[27] Hoang V. T., Miller D., Trieu N., "Attacks Only Get Better: How to Break FF3 on Large Domains", *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2019, 85–116.

[28] Dunkelman O., Kumar A., Lambooij E., Sanadhya S. K., "Cryptanalysis of Feistel-Based Format-Preserving Encryption", *Cryptology ePrint Archive, Report 2020/1311*, 2020, `https://eprint.iacr.org/2020/1311`.

[29] Keedwell A., Denes J., *Latin squares and their applications, 2nd edition*, Burlington, North Holland, 2015, 438 pp.

[30] Shcherbacov V., *Elements of quasigroup theory and applications*, CRC Press, 2017

[31] Glukhov M., "Some applications of quasigroups in cryptography", *Applied Discrete Mathematics*, **2** (2008), 28–32, In Russian.

[32] Horváth G., Nehaniv C., Szabó C., "An assertion concerning functionally complete algebras and NP-completeness", *Theoretical computer science*, **407**:1-3 (2008), 591–595.

[33] Artamonov V. A., Chakrabarti S., Pal S. K., "Characterization of polynomially complete quasigroups based on Latin squares for cryptographic transformations", *Discrete Applied Mathematics*, **200** (2016), 5–17.

[34] Artamonov V. A., "Quasigroups and their applications", *Chebyshevskii Sbornik*, **19**:2 (2018), In Russian.

[35] Galatenko A., Pankratiev A., Rodin S., "Polynomial completeness of finite quasigroups", *Intell. Syst.*, **23**:1 (2019), 81–87, In Russian.

[36] Chang D., Ghosh M., Gupta K. C., Jati A., Kumar A., Moon D., Ray I. G., Sanadhya S. K., "SPF: a new family of efficient format-preserving encryption algorithms", *International Conference on Information Security and Cryptology*, 2016, 64–83.

[37] Granboulan L., Pornin T., "Perfect block ciphers with small blocks", *International Workshop on Fast Software Encryption*, 2007, 452–465.

[38] Thorp E. O., "Nonrandom shuffling with applications to the game of Faro", *Journal of the American Statistical Association*, **68** (1973), 842–847.

[39] Morris B., Rogaway P., Stegers T., "Deterministic encryption with the Thorp shuffle", *Journal of Cryptology*, **31**:2 (2018), 521–536.

Symmetric Cryptography
Permutations and Substitutions

On Differential Uniformity of Permutations Derived Using a Generalized Construction

Denis Fomin and Maria Kovrizhnykh

Higher School of Economics, Russia
dfomin@hse.ru, makovrizhnykh@gmail.com

Abstract

The work is dedicated to the theoretical substantiation of a directed search for 8-bit permutations with given cryptographic properties: differential uniformity and nonlinearity. The statements about the partition into equivalence classes of the set of vector Boolean functions derived using generalized construction are formulated and proved. The statements that allow one to reject functions from equivalence classes either by a high differential uniformity or since they are not permutations are justified. The results of this work can be used to construct permutations with specified cryptographic properties, ensuring the resistance of encryption algorithms against the linear and differential methods of cryptographic analysis.

Keywords: Boolean function, permutation, differential uniformity.

Introduction

Vector Boolean functions (S-boxes) are one of the main primitives of modern symmetric ciphers that provide Shannon's confusion [1]. S-boxes must have cryptographic properties that guarantee the impossibility of using differential and linear methods of cryptographic analysis. Thus, S-boxes with high nonlinearity can ensure the cipher resistance to linear cryptographic analysis, since they can not be effectively replaced by a linear analog of the same or less dimension. Moreover, S-boxes with the minimum possible differential uniformity are used for constructing cryptographic algorithms that are resistant to differential analysis.

Construction of $n \geq 8$ bits permutations with given cryptographic properties is a difficult and urgent task, which is confirmed by a large number of the latest scientific publications and reports at all-Russian and international conferences (e.g. [2, 3, 4, 5, 6, 7, 8, 9, 10]) dedicated to this theme. The known approaches to constructing permutations can be divided into explicit algebraic methods, pseudo-random generation, and heuristic algorithms (see, e.g., an overview in [2]).

The idea of a combination of the above approaches seems promising, in particular, the use of functional circuits to derive permutations using functions of lower dimension (see, e.g., an overview in [9]). Moreover, such schemes usually have some parameters, the appropriate choice of which can improve the cryptographic properties of constructed permutations.

Thus, in the work [4] a new construction of 8-bit S-boxes with nonlinearity up to 108, differential uniformity 6 or 8, algebraic degree 7, and algebraic immunity 3 is proposed. It utilizes an inversion in the field $\mathbb{F}_{2^4}$ and two arbitrary permutations of the space V_4.

In articles [5, 6] new schemes based on the well-known Feistel and Lai-Massey structures for generating permutations of dimension $n = 2k$, $k > 2$ are presented. The proposed constructions use inversion in the field $\mathbb{F}_{2^k}$, an arbitrary k-bit non-bijective function (which has no pre-image for 0), and any k-bit permutation. New 8-bit permutations without fixed points, which have the same strong combination of cryptographic properties as in [4] are introduced.

In the paper [7] new classes of 8-bit permutations based on the butterfly structure are proposed. It is shown that there are at least 36 new constructions for permutations that have the nonlinearity 108, differential uniformity 6, algebraic degree 7, and graph algebraic immunity 3.

The papers [9, 10] extend the methods of constructing permutations from [7] to the case of an arbitrary vector space V_{2m} and theoretically substantiate the experimental results obtained in [7]. TU-decomposition described in [11, 12] is used as a functional circuit. Necessary and, in some cases, sufficient conditions for the resulting permutation to have given nonlinearity, algebraic degree, and differential uniformity are proved. Also, new generalized construction of vector functions is described. It utilizes monomial permutations as the basic constituent elements. In the case $m = 4$, 768 tuples of parameters of the generalized construction were experimentally found, using which, with the correct choice of auxiliary 4-bit permutations, 8-bit permutations with nonlinearity 108, differential uniformity 6, and algebraic degree 7 can be obtained.

The purpose of this work is the theoretical substantiation of a directed search for 8-bit permutations with given cryptographic properties: differential uniformity and nonlinearity, among vector Boolean functions obtained using a generalized construction that admits TU-decomposition.

This paper is structured as follows. Section 1 contains the main definitions and notations used in the work. In Section 2 we consider a generalized construction of $(2m, 2m)$-function and show that this construction admits

TU-decomposition. In Section 3 we introduce an equivalence relation on the set of all vector Boolean functions defined by generalized construction. Each equivalence class is determined by a tuple of exponents of monomial permutations. In Section 4, we prove several statements that allow us to reject the equivalence classes of 8-bit S-boxes that do not contain permutations with a low differential uniformity. Non-rejected classes can be used to generate 6-uniform 8-bit permutations with 108 nonlinearity.

1 Definitions and Notations

Let V_n be n-dimensional vector space over the field of two elements $\mathbb{F}_2$, $V_n^\times = V_n \setminus \{0\}$. The finite field of 2^n elements is denoted by $\mathbb{F}_{2^n}$, where $\mathbb{F}_{2^n} = \mathbb{F}_2[x]/g(x)$, $g(x)$ is an irreducible polynomial of degree n over the field $\mathbb{F}_2$. We denote by $\mathbb{Z}/2^n$ the ring of the integers modulo 2^n. There is a bijective mapping $\mathbb{Z}/2^n \to V_n$ that associates an element of the ring $\mathbb{Z}/2^n$ with its binary representation, and a bijective mapping $V_n \to \mathbb{F}_{2^n}$ that assigns a binary string to an element of the field $\mathbb{F}_{2^n}$. The operations of addition and multiplication in the field $\mathbb{F}_{2^n}$ are denoted by the signs "$+$" and "$\cdot$", respectively.

It is well known [15] that there are only three irreducible polynomials of degree 4 over the field $\mathbb{F}_2$. For definiteness, we will further work in the field $\mathbb{F}_{2^4} = \mathbb{F}_2[x]/x^4 + x + 1$.

Concatenation of the vectors $a \in V_n$, $b \in V_m$ is denoted by $a\|b \in V_{n+m}$. The *dot product* of two vectors $a, b \in V_n$ is an element of the field $\mathbb{F}_2$, calculated by the formula $\langle a, b\rangle = a_{n-1}b_{n-1} + \ldots + a_0b_0$ where addition and multiplication are carried out in the field $\mathbb{F}_2$. Note that the direct product of vector spaces $V_m \times V_m$ can be associated with V_{2m}.

Definition 1. *The vector Boolean (n, m)–function is a mapping $V_n \to V_m$. Permutation over V_n is a bijective (n, n)–function.*

The symmetric group of all permutations of the space V_n is denoted by $S(V_n)$.

Monomial permutations of the field $\mathbb{F}_{2^m}$ are permutations of the form x^d, where d is a positive integer such that $\gcd(d, 2^m - 1) = 1$. In this case, only the values $d < 2^m - 1$ can be considered. In particular, for $m = 4$, monomial permutations are obtained for $d \in \{1, 2, 4, 7, 8, 11, 13, 14\}$. Moreover, linear monomial permutations of the field $\mathbb{F}_{2^4}$ are x^d for $d \in \{1, 2, 4, 8\}$ [15].

Definition 2. *Let F be (n, m)-function, $1 \le t \le \min(n, m)$, $x_1, y_1 \in V_t$, $x_2 \in V_{n-t}$, $y_2 \in V_{m-t}$, $x = x_1\|x_2$, and $y = y_1\|y_2$. Let $T(x_1, x_2)$ be (n, t)-function such that when fixing an arbitrary x_2 the function T be a bijection*

with respect to the variable x_1, and U be $(n, m-t)$-function. Then if the function F is represented as:

$$F(x) = F(x_1 \| x_2) = (T(x_1, x_2), U(x_2, T(x_1, x_2))), \tag{1}$$

then such a representation of the function F will be called TU-decomposition [12].

Definition 3. *The differential uniformity of (n, m)-function F is defined as*

$$\delta_F = \max_{a \in V_n^{\times}, b \in V_m} \delta_F(a, b),$$

where $\delta_F(a, b) = |\{x \in V_n \,|\, F(x + a) + F(x) = b\}|$.

The use of functions with a lower differential uniformity in the synthesis of cryptographic algorithms makes it possible to guarantee resistance against the differential method of cryptographic analysis.

Definition 4. *The nonlinearity N_F of the (n, m)-function F is a value calculated by the formula*

$$N_F = 2^{n-1} - \frac{1}{2} \max_{a \in V_n, b \in V_m^{\times}} \left| \sum_{x \in V_n} (-1)^{\langle a, x \rangle + \langle b, F(x) \rangle} \right|.$$

The use of functions with greater nonlinearity in the synthesis of cryptographic algorithms makes it possible to guarantee resistance against the linear method of cryptographic analysis.

2 Generalized construction of $(2m, 2m)$-functions

Let $(2m, 2m)$-function $F(x_1, x_2) = y_1 \| y_2$, where $x_1, x_2, y_1, y_2 \in V_m$, be given by the following *generalized construction*, first introduced in [8],

$$\begin{aligned} y_1 = G_1(x_1, x_2) &= \begin{cases} x_1^{\alpha} \cdot x_2^{\beta}, & x_2 \neq 0, \\ \widehat{\pi}_1(x_1), & x_2 = 0, \end{cases} \\ y_2 = G_2(x_1, x_2) &= \begin{cases} x_1^{\gamma} \cdot x_2^{\delta}, & x_1 \neq 0, \\ \widehat{\pi}_2(x_2), & x_1 = 0. \end{cases} \end{aligned} \tag{2}$$

Hereinafter, one should go from the vectors of the space V_m to the corresponding elements of the field $\mathbb{F}_{2^m}$ and perform exponentiation and multiplication in the field $\mathbb{F}_{2^m}$. Moreover, in (2), $\widehat{\pi}_1$, $\widehat{\pi}_2$ are permutations over V_m. Without loss of generality, we assume that the following equalities hold

$$\widehat{\pi}_1(0) = 0, \quad \widehat{\pi}_2(0) = 0. \tag{3}$$

The parameters of the function (2) are the tuple of indexes $(\alpha, \beta, \gamma, \delta)$ of monomial permutations and permutations $\widehat{\pi}_1$, $\widehat{\pi}_2$.

For the system (2) to specify a bijective mapping under the condition (3), it is sufficient that the system

$$\begin{cases} G_1(x_1, x_2) &= b_1, \\ G_2(x_1, x_2) &= b_2, \end{cases}$$

has solutions for arbitrary $b_1, b_2 \in V_m$.

Statement 1. *The construction* (2) *admits TU-decomposition* (1).

Proof. Indeed, put $T(x_1, x_2) = G_1(x_1, x_2)$, note that for a fixed arbitrary x_2 the function T is a bijection with respect to the variable x_1, then

$$U(x_2, T(x_1, x_2)) = \begin{cases} (T(x_1, x_2))^{\lambda} \cdot x_2^{\mu}, & x_2 \neq 0,\ x_1 \neq 0, \\ \widehat{\pi}_2(x_2), & x_1 = 0, \\ 0, & x_2 = 0, \end{cases}$$

where $\alpha\lambda = \gamma \mod 2^m - 1$, $\mu = \delta - \beta\lambda \mod 2^m - 1$. □

Note that the GOST 34.12-2018 (Kuznyechik) permutation and the only known (up to CCZ-equivalence) 2-uniform permutation of the space V_n for even n also allow the TU-decomposition. The study of constructions that allow TU-decomposition seems to be important.

We can assume that $\delta_F > 8$ is a large value of the differential uniformity for the case $2m = 8$ since 8-bit permutation with $\delta_F = 8$ can be obtained by pseudo-random search [2, 16, 17].

3 On equivalence in the class of functions derived using a generalized construction

In this section, we propose the principle of partitioning the set of functions derived using a generalized construction into disjoint equivalence classes. The corresponding statement is proved. It is shown how to obtain the entire equivalence class from one of its representatives.

Let us present a lemma from the work [10, Lemma 1] for the case of functions that are obtained using the construction (2).

Lemma 1. *Let* $(2m, 2m)$*-function* F *be obtained using the construction* (2)*, and* $a_1, a_2, b_1, b_2 \in V_m$*, then* $\delta_F(a_1 \| a_2, b_1 \| b_2)$ *is greater than or equal to the*

number of solutions to the system of equations

$$\begin{cases} (x_1+a_1)^{\alpha}\cdot(x_2+a_2)^{\beta}+x_1^{\alpha}\cdot x_2^{\beta} &= b_1,\\ (x_1+a_1)^{\gamma}\cdot(x_2+a_2)^{\delta}+x_1^{\gamma}\cdot x_2^{\delta} &= b_2, \end{cases} \tag{4}$$

with the following constraints on the values of the variables x_1 *and* x_2

$$x_2\neq 0,\quad x_2\neq a_2,\quad x_1\neq 0,\quad x_1\neq a_1. \tag{5}$$

Proof of the lemma is obvious since under the constraints (5) the equations defining the function have the form (4). □

The following statement is a generalization of the corresponding statement from the work [10].

Statement 2. *The system* (4) *with a tuple of parameters* $(\alpha,\beta,\gamma,\delta)$*, where* x^{α}*,* x^{β}*,* x^{γ}*, and* x^{δ} *define monomial permutations, has the maximum number of solutions* (x_1,x_2)*,* $x_1,x_2\in\mathbb{F}_{2^m}$*, satisfying the conditions* (5)*, for* $a_1,a_2,b_1,b_2\in\mathbb{F}_{2^m}$ *(*a_1 *and* a_2 *do not vanish simultaneously), that is coincide with the maximum number of solutions of systems of the form* (4) *under constraints* (5) *with the following tuples of parameters*

$$\begin{gathered} (\alpha\cdot d_1,\ \beta\cdot d_1,\ \gamma\cdot d_2,\ \delta\cdot d_2) \mod 2^m-1,\\ (\alpha\cdot d_1,\ \beta\cdot d_2,\ \gamma\cdot d_1,\ \delta\cdot d_2) \mod 2^m-1,\\ (\gamma,\ \delta,\ \alpha,\ \beta),\qquad (\beta,\ \alpha,\ \delta,\ \gamma),\qquad (\delta,\ \gamma,\ \beta,\ \alpha), \end{gathered}$$

where x^{d_1}*,* x^{d_2} *define linear permutations.*

Proof. Let us consider the following system

$$\begin{cases} (x_1+a_1)^{d_1\cdot\alpha}\cdot(x_2+a_2)^{d_1\cdot\beta}+x_1^{d_1\cdot\alpha}\cdot x_2^{d_1\cdot\beta} &= b_1^{d_1},\\ (x_1+a_1)^{d_2\cdot\gamma}\cdot(x_2+a_2)^{d_2\cdot\delta}+x_1^{d_2\cdot\gamma}\cdot x_2^{d_2\cdot\delta} &= b_2^{d_2}, \end{cases} \tag{6}$$

where $a_1,a_2,b_1,b_2\in\mathbb{F}_{2^m}$ and a_1, a_2 do not vanish simultaneously. Note that, because of x^{d_1} and x^{d_2} are the bijective mappings, if b_1 and b_2 take all values from the field $\mathbb{F}_{2^m}$, then $b_1^{d_1}$, $b_2^{d_2}$ also take all values from this field. Taking into account the linearity of the mappings x^{d_1} and x^{d_2}, we write the system (6) in the form

$$\begin{cases} ((x_1+a_1)^{\alpha}\cdot(x_2+a_2)^{\beta}+x_1^{\alpha}\cdot x_2^{\beta})^{d_1} &= b_1^{d_1},\\ ((x_1+a_1)^{\gamma}\cdot(x_2+a_2)^{\delta}+x_1^{\gamma}\cdot x_2^{\delta})^{d_2} &= b_2^{d_2}. \end{cases}$$

Again, due to the bijectivity of the functions x^{d_1} and x^{d_2}, this system is equivalent to the system (4). Thus, a system with a tuple of parameters

$(\alpha \cdot d_1,\ \beta \cdot d_1,\ \gamma \cdot d_2,\ \delta \cdot d_2) \mod 2^m - 1$ has the maximum number of solutions that satisfy the conditions (5), which coincides with the maximum number of solutions of the system (4).

Further, consider the following system

$$\begin{cases} (x_1 + a_1)^{d_1 \cdot \alpha} \cdot (x_2 + a_2)^{d_2 \cdot \beta} + x_1^{d_1 \cdot \alpha} \cdot x_2^{d_2 \cdot \beta} &= b_1, \\ (x_1 + a_1)^{d_1 \cdot \gamma} \cdot (x_2 + a_2)^{d_2 \cdot \delta} + x_1^{d_1 \cdot \gamma} \cdot x_2^{d_2 \cdot \delta} &= b_2, \end{cases} \tag{7}$$

where $a_1, a_2, b_1, b_2 \in \mathbb{F}_{2^m}$ and a_1, a_2 do not vanish simultaneously. Taking into account the linearity of the mappings x^{d_1} and x^{d_2}, we write the system (7) in the form

$$\begin{cases} (x_1^{d_1} + a_1^{d_1})^{\alpha} \cdot (x_2^{d_2} + a_2^{d_2})^{\beta} + (x_1^{d_1})^{\alpha} \cdot (x_2^{d_2})^{\beta} &= b_1, \\ (x_1^{d_1} + a_1^{d_1})^{\gamma} \cdot (x_2^{d_2} + a_2^{d_2})^{\delta} + (x_1^{d_1})^{\gamma} \cdot (x_2^{d_2})^{\delta} &= b_2. \end{cases}$$

Making the replacement $x_1^{d_1} = y_1$, $x_2^{d_2} = y_2$, $a_1^{d_1} = \overline{a}_1$, and $a_2^{d_2} = \overline{a}_2$, we get a system of the form (4)

$$\begin{cases} (y_1 + \overline{a}_1)^{\alpha} \cdot (y_2 + \overline{a}_2)^{\beta} + y_1^{\alpha} \cdot y_2^{\beta} &= b_1, \\ (y_1 + \overline{a}_1)^{\gamma} \cdot (y_2 + \overline{a}_2)^{\delta} + y_1^{\gamma} \cdot y_2^{\delta} &= b_2. \end{cases}$$

That is why, a system with a tuple of parameters $(\alpha \cdot d_1,\ \beta \cdot d_2,\ \gamma \cdot d_1,\ \delta \cdot d_2) \mod 2^m - 1$ has the maximum number of solutions that satisfy the conditions (5), which coincides with the maximum number of solutions of the system (4).

The systems of the form (4) with tuples of parameters $(\alpha, \beta, \gamma, \delta)$, $(\gamma, \delta, \alpha, \beta)$, $(\beta, \alpha, \delta, \gamma)$, and $(\delta, \gamma, \beta, \alpha)$ coincide up to a change in the order of writing equations or renaming variables. □

Further, throughout this section, we will consider the case $m = 4$.

Remark 1. *Note that the sets* $\{1, 2, 4, 8\}$ *and* $\{7, 11, 13, 14\}$ *are closed under multiplication by* $d \in \{1, 2, 4, 8\}$ *modulo 15. Then, by virtue of Statement 2, we obtain that* $8^4 = 2^{12} = 4096$ *of all possible parameter tuples* $(\alpha, \beta, \gamma, \delta)$ *of the functions from the family* (2) *are split into disjoint equivalence classes with the same maximum number of solutions of the system* (4) *under the constraints* (5) *in each class. A distinct equivalence class can be obtained from one of its representatives* $(\alpha, \beta, \gamma, \delta)$*, by composing different tuples from the following ones*

$$(\alpha \cdot d_1 \cdot d_3,\quad \beta \cdot d_1 \cdot d_4,\quad \gamma \cdot d_2 \cdot d_3,\quad \delta \cdot d_2 \cdot d_4) \mod 2^m - 1, \tag{8a}$$
$$(\gamma \cdot d_1 \cdot d_3,\quad \delta \cdot d_1 \cdot d_4,\quad \alpha \cdot d_2 \cdot d_3,\quad \beta \cdot d_2 \cdot d_4) \mod 2^m - 1, \tag{8b}$$
$$(\beta \cdot d_1 \cdot d_3,\quad \alpha \cdot d_1 \cdot d_4,\quad \delta \cdot d_2 \cdot d_3,\quad \gamma \cdot d_2 \cdot d_4) \mod 2^m - 1, \tag{8c}$$
$$(\delta \cdot d_1 \cdot d_3,\quad \gamma \cdot d_1 \cdot d_4,\quad \beta \cdot d_2 \cdot d_3,\quad \alpha \cdot d_2 \cdot d_4) \mod 2^m - 1, \tag{8d}$$

where $m = 4, \quad d_1, d_2, d_3, d_4 \in \{1, 2, 4, 8\}$.

Statement 3. *There are* 64 *different tuples of the form*

$$(d_1 \cdot d_3, \quad d_1 \cdot d_4, \quad d_2 \cdot d_3, \quad d_2 \cdot d_4) \mod 2^m - 1, \tag{9}$$

where $m = 4$, $d_1, d_2, d_3, d_4 \in \{1, 2, 4, 8\}$.

Proof. At the beginning, let us put $d_1 \cdot d_3 = 1$, this is possible in four cases: $d_1 = d_3 = 1$, or $d_1 = d_3 = 4$, or $d_1 = 2$, $d_3 = 8$, or $d_1 = 8$, $d_3 = 2$. Note that in the first case tuples of the form $(1,\ d_4,\ d_2,\ d_2 \cdot d_4) \mod 2^m - 1$ are specified. Taking into account that $d_2, d_4 \in \{1, 2, 4, 8\}$, we get 16 different tuples. In the remaining three cases, the values d_1 and d_3 generate tuples that coincide with these 16 already considered ones. Similarly, we obtain 16 different tuples for the cases $d_1 \cdot d_3 = 2$, $d_1 \cdot d_3 = 4$, $d_1 \cdot d_3 = 8$. This implies the validity of the statement. □

Thus, by virtue of Statement 2, the set of $(8, 8)$-functions derived using the generalized construction (2) is divided into equivalence classes in conformity with the tuples of parameters $(\alpha, \beta, \gamma, \delta)$ with the same maximum number of solutions to (4), (5) for functions from the same class. Moreover, due to Lemma 1, functions from the same class have the same lower bound for differential uniformity. The auxiliary Statement 3 that is proven in this section we will use to calculate the cardinality of each equivalence class.

4 Justification of criteria for rejection of vector Boolean functions derived using a generalized construction

In this section, we have proved statements that allow us to reject functions given by the construction (2), either by the high differential uniformity or by the fact that they are not permutations. Moreover, by the virtue of the statements substantiated in the previous and present sections, the conclusion about all functions from the equivalence class is based on the analysis of only one of its representatives.

4.1 On differential uniformity

This paragraph is devoted to rejection of $(2m, 2m)$-functions (2), in the case of $m = 4$, by differential uniformity $2^m - 2 = 14$ and higher. Moreover, some of the statements below are also true in the general view (without restriction $m = 4$), which will be noted in their wording.

Proposition 1. *Let F be a $(2m, 2m)$-function given by the construction* (2). *If x^α, x^γ define linear permutations, then $\delta_F \geq 2^m - 2$.*

Proof. We seek $x_1 \neq 0$, $x_2 \neq 0$. Consider the case $a_2 = 0$, then the system of equations (4) can be written in the form

$$\begin{cases} x_2^\beta((x_1 + a_1)^\alpha + x_1^\alpha) &= b_1, \\ x_2^\delta((x_1 + a_1)^\gamma + x_1^\gamma) &= b_2. \end{cases}$$

Since the permutations x^α and x^γ are linear, we obtain

$$\begin{cases} x_2^\beta(x_1^\alpha + a_1^\alpha + x_1^\alpha) &= b_1, \\ x_2^\delta(x_1^\gamma + a_1^\gamma + x_1^\gamma) &= b_2, \end{cases}$$

$$\begin{cases} x_2^\beta \cdot a_1^\alpha &= b_1, \\ x_2^\delta \cdot a_1^\gamma &= b_2. \end{cases} \tag{10}$$

Further, we fix arbitrarily $a_1, b_1 \in \mathbb{F}_{2^m}$, $a_1 \neq 0$, $b_1 \neq 0$. Because of the bijectivity of the mapping x^β from the first equation of the system (10) we find unique $x_2 \neq 0$ and substitute it into the second equation, thereby defining b_2. Thus, for fixed permissible values of a_1, b_1, b_2, the system (10) is compatible with respect to x_2, while x_1 can take any admissible values. Therefore, taking into account the constraints $x_1 \neq 0$, $x_1 \neq a_1$, we obtain that the number of solutions of the system is at least $2^m - 2$. Using Lemma 1, we get that $\delta_F \geq 2^m - 2$. □

Remark 2. *By virtue of Proposition 1 and Statement 2 in the case $m = 4$ we have $2 \cdot 4^2 \cdot 8^2 - 4^4 = 1792$ tuples of parameters $(\alpha, \beta, \gamma, \delta)$ corresponding to $(8, 8)$-functions from the family* (2) *with a large differential uniformity.*

Proposition 2. *Let F be a $(2m, 2m)$-function given by the construction* (2). *If $\alpha = \beta = \gamma = \delta$, then $\delta_F \geq 2^m - 2$.*

Proof. We seek $x_1 \neq 0$, $x_2 \neq 0$. Let $a_2 = 0$, $b_1 = b_2 = 1$, then the system of equations (4) is reduced to one equation

$$x_2^\alpha((x_1 + a_1)^\alpha + x_1^\alpha) = 1. \tag{11}$$

Since the mapping x^α is bijective, if x_2 takes all $2^m - 1$ values from the multiplicative group of the field $\mathbb{F}_{2^m}$, then the inverse element to x_2^α, which we denote c, also takes all values from the multiplicative group. Thus, the equation (11) can be written as

$$(x_1 + a_1)^\alpha + x_1^\alpha = c, \tag{12}$$

where $c \in \mathbb{F}_{2^m} \setminus \{0\}$. It is known [18, Sec. 4] that the total number of solutions to the equation (12) is equal to 2^m, where $a_1 \neq 0$ is a fixed value and c takes all $2^m - 1$ values from the multiplicative group of the field $\mathbb{F}_{2^m}$. Therefore, taking into account the constraints $x_1 \neq 0$, $x_1 \neq a_1$ (5), the number of solutions of the original system is no less than $2^m - 2$. By virtue of the Lemma 1 the same estimation is true for δ_F. □

Remark 3. *By virtue of Proposition 2 and Statements 2,3 in the case* $m = 4$ *we have* 64 *tuples of parameters* $(\alpha, \beta, \gamma, \delta)$ *that were not previously considered in Proposition 1. These tuples define* $(8, 8)$*-functions from the family* (2) *with a large differential uniformity. The representative of this equivalence class is* $(7, 7, 7, 7)$.

Proposition 3. *Let* F *be a* $(8, 8)$*-function given by the construction* (2)*. If* $\alpha = 11$, $\beta = \gamma = 1$, $\delta = 13$, *then* $\delta_F \geq 14$.

Proof. Let $a_1 = a_2 = x \in \mathbb{F}_{2^4}$, $b_1 = 0$, $b_2 = x^3 + 1 \in \mathbb{F}_{2^4}$, then the system of equations (4) can be written in the form

$$\begin{cases} (x_1 + x)^{11} \cdot (x_2 + x) + x_1^{11} \cdot x_2 &= 0, \\ (x_1 + x) \cdot (x_2 + x)^{13} + x_1 \cdot x_2^{13} &= x^3 + 1. \end{cases} \tag{13}$$

From the first equation in (13) it follows that $x_1 \neq 0$, $x_2 \neq 0$, $x_1 \neq x = a_1$, $x_2 \neq x = a_2$, in addition, $(x_1 + x)^{11} \cdot (x_2 + x) = x_1^{11} \cdot x_2$, therefore

$$(x_1 + x) \cdot (x_2 + x)^{11} = x_1 \cdot x_2^{11}. \tag{14}$$

Substituting the expression for the left-hand side from (14) into the second equation of the system (13), we obtain the chain of equations

$$x_1 \cdot x_2^{11} \cdot (x_2 + x)^2 + x_1 \cdot x_2^{13} = x^3 + 1 \Leftrightarrow x_1 \cdot x_2^{11} \cdot (x_2^2 + x^2 + x_2^2) = x^3 + 1 \Leftrightarrow$$
$$\Leftrightarrow x_1 \cdot x_2^{11} \cdot x^2 = x^3 + 1 \Leftrightarrow x_1 \cdot x_2^{11} = (x^3 + 1) \cdot x^{13} \Leftrightarrow x_1 \cdot x_2^{11} = x^{12}.$$

Hence, taking into account the conditions $x_1 \neq x$, $x_2 \neq x$, we get 14 solutions (x_1, x_2). By Lemma 1, we obtain that $\delta_F \geq 14$. □

Remark 4. *For the representative* $(\alpha, \beta, \gamma, \delta) = (11, 1, 1, 13)$ *all different tuples of its equivalence class can be obtained by the formulas* (8*a*) *and* (8*b*)*, since the formulas* (8*c*) *and* (8*d*) *will give the same tuples. Indeed, the tuple of parameters* $(13, 1, 1, 11)$ *in* (8*d*) *is obtained from* $(11, 1, 1, 13)$ *in* (8*a*) *for* $d_1 = 2$, $d_2 = d_3 = 4$, $d_4 = 8$, *the tuple* $(1, 11, 13, 1)$ *in* (8*c*) *is produced from* $(1, 13, 11, 1)$ *in* (8*b*) *for* $d_1 = 2$, $d_2 = d_4 = 1$, $d_3 = 8$. *Hence, by virtue of Proposition 3 and Statements 2,3 we get 128 tuples of parameters* $(\alpha, \beta, \gamma, \delta)$ *corresponding to* $(8, 8)$*-functions from the family* (2) *with a large differential uniformity.*

Proposition 4. *Let F be a $(8,8)$-function given by the construction (2). If $\alpha = 7$, $\beta = \gamma = 1$, $\delta = 7$, then $\delta_F \geq 14$.*

Proof. Let $a_1 = a_2 = a \in \mathbb{F}_{2^4}$, $a \neq 0$, $b_1 = b_2 = b \in \mathbb{F}_{2^4}$, $b \neq 0$, $x_1 = x_2 \neq 0$, then the system of equations (4) can be written in the form

$$\begin{cases} (x_1+a)^7 \cdot (x_1+a) + x_1^7 \cdot x_1 &= b, \\ (x_1+a) \cdot (x_1+a)^7 + x_1 \cdot x_1^7 &= b, \end{cases} \tag{15}$$

The system (15) is reduced to equation $(x_1+a)^8 + x_1^8 = b$, or

$$a^8 = b. \tag{16}$$

Let us choose $a, b \in \mathbb{F}_{2^4}$ satisfying the equality (16). Then, taking into account the conditions (5) we get 14 solutions (x_1, x_1). By Lemma 1, we obtain that $\delta_F \geq 14$. □

Remark 5. *For the representative $(\alpha,\beta,\gamma,\delta) = (7,1,1,7)$ all different tuples of its equivalence class can be obtained by the formulas (8a) and (8b), since the tuples $(\beta,\alpha,\delta,\gamma)$ and $(\delta,\gamma,\beta,\alpha)$ in the formulas (8c) and (8d) are identical to tuples $(\gamma,\delta,\alpha,\beta)$ and $(\alpha,\beta,\gamma,\delta)$ in (8b) and (8a) respectively. Hence, by virtue of Proposition 4 and Statements 2,3 we have 128 tuples of parameters $(\alpha,\beta,\gamma,\delta)$ corresponding to $(8,8)$-functions from the family (2) with a large differential uniformity.*

4.2 On the functions that are not permutations

This section is devoted to rejecting such $(2m,2m)$-functions (2), which can not be used to construct a permutation. The possibility of rejecting the entire equivalence class by one of its representatives, which is not a bijection for any values of auxiliary permutations $\widehat{\pi}_1$, $\widehat{\pi}_2$, is justified. Further, in the case $m = 4$, a proposition to discard representatives of seven equivalence classes by the indicated condition is proved.

Statement 4. *Let F be a $(2m,2m)$-function given by the construction (2) with a tuple of parameters $(\alpha,\beta,\gamma,\delta)$, where x^α, x^β, x^γ, and x^δ define monomial permutations. If F is not a bijection for any values of the permutations $\widehat{\pi}_i(x_i)$, $i \in \{0,1\}$, then any $(2m,2m)$-function from the family (2) with the following tuples of parameters*

$$\begin{gathered} (\alpha \cdot d_1,\ \beta \cdot d_1,\ \gamma \cdot d_2,\ \delta \cdot d_2) \mod 2^m - 1, \\ (\alpha \cdot d_1,\ \beta \cdot d_2,\ \gamma \cdot d_1,\ \delta \cdot d_2) \mod 2^m - 1, \\ (\gamma,\ \delta,\ \alpha,\ \beta), \qquad (\beta,\ \alpha,\ \delta,\ \gamma), \qquad (\delta,\ \gamma,\ \beta,\ \alpha), \end{gathered}$$

where x^{d_1}, x^{d_2} define a linear permutation, is also not a bijection for any values of the permutations $\widehat{\pi}_i(x_i)$, $i \in \{0,1\}$.

Proof. By the wording of the statement, the $(2m,2m)$-function F from the family (2) with parameters $(\alpha,\beta,\gamma,\delta)$ is not a bijection for any values of the permutations $\widehat{\pi}_i(x_i)$, $i \in \{0,1\}$, that is, there are such values $x_1, x_2, \widetilde{x}_1, \widetilde{x}_2 \in \mathbb{F}_{2^m}$, $x_1, x_2, \widetilde{x}_1, \widetilde{x}_2 \neq 0$, moreover, $x_1 \neq \widetilde{x}_1$ or $x_2 \neq \widetilde{x}_2$, that the following equalities hold

$$y_1 = G_1(x_1,x_2) = x_1^{\alpha} \cdot x_2^{\beta} = \widetilde{x}_1^{\alpha} \cdot \widetilde{x}_2^{\beta} = G_1(\widetilde{x}_1,\widetilde{x}_2) = \widetilde{y}_1,$$
$$y_2 = G_2(x_1,x_2) = x_1^{\gamma} \cdot x_2^{\delta} = \widetilde{x}_1^{\gamma} \cdot \widetilde{x}_2^{\delta} = G_2(\widetilde{x}_1,\widetilde{x}_2) = \widetilde{y}_2.$$

Then for the same values of $x_1, x_2, \widetilde{x}_1, \widetilde{x}_2$ for the tuples $(\alpha \cdot d_1,\ \beta \cdot d_1,\ \gamma \cdot d_2,\ \delta \cdot d_2) \mod 2^m - 1$ we have $G_1(x_1,x_2) = x_1^{d_1\alpha} \cdot x_2^{d_1\beta} = (x_1^{\alpha} \cdot x_2^{\beta})^{d_1} = (\widetilde{x}_1^{\alpha} \cdot \widetilde{x}_2^{\beta})^{d_1} = \widetilde{x}_1^{d_1\alpha} \cdot \widetilde{x}_2^{d_1\beta} = G_1(\widetilde{x}_1,\widetilde{x}_2)$, and similarly, $G_2(x_1,x_2) = G_2(\widetilde{x}_1,\widetilde{x}_2)$.

Based on the bijectivity of the mappings x^{d_1}, x^{d_2}, which define linear permutations, for the tuples $(\alpha \cdot d_1,\ \beta \cdot d_2,\ \gamma \cdot d_1,\ \delta \cdot d_2) \mod 2^m - 1$ we uniquely find values $v_1, v_2, \widetilde{v}_1, \widetilde{v}_2 \in \mathbb{F}_{2^m}$, $v_1, v_2, \widetilde{v}_1, \widetilde{v}_2 \neq 0$, such that $v_1^{d_1} = x_1$, $v_2^{d_2} = x_2$, $\widetilde{v}_1^{d_1} = \widetilde{x}_1$, $\widetilde{v}_2^{d_2} = \widetilde{x}_2$. Then $G_1(v_1,v_2) = v_1^{d_1\alpha} \cdot v_2^{d_2\beta} = x_1^{\alpha} \cdot x_2^{\beta} = \widetilde{x}_1^{\alpha} \cdot \widetilde{x}_2^{\beta} = \widetilde{v}_1^{d_1\alpha} \cdot \widetilde{v}_2^{d_2\beta} = G_1(\widetilde{v}_1,\widetilde{v}_2)$, and similarly, $G_2(v_1,v_2) = G_2(\widetilde{v}_1,\widetilde{v}_2)$.

For tuples of parameters $(\gamma,\delta,\alpha,\beta)$, $(\beta,\alpha,\delta,\gamma)$, $(\delta,\gamma,\beta,\alpha)$ the equal values $y_1 = \widetilde{y}_1$ and $y_2 = \widetilde{y}_2$ are obtained by the corresponding transposition of arguments $x_1, x_2, \widetilde{x}_1, \widetilde{x}_2$. □

Proposition 5. *$(8,8)$-function F given by the construction (2) with the parameters $(\alpha,\beta,\gamma,\delta)$ from the list below 1) $(7,7,7,13)$, 2) $(1,7,7,7)$, 3) $(4,7,7,7)$, 4) $(7,7,2,2)$, 5) $(1,1,7,13)$, 6) $(2,7,7,7)$, 7) $(7,2,2,7)$, is not a bijection for any values of the permutations $\widehat{\pi}_i(x_i)$, $i \in \{0,1\}$.*

Proof. For the construction (2) with each of the seven specified tuples of parameters from the wording of the proposition, it suffices to indicate the values $x_1, x_2, \widetilde{x}_1, \widetilde{x}_2 \in \mathbb{F}_{2^4}$, $x_1, x_2, \widetilde{x}_1, \widetilde{x}_2 \neq 0$, moreover, $x_1 \neq \widetilde{x}_1$ or $x_2 \neq \widetilde{x}_2$, such that $y_1 = G_1(x_1,x_2) = x_1^{\alpha} \cdot x_2^{\beta} = \widetilde{x}_1^{\alpha} \cdot \widetilde{x}_2^{\beta} = G_1(\widetilde{x}_1,\widetilde{x}_2) = \widetilde{y}_1$ and $y_2 = G_2(x_1,x_2) = x_1^{\gamma} \cdot x_2^{\delta} = \widetilde{x}_1^{\gamma} \cdot \widetilde{x}_2^{\delta} = G_2(\widetilde{x}_1,\widetilde{x}_2) = \widetilde{y}_2$.

1. Let $x_1 = x \in \mathbb{F}_{2^4}$, $x_2 = x^3 + x^2 = x^6 \in \mathbb{F}_{2^4}$, $\widetilde{x}_1 = \widetilde{x}_2 = x^3 + x^2 + x = x^{11} \in \mathbb{F}_{2^4}$. Then $y_1 = x_1^7 \cdot x_2^7 = x^7 \cdot (x^6)^7 = x^4$, $y_2 = x_1^7 \cdot x_2^{13} = x^7 \cdot (x^6)^{13} = x^{10}$, $\widetilde{y}_1 = \widetilde{x}_1^7 \cdot \widetilde{x}_2^7 = (x^{11})^7 \cdot (x^{11})^7 = x^4$, $\widetilde{y}_2 = \widetilde{x}_1^7 \cdot \widetilde{x}_2^{13} = (x^{11})^7 \cdot (x^{11})^{13} = x^{10}$.

For the other tuples, the proof can be carried out similarly; therefore, we present only appropriate values $x_1, x_2, \widetilde{x}_1, \widetilde{x}_2$.

2. $x_1 = 1$, $x_2 = x^2 + x$, $\widetilde{x}_1 = x^3 + x^2 + x$, $\widetilde{x}_2 = x^2 + x + 1$;

3. $x_1 = x^3 + x^2 + x$, $x_2 = x$, $\widetilde{x}_1 = x$, $\widetilde{x}_2 = x^3 + x^2 + x$;

4. $x_1 = x_2 = 1$, $\widetilde{x}_1 = x^3 + x^2 + x + 1$, $\widetilde{x}_2 = x^3$;
5. $x_1 = x$, $x_2 = x^3$, $\widetilde{x}_1 = x^3 + x^2 + x$, $\widetilde{x}_2 = x^2 + 1$;
6. $x_1 = 1$, $x_2 = x^3 + x$, $\widetilde{x}_1 = \widetilde{x}_2 = x^3 + x^2 + x + 1$;
7. $x_1 = 1$, $x_2 = x^3 + x^2$, $\widetilde{x}_1 = x^3 + x^2 + x + 1$, $\widetilde{x}_2 = x^3 + x$. □

Corollary 1. $(8, 8)$*-functions* F *from the family* (2) *with parameters* $(\alpha, \beta, \gamma, \delta)$ *from the equivalence classes generated by the tuples of parameters indicated in the Proposition* 5*, are not bijections for any values of the permutations* $\widehat{\pi}_i(x_i)$*,* $i \in \{0, 1\}$*.*

Taking into account the Corollary 1, we reject all tuples of parameters from the equivalence classes with representatives specified in the Proposition 5.

Remark 6. *For the representative* $(\alpha, \beta, \gamma, \delta) = (7, 7, 7, 13)$ *all different tuples of its equivalence class can be obtained by the formula* (8*a*)*, since the formulas* (8*b*)*,* (8*c*)*, and* (8*d*) *will give the same tuples. Therefore, in the equivalence class generated by the representative* $(7, 7, 7, 13)$*, there are 64 tuples of parameters. Further, the representatives of* $(4, 7, 7, 7)$*,* $(1, 7, 7, 7)$ *and* $(2, 7, 7, 7)$ *generate three classes with 256 tuples in each one (768 tuples in total). Reasoning similarly to the Remark 5, we can show that the representatives of* $(1, 1, 7, 13)$*,* $(7, 7, 2, 2)$ *and* $(7, 2, 2, 7)$ *generate equivalence classes of 128 tuples in each one (384 tuples in total).*

In Table 1 we show the representatives of the equivalence classes and the reasons for rejection.

№	The representative of the equivalence class	The number of elements	The reason for rejection
1	Generalized representative: $(\alpha, \beta, \gamma, \delta)$, where $\alpha, \gamma \in \{1, 2, 4, 8\}$	1792	$\delta_F \geq 14$, according to Statement 1
2	(7,7,7,7)	64	$\delta_F \geq 14$, according to Statement 2
3	(11,1,1,13)	128	$\delta_F \geq 14$, according to Statement 3
4	(7,1,1,7)	128	$\delta_F \geq 14$, according to Statement 4
5	(7,7,7,13)	64	are not permutations, according to Statement 5
6	(1,7,7,7)	256	
7	(4,7,7,7)	256	
8	(7,7,2,2)	128	
9	(1,1,7,13)	128	
10	(2,7,7,7)	256	
11	(7,2,2,7)	128	
12	(1,1,7,11)	256	are not rejected
13	(1,7,7,11)	256	
14	(1,7,7,2)	128	
15	(7,7,7,11)	128	

Table 1: Summary table of the equivalence classes for $m = 4$.

Conclusion

The statements proved in this paper justify the rejection of 3328 tuples of parameters $(\alpha, \beta, \gamma, \delta)$ of $(8, 8)$-functions F defined by the construction (2) due to the value $\delta_F \geq 14$ or because F is not a bijection. The 768 tuples of parameters $(\alpha, \beta, \gamma, \delta)$ remained unrejected, which are split by Statement 2 and Remark 1 into 4 equivalence classes with representatives $(1, 1, 7, 11)$, $(1, 7, 7, 11)$ with 256 tuples in each class, $(1, 7, 7, 2)$, $(7, 7, 7, 11)$ with 128 tuples in each class (see table 1). In works [8, 10] it is indicated that using these tuples of parameters with the correct choice of permutations $\widehat{\pi}_i(x_i)$, $i \in \{0, 1\}$ 6-uniform permutations with nonlinearity 108 can be obtained.

References

[1] Shannon C.E., "Communication theory of secrecy systems", *Bell Syst. Techn. J.*, **28** (1949), 656–715.

[2] Menyachikhin A.V., "Spectral-linear and spectral-differential methods for generating S-boxes having almost optimal cryptographic parameters", *Mat. Vopr. Kriptogr.*, **8**:2 (2017), 97–116, https://doi.org/10.4213/mvk227.

[3] Fomin D., "On the way of constructing 2n-bit permutations from n-bit ones", The VIIIth Workshop on Current Trends in Cryptology (CTCrypt 2019), 2019, https://ctcrypt.ru/files/files/2019/materials/07_Fomin.pdf.

[4] De la Cruz Jiménez R.A., "Generation of 8-bit S-Boxes Having Almost Optimal Cryptographic Properties Using Smaller 4-bit S-Boxes and Finite Field Multiplication", *LNCS*, Progress in Cryptology — LATINCRYPT 2017, **11368**, eds. T. Lange and O. Dunkelman, Springer Nature, Switzerland AG, 2019, 191–206, https://doi.org/10.1007/978-3-030-25283-0_11.

[5] De la Cruz Jiménez R.A., *On some methods for constructing almost optimal S-Boxes and their resilience against side-channel attacks Cryptology ePrint Archive, Report 2018/618*, https://eprint.iacr.org/2018/618.

[6] De la Cruz Jiménez R.A., "A method for constructing permutations, involutions and orthomorphisms with strong cryptographic properties", *PDM. Prilozheniye*, **12** (2019), 145–151, https://doi.org/10.17223/2226308X/12/42.

[7] Fomin D.B., "New classes of 8-bit permutations based on a butterfly structure", *Mat. Vopr. Kriptogr.*, **10**:2 (2019), 169–180, https://doi.org/10.4213/mvk294.

[8] Fomin D.B., "On approaches to constructing low-resource nonlinear transformations", *Obozreniye prikladnoy i promyshlennoy matematiki*, **25**:4 (2018), 379–381, In Russian.

[9] Fomin D.B., "Constructing permutations of the space V_{2m} using $(2m, m)$-functions", *Mat. Vopr. Kriptogr.*, **11**:3 (2020), 121-138, In Russian.

[10] Fomin D.B., "On algebraic degree and differential uniformity of permutations of the space V_{2m}, constructed using $(2m, m)$ - functions", *Mat. Vopr. Kriptogr.*, **11**:4 (2020), 133-149, In Russian.

[11] Biryukov A., Perrin L., Udovenko A., "Reverse-engineering the s-box of streebog, kuznyechik and stribobr1", *Lecture Notes in Computer Science*, EUROCRYPT (1), **9665**, eds. Marc Fischlin and Jean-Sébastien Coron, Springer, 2016, 372–402.

[12] Canteaut A., Perrin L., *Cryptology ePrint Archive, Report 2018/713*, https://eprint.iacr.org/2018/713.

[13] Kostrikin A.I., *Introduction to Algebra*, Springer-Verlag, New York, 1982, 577 pp.

[14] Kazymyrov O.V., Kazymyrova V.N., Oliynykov R.V., "A method for generation of high-nonlinear S-boxes based on gradient descent", *Mat. Vopr. Kriptogr.*, **5**:2 (2014), 71–78, https://doi.org/10.4213/mvk118.

[15] Lidl R., Niederreiter H., *Finite Fields*, 2nd, Cambridge University Press, 1997, 755 pp.

[16] Knuth D., *Art of Computer Programming*. V. 2: *Seminumerical Algorithms*, 3rd, Addison-Wesley Professional, 1997.

[17] Kazymyrov O.V., *Methods and tools for generating nonlinear replacement nodes for symmetric cryptographic algorithms*, Dissertatsiya kand. tekhn. nauk. Khar'kov, 2013, 190 pp., In Russian.

[18] Heys H., *A Tutorial on Linear and Differential Cryptanalysis, 2002*, http://www.engr.mun.ca/~howard/PAPERS/ldc_tutorial.pdf.

On the Generation of Cryptographically Strong Substitution Boxes from Small Ones and Heuristic Search

Alejandro Freyre-Echevarría

Institute of Cryptography, Cuba
alefreyre.43@gmail.com

Abstract

The design of cryptographically strong S-Boxes is a wide studied field in symmetric cryptography. Several techniques to produce resilient substitutions have been developed through the years having algebraic constructions the most interesting results. However the most representative solution of these constructions, the finite field inversion, does not warranty total security against algebraic attacks, since the graph algebraic immunity of such permutation is not optimal. In this paper is proposed a combination of algebraic construction and heuristic method to produce a large set of different 8-bit substitution boxes with optimal graph algebraic immunity, maximum value of minimum algebraic degree and almost optimal values of nonlinearity and differential uniformity for application to symmetric cryptographic schemes.

Keywords: Substitution boxes, cryptographic properties, heuristic method.

1 Introduction

The design of robust symmetric encryption algorithms is dependent of its components, where substitution boxes often play the role of nonlinear component of these encryption schemes. Hence, they are the target for numerous attack which threaten the security and integrity of the data. Among the most prominent attacks to S-Boxes figure linear, differential and algebraic cryptanalysis as well as side-channel attacks [1, 2, 3, 4, 5, 6, 7, 8]. In consequence, cryptographic properties of S-Boxes attempt to measure the resilience against such attacks. Section 2.2 is dedicated to explain briefly the properties of balance, nonlinearity, differential uniformity, algebraic degree and algebraic immunity. However, these are a fraction of all properties related to S-Boxes, which were not taken into account because they are not in the scope of the present paper. Nonetheless, for the interested lecturer, here is left a compilation of some other studied properties of S-Boxes: absolute indicator and sum of squared indicator [9], strict avalanche criteria [10], bit

independence criteria [11], SNR-DPA [12], transparency order [13], modified transparency order [14], revisited tranparency order [15], confusion coefficient variance [16], non-absolute indicator [17].

In literature survey, there exist three widely known methods to produce substitution boxes. The first, pseudo-random generation, which include S-Boxes generated through chaotic sequences [18, 19, 20, 21], produce permutations with nonlinearity values up to 100 and differential uniformity equal or greater than 8. The second method, monomial power functions over the finite field $\mathbb{F}_{2^8}$, e.g. the finite field inversion used to construct the AES substitution box [22], present the best known values of nonlinearity and differential uniformity for any bijective substitution box (see Table 1). In addition, the simple interpolation polynomial of substitution boxes obtained by such constructions allows efficient masking as a countermeasure to side-channel attacks [23]. However, the monomial permutations on $\mathbb{F}_{2^8}$ present a major flaw, the graph algebraic immunity of such S-Boxes is not optimal, i.e. a weakness towards algebraic cryptanalysis. Finally, heuristic methods try to obtain similar values of nonlinearity and differential uniformity as achieved by monomial power functions, maintaining a random structure in the representation of the S-Box, and often present optimal graph algebraic immunity. In addition, substitution boxes generated by heuristic methods present a complex interpolation polynomial, therefore they are difficult to mask. Although the aforementioned methods are commonly used to generate substitution boxes, there is other mechanism to obtain good S-Boxes based on construct them from smaller ones. The most representative papers in this area of research are the works from Fomin [24] and de la Cruz [25]. In both papers, the authors present an 8-bit permutation having nonlinearity 108, differential uniformity 6 and optimal values of algebraic degree and graph algebraic immunity. Moreover, such S-Boxes present the best nonlinearity and differential uniformity known, up to date, for any 8-bit substitution box with minimum value of algebraic degree equals to 7 and graph algebraic immunity equals to 3.

1.1 Contribution of this paper

The main contribution of this paper rest on the design of a new optimization algorithm for efficient search of 4-bit substitution boxes which can be part of construction schemes for generation of cryptographically strong 8-bit permutations. Other contributions are made in the form of analysis on the average values of significant properties of s-boxes produced by constructions in [24, 25], as well as the proposal of a new butterfly structure based on

Fomin's A and B constructions from [24] and the consequent application of the proposed optimization method to produce a quality set of 8-bit S-Boxes for application in symmetric cryptographic algorithms.

2 Preliminaries

Let V_n the n-dimensional vector space over the finite field $\mathbb{F}_2$ with two elements, 0 and 1. Hence, any binary vector $x \in V_n$ is of the form $\{0,1\}^n$. We denote $\mathbb{F}_{2^n}$ the finite field with 2^n elements for some irreducible polynomial $\lambda(x)$ of degree n. Moreover, one can identify the vector space V_n with the field $\mathbb{F}_{2^n}$ defining an isomorphism for a fixed basis (Section 2.1). We also refer to substitution boxes as *(n,m)-functions*, S-Boxes or permutations (indicating bijection).

2.1 Substitution boxes and their representations

An *(n,m)-function* f is a mapping from V_n into V_m, i.e., $f : V_n \to V_m$. When $m = 1$, f is called a Boolean function and if $m > 1$ the function f is known as vectorial Boolean function or substitution box (S-Box).

Any substitution box S, can be defined as the vector $S = (f_1, f_2, ..., f_m)$ where the Boolean functions $f_i : V_n \to V_1$ are called the coordinate functions of S. The set of all linear combinations of the coordinate functions is called the component functions of S, which are usually involved for determining the cryptographic properties of the substitution [26].

S-Boxes can be represented as a list of values (lookup table) with each output value ranging from 0 to $2^m - 1$. Altogether, the S-Box can be represented as the binary matrix of 2^n rows and m columns, where each column represent one of the coordinate functions of the substitution, which is known as truth table. In addition, when representing an *n-bit* S-Box, it may be convenient to identify the vector space V_n with the finite field with 2^n elements, $\mathbb{F}_{2^n}$. Then, for any fixed basis of V_n defining an isomorphism between V_n and $\mathbb{F}_{2^n}$, the *n-bit* S-Box S can be represented as a univariate polynomial in $\mathbb{F}_{2^n}[X]$ [26]:

$$S(X) = \sum_{i=0}^{2^n-1} A_i X^i, A_i \in \mathbb{F}_{2^n} \tag{1}$$

The result from Proposition 2.4 of [26] resolves that the values of A correspond to the values of S. Moreover, this polynomial representation is unique, since if not, there would exist two distinct polynomials of degree less than

or equal to $(2^n - 1)$ taking the same value at 2^n distinct points, which is impossible. One should note that the univariate representation is dependent on the basis chosen for identifying V_n and $\mathbb{F}_{2^n}$.

Finally, we refer to the *algebraic normal form (ANF)* representation of each component function of one substitution S, given that the algebraic degree and the algebraic immunity of both, Boolean functions and s-boxes, is related to this representation. Let $f : V_n \to V_1$ be an arbitrary n-variable Boolean function. For some fixed $i = 0, 1, ..., n-1$, f can be written as a sum over $\mathbb{F}_2$ of distinct t-order products of its arguments, $0 \leq t \leq n-1$; this is called the *algebraic normal form* of f [25].

2.2 Properties

Let $S : V_n \to V_m$ be a substitution box, S is said to be balanced if each value $x \in V_m$ appears the same number of 2^{n-m} times. When $n = m$, it is usual that S is a bijective mapping from V_n to itself, i.e, that each output appears exactly once. Such S-Boxes are permutations on V_n [27]. This paper is restricted to the study of bijective 8×8 S-Boxes only. Although, general definitions are given to all properties regardless the values of n and m.

2.2.1 Nonlinearity

The Walsh-Hadamard transform of one *(n,m)-function* S is defined as [27]:

$$W_S(x, y) = \sum_{z \in V_n} (-1)^{\langle y, S(z) \rangle \oplus \langle x, z \rangle} \tag{2}$$

where $x \in V_n$, $y \in V_m^* = V_m \backslash \{0\}$ and $\langle a, b \rangle = \oplus_{i=1}^{k} a_i b_i$ is the inner product of the vectors $a, b \in V_k$. Here, $\oplus$ represents the addition modulo two or bitwise e**X**clusive **OR** (XOR). The superset $\Lambda_S = \{(x, y) : V_n \times V_m^* | W_S(x, y)\}$ is known as the Walsh-Hadamard spectrum of S.

The maximum absolute value of Λ_S is known as the linearity of S and it is denoted in this paper as $\mathcal{L}_S$. The other property related to the Walsh-Hadamard spectrum of S-Boxes is called nonlinearity of the S-Box [27]. Nonlinearity is directly related to maximum absolute value of the Walsh-Hadamard spectrum, i.e, $\mathcal{L}_S$. Hence, one can express nonlinearity as follows:

$$\mathcal{N}_S = \frac{2^n - \mathcal{L}_S}{2} \tag{3}$$

The maximum achievable nonlinearity for bijective S-Boxes cannot be greater than the Sidelnikov-Chabaud-Vaudenay (SCV) bound [28]:

$$\mathcal{N}_S \leq 2^{n-1} - 2^{\frac{n-1}{2}} \tag{4}$$

n	$\mathcal{N}_S$	δ_S
3	2	2
4	4	4
5	12	2
6	24	2
7	56	2
8	112	4

Table 1: Best known values of nonlinearity and differential uniformity for bijective s-boxes of dimensions from 3×3 up to 8×8.

and when equality holds in (4) we talk about of almost bent (AB) functions. As for APN condition (see Section 2.2.2, [29]), AB functions only exists when n is an odd number [27]. When n is even, the best value of nonlinearity is obtained through power permutations and equals to [26, 27, 30]:

$$\mathcal{N}_S = 2^{n-1} - 2^{\frac{n}{2}} \tag{5}$$

Table 1 present the best know values of nonlinearity (also differential uniformity) for bijective substitution boxes of *n-variables*, $n \in \{3, 4, 5, 6, 7, 8\}$.

2.2.2 Differential uniformity

According to Nyberg [29, 31], for any function $S : V_n \to V_m$ and any $x \in V_n$ and $y \in V_m$ one can define

$$\delta(x, y) = \#\{z \in V_n : S(x + z) + S(z) = y\} \tag{6}$$

Then the multi-set $\Delta_S = \{x \in V_n, b \in V_m : \delta(x, y)\}$ represents the differential spectrum of S, and its maximum

$$\delta_S = \max_{x \neq 0, y} \delta(x, y) \tag{7}$$

is called the differential uniformity of S. For any bijective S-Box S the differential uniformity of S satisfies $\delta_S \geq 2$ [29]. The S-Boxes for which equality holds are called almost perfect nonlinear (APN) functions. It is worth noticing that the APN condition only exists for odd number of variables, and when $n = 6$ [32]. In the case of even number of variables, the best known differential uniformity value is 4 [26, 27, 29, 30].

2.2.3 Algebraic degree and algebraic immunity

Recall from Section 2.1 the representation of one s-box in the algebraic normal form. The algebraic degree of a Boolean function $f : V_n \to V_1$ is the

maximum order of the terms appearing in its algebraic normal form. Hence, the algebraic degree of a substitution box $S : V_n \to V_m$ is the maximum algebraic degree of its component functions [33], denoted as $deg(S)$. Moreover, one should note that the minimum degree of S, i.e., the smallest degree of the component Boolean functions of S, must be as high as possible [25]. In this paper we denote such degree as ρ_S. For balanced n-variable S-Boxes, the following inequality holds for the algebraic degree of the same [25]:

$$1 \leq \rho_S \leq deg(S) \leq n - 1 \tag{8}$$

The annihilator of a Boolean function $f : V_n \to V_1$ is a Boolean function $g : V_n \to V_1$ such that $f \cdot g = 0$ [25]. For any Boolean function f, the algebraic immunity of f is the minimum value d such that f or $f \oplus 1$ has nonzero annihilator of degree d. There are different definitions of the algebraic immunity of S-Boxes [27]. Particularly, we focus on the graph algebraic immunity and before introduce its definition, let review the concept of annihilating set. Let be $U \subseteq V_{2n}$, then the annihilating set of U is defined as [25]:

$$\{p \in \mathbb{F}_2[z_1, ..., z_{2n}] | p(U) = 0\} \tag{9}$$

then, the algebraic immunity of U is:

$$AI(U) = \min\{deg\ p | 0 \neq p \in \mathbb{F}_2[z_1, ..., z_{2n}], p(U) = 0\} \tag{10}$$

Let $S : V_n \to V_m$ be an arbitrary S-Box, the graph algebraic immunity of S is defined as [25]:

$$AI_{gr}(S) = \min\{deg\ p | 0 \neq p \in \mathbb{F}_2[z_1, ..., z_{2n}], p(gr(S)) = 0\} \tag{11}$$

where $gr(S) = \{(x, S(x)) | x \in V_n\} \subseteq V_{2n}$. In [33], some bound on the values of the algebraic immunity is given. The graph algebraic immunity of S is upper bounded by the value d, which is the minimum positive integer that satisfies:

$$\sum_{i=0}^{d} \frac{2n!}{(2n-i)! \cdot i!} > 2^n \tag{12}$$

3 Construction of 2k-bit permutations from k-bit permutations and finite field multiplication in $\mathbb{F}_{2^k}$

This section briefly resume the works from de la Cruz [25] and Fomin [24] to construct cryptographically strong substitution boxes from smaller 4-bit components.

The early work from de la Cruz introduced a new method for generation of 2k-bit bijective substitution boxes using k-bit components and finite field multiplication in $\mathbb{F}_{2^k}$ [25]. In the particular case of generation of 8-bit substitution boxes, de la Cruz propose to work over the finite field $\mathbb{F}_{2^4}$ using one of the three existing irreducible polynomials, such that $\lambda(x) = x^4 + x + 1$ [25]. Moreover, the polynomial permutation $P_d(x) = x^{2^k-2}$ can be expressed as $P_d(x) = x^{14}$.

De la Cruz established a set of criteria to determine whether the obtained substitution box can be considered for application in symmetric encryption algorithms or not, based on the properties mentioned in the previous section. In addition, one must verify the absence of fixed points in the generated substitution once applied the construction **H** [25].

$$\mathcal{N}_S \geq 100 \quad \delta_S \leq 8 \quad \rho_S = 7 \quad AI_{gr}(S) = 3$$

The largest experiment conducted by de la Cruz, involving 2^{20} 4-bit permutations achieves substitution boxes having nonlinearity up to 108 and, at most, differentially 6-uniform. The best values of differential uniformity and nonlinearity (6, 108) were obtained for $h_1 = h_2 = \{0, 1, e, 9, f, 5, c, 2, b, a, 4, 8, d, 6, 3, 7\}$. Nonetheless, de la Cruz shown that it is not necessary that $h_1 = h_2$ to obtain s-boxes which satisfy the aforementioned criteria on their properties.

Later in 2019, Fomin propose several constructions with a butterfly structure derivated from functions based on the constructions of Mairoma-MacFarland [24]. Fomin discussed the results achieved through two different schemes based in two A constructions and two B constructions [24].

These constructions receive four different 4-bit components h_i and π_i, $i = 1, 2$. The permutations h_i are selected from the search space of 4-bit bijective S-Boxes, while permutations π_i are result from monomial power functions such that the final 8-bit substitution box have almost optimal nonlinearity and differential uniformity. Fomin studied the behavior of different combinations of components for each construction concluding that there are at least 36 different substitution boxes, having the same values of the best S-Box reported in [25]. Additionally, the construction from [25] is generalized to a butterfly structure, partitioning the obtained 8-bit permutations according the monomial power function employed as component(s) of the construction [24].

For the purposes of this paper, we subject to analysis the constructions **H** [25] and **AA** [24]. In the remaining of this section we briefly summarize the average behavior of aforementioned properties for S-Boxes generated by

these constructions.

Depending on the selection of the 4-bit components h_i (also π_i for construction **AA**) we are in presence of different output scenarios for constructions **H** and **AA**. The particular analysis of construction **AA** is a bit more complex since there are four different outcomes according the values of h_i and π_i:

$$h_1 \neq h_2,\, \pi_1 \neq \pi_2 \quad h_1 \neq h_2,\, \pi_1 = \pi_2 \quad h_1 = h_2,\, \pi_1 \neq \pi_2 \quad h_1 = h_2,\, \pi_1 = \pi_2$$

With regards to the values of the 4-bit monomial permutations π_1 and π_2 we assume the following criteria based on the analysis of results presented by Fomin in [24]:

- If $\pi_1 \neq \pi_2$ then $\pi_1(x) = x^{14}$ and $\pi_2(x) = x^{11}$
- If $\pi_1 = \pi_2$ then $\pi_i(x) = x^{14}$

Table 2 present the average values of nonlinearity, differential uniformity, minimum algebraic degree and graph algebraic immunity of 100000 8-bit substitution boxes generated by constructions **H** and **AA**. As one can see in table 2, the average value of the referred properties indicate that the common values for nonlinearity, differential uniformity, minimum degree and graph algebraic immunity are 104, 8, 6 and 3 respectively. The amount of S-Boxes having nonlinearity 106, regardless the configuration settings, is below 1000 on each case, which mean that less than 1% of the cases generate such subtitutions. Moreover, no substitution with nonlinearity value equals to 108 was reported. Hence, the cases where substitution boxes have the best values of nonlinearity and differential uniformity reported in [24, 25] are not frequently produced by these constructions.

Construction	Configuration	N_S	δ_S	ρ_S	$\Gamma_{gr}(S)$
H	$h_1 = h_2$	103.32	7.5	6.29	2.96
	$h_1 \neq h_2$	103.26	7.64	6.08	2.96
AA	$h_1 \neq h_2,\, \pi_1 \neq \pi_2$	103.27	7.64	6.03	2.96
	$h_1 \neq h_2,\, \pi_1 = \pi_2$	103.26	7.64	6.03	2.96
	$h_1 = h_2,\, \pi_1 \neq \pi_2$	103.32	7.51	6.25	2.96
	$h_1 = h_2,\, \pi_1 = \pi_2$	103.32	7.5	6.25	2.96

Table 2: Average values of properties for constructions **H** and **AA**.

4 Optimization algorithms as the tool to obtain cryptographically strong s-boxes

Evolutionary techniques have been successfully applied to the design of S-Boxes for different purposes. Among the pioneer papers on this area of research are the works from Millan *et al.* [34, 35] and Clark *et al.* [36, 37]. Through the years, the quality of result achieved by heuristic methods applied to evolution of S-Boxes has grown, improving the earlier results in this field. We take as reference the works from Tesař [38], Kazymyrov *et al.* [39], Picek *et al.* [16, 40, 41], Ivanov *et al.* [42, 43], Isa *et al.* and Ahmad *et al.* [44, 45]. Moreover, in 2020, an extense number of research papers dedicated to heuristic generation of S-Boxes with good criptographic properties, or the analysis of existing trade off between some of their properties were published [20, 21, 46, 47, 48, 49, 50]. The best result w.r.t nonlinearity and differential uniformity of any optimization algorithm which receive pseudo-random s-boxes as input input is the achieved, as far as the author knows, by Ivanov *et. al.* in [42], where the authors affirm they produce thousands of S-Boxes having differential uniformity 6 and nonlinearity 104. Better values of nonlinearity and/or differential uniformity obtained by any optimization algorithm is the result of seeding the algorithm with S-Boxes produced through algebraic techniques like substitutions presented in [43].

4.1 An external parameter independent cost function

Heuristic algorithms are sensitive to the initial pool of solutions as well the fitness function(s) used to lead the optimization process. In this section we discuss a novel cost function related to nonlinearity of S-Boxes, which is used in the experiments presented on this paper.

The works from Clark *et al.* [37] and Picek *et al.* [41], present different cost functions related to the property of nonlinearity of S-Boxes. A detailed analysis on these cost functions and their results are given in [37, 38, 41]. Furthermore, a comparison on their performance and the existing relation to nonlinearity is analyzed in [50], where the authors propose a new cost function for evolution of bijective substitution boxes which is independent to any external parameter, unlike those presented in [37, 41].

Let W be the set of all absolute coefficients lower or equals to the SCV bound [28], i.e., $W = \{0, 4, ..., 2^{\frac{n}{2}+1}\}$ given that $n = 2k$ for the purposes of

this paper. Then, the function presented in [50] is defined as:

$$Cost(S) = \sum_{y \in V_n} \sum_{x \in V_n} \prod_{z \in W} ||W_S(x, y)| - z| \tag{13}$$

Propositions 1 and 2 of [50] warranties that minimizing *Cost(S)* will increase the final nonlinearity of the analyzed s-box S. Further information about this cost function is presented in [50].

4.2 Local search algorithm

Based on the results from minor experiments using the optimization algorithm proposed in this section, we decided that permutations h_1 and h_2 are equal. Moreover, in the particular case of construction **AA**, where the values of the 4-bit monomial permutations π_1 and π_2 may differ, it is assumed, for the sake of simplicity, that $\pi_1(x) = \pi_2(x) = x^{14}$.

Let $C \in \{\mathbf{H}, \mathbf{AA}\}$ the selected construction and let define the neighborhood $N(h)$ of any 4-bit permutation h as the S-Boxes result of swapping a pair of output values corresponding to a pair of different inputs in h. The cardinality of $N(h)$, $\#N(h) = \binom{16}{2} = 120$ different S-Boxes. Hence, on each iteration of the local search algorithm, one must review 120 possible solutions, and the best of them substitute the 4-bit permutation h used to generate the 8-bit S-Box through construction C. For any 4-bit permutations $h, h' \in N(h)$, one can say that h' is better than h if:

- $\mathcal{N}_{S_h} < \mathcal{N}_{S_{h'}}$
- $\mathcal{N}_{S_h} < \mathcal{N}_{S_{h'}}$ and $Cost(S_h) > Cost(S_{h'})$

where S_h and $S_{h'}$ refers to S-Boxes generated by construction C using h and h' as 4-bit component respectively. The remaining properties are used as constrains in the optimization process, i.e., the resulting S-Boxes should have differential uniformity at most 6, minimum algebraic degree value equals to 7 and graph algebraic immunity value of 3.

Algorithm 1 present the pseudo-code of the local search employed in this paper for optimization of cryptographically strong 8-bit bijective S-Boxes. The insert mutation [51] (algorithm 2) introduced at the end of the algorithm prevent the method from infinite loops, because if no upgrade is found in the neighborhood of the permutation h, then the algorithm has no other mechanism to review a fresh set of 4-bit components in the next iteration, hence it will fall on an infinite loop. It worth to remark that the algebraic structure of the substitutions obtained through this technique is not altered since the

algorithm is used to explore the space of the 4-bit components instead of modifying the structure of the final 8-bit permutation, which represent an advantage w.r.t other research papers that directly work over the representation of the substitution to be optimized [16, 38, 39, 41, 42, 43, 47, 49, 50].

input : A random 4-bit permutation h and the desired value of nonlinearity $\mathcal{N}_G$
output: An 8-bit permutation having high nonlinearity, low differential uniformity and optimal algebraic degree and immunity.

Initialization
$S \leftarrow C_h$ `// Apply construction C using h as 4-bit component`

while *True* **do**
  upgrade = 0
  if $\rho_S = 7$ ***and*** $AI_{gr}(S) = 3$ ***and*** $\delta_S \leq 6$ ***and*** $\mathcal{N}_S \geq \mathcal{N}_G$ **then**
    return C_h
  end
  else ; `// search in the neighborhood of h a permutation that improves the application of C as explained`
    foreach $h' \in N(h)$ **do**
      $S' \leftarrow C_{h'}$
      if $\mathcal{N}_S < \mathcal{N}_{S'}$ *or* $(\mathcal{N}_S = \mathcal{N}_{S'}$ *and* $Cost(S') < Cost(S))$ **then**
        $S \leftarrow S'$
        $h \leftarrow h'$
        upgrade = 1
      end
    end
    if *upgrade = 0* **then** ; `// Insert mutation on h`
      $h \leftarrow InsertMutation(h, 0, 15)$
      $S \leftarrow C_h$
    end
  end
end

Algorithm 1: Pseudo-code of the local search algorithm.

5 Experimental results

Algorithm 1 stop if we found any 8-bit permutation having nonlinearity greater or equal to $\mathcal{N}_G$ and differential uniformity, minimum degree and graph algebraic immunity equal to the best results presented in [24, 25]. More important, though, is obtain these results reviewing the minimum amount of 4-bit components.

input : An array of elements A and indexes p_1, p_2 such that $(p_1 < p_2)$
output: The array A with value at index p_2 inserted at index p_1

$tmp \leftarrow A[p_2]$

for $i \leftarrow p_2$ *down to* $p_1 + 1$ **do**
| $A[i] \leftarrow A[i-1]$
end
$A[p_1] \leftarrow tmp$
return A

Algorithm 2: Pseudo-code of the insert mutation algorithm.

The first round of experiments using algorithm 1 attempt to measure the performance of the method using the constructions **H** and **AA**. Thus, we set input parameter $\mathcal{N}_G = 106$. Given the low probability of generate one s-box with nonlinearity equal to 108, differential uniformity 6, minimum degree 7 and graph algebraic immunity 3 from random components, the proposed algorithm is restricted to evaluate, at most, $120000 \approx 2^{17}$ solutions, which can be translated is less than five hours of computation in an 8GB of RAM Dell Latitude E7440 with Intel®Core™i7-4600U CPU @ 2.1 GHz. In each round of experiments the local search algorithm was set to perform one hundred individual executions, first to obtain S-Boxes with nonlinearity 106 and then to produce permutations having nonlinearity 108, also satisfying the restrictions on the remaining properties.

For the case of nonlinearity 106, all one hundred executions produce S-Boxes with desired nonlinearity. Moreover, the average solution evaluations to fulfill the stop condition in these experiments is upper bounded by **350**. When input parameter $\mathcal{N}_G$ equals to 108, the local search method produce several S-Boxes with desired nonlinearity which were not taken into account since the values of differential uniformity and/or graph algebraic immunity does not satisfies the restriction imposed to stop the algorithm. However, in all the executions, a permutation with identical properties to the best results from [24, 25] was obtained. Moreover, all resulting substitution boxes are different to each other, which extend the results presented by de la Cruz and Fomin to a large set of 4-bit components and not restricted only to configurations discussed in [24, 25]. The average number of solutions evaluated by the local search algorithm is less than 4000 permutations in both cases, which is lower in magnitude than the bound established to force the termination of the algorithm. In addition, one can see in table 3 that there is a slight advantage of using algorithm 1 over construction **AA**.

Table 3: Average solution evaluations to obtain desired nonlinearity

Construction/N_G	106	108
AA	329.05	3675.21
H	339.38	3999.31

Configuration	N_S	δ_S	ρ_S	$\Gamma_{gr}(S)$
$h_1 \neq h_2$, $\pi_1 \neq \pi_2$	103.27	7.63	6.03	2.96
$h_1 \neq h_2$, $\pi_1 = \pi_2$	103.27	7.63	6.03	2.96
$h_1 = h_2$, $\pi_1 \neq \pi_2$	103.32	7.51	6.26	2.96
$h_1 = h_2$, $\pi_1 = \pi_2$	103.33	7.5	6.26	2.96

Table 4: Average values of properties for construction **AB**.

5.1 Construction based on mixed butterfly structure

The design of mixed butterfly structures based on the constructions proposed by Fomin is left as research proposal in [24]. This section present a simple butterfly structure based in the combination of one A and one B construction. In **AB** structure, the value of y_0 is conceived through construction A. Then, the value of x_0 is obtained through construction B depending, as **AA** and **BB** constructions, on the value of parameter y_0. The high level scheme of butterfly structure **AB** is shown in figure 1.

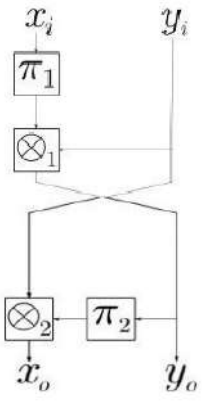

Figure 1: Butterfly structure based on A and B constructions (**AB**)

.

Similarly to the analysis carried in Section 3, we subject to evaluation the average values of the properties from S-boxes generated by construction **AB**, presented in table 4. Here, we want to highlight the almost identical behavior of construction **AB** w.r.t construction **AA** on each configuration setting of the 4-bit components h_i and π_i. Moreover, all three constructions present similar overall behavior, with a minor improvement when $h_1 = h_2$ and $\pi_1 = \pi_2$ in the particular cases of **AA** and **AB**.

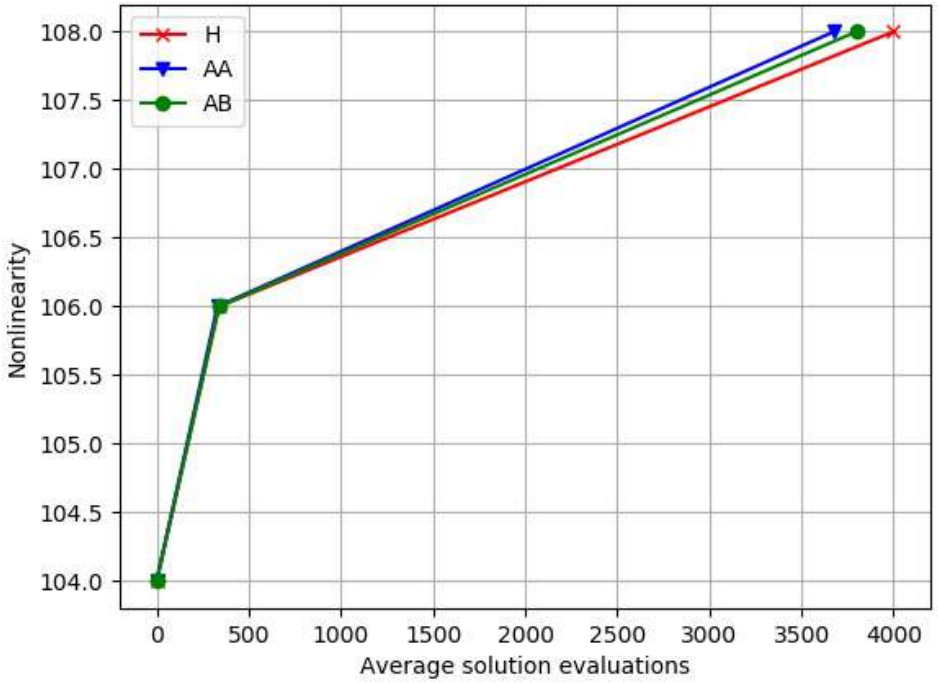

Figure 2: Convergence rate of the local search method.

.

We reproduce the experiments conducted earlier to measure the performance of algorithm 1 for construction **AB**. As expected, the results of the local search method are in the range of the achieved using constructions **H** and **AA**. In the case of $\mathcal{N}_G = 106$, the algorithm repeatedly produce substitutions with desired nonlinearity in an average of 337.82 solution evaluations. When $\mathcal{N}_G = 108$, the average evaluations to generate a S-Box with desired cryptographic properties is approximately 3798 (3797.55) solutions. The convergence of the local search method for constructions **H**, **AA** and **AB** is presented in figure 2. Here, one can observe that the best performance (in terms of solution evaluations) of algorithm 1 is reached using the constructions based on a butterfly structure.

Although the proposed method manages to produce several S-Boxes with identical values of nonlinearity, differential uniformity, algebraic degree and graph algebraic immunity than the achieved in [24, 25], the algorithm was not able to obtain any substitution box having optimal algebraic degree and graph algebraic immunity which present better values of nonlinearity and/or differential uniformity than the best reported in [24, 25]. Hence, it remains as open question if such substitution box exist.

6 Conclusions

In this paper we propose a novel optimization algorithm for efficient exploitation of the space of 4-bit bijective substitution boxes to serve as components of the three constructions analyzed. Moreover, our algorithm present major advantages with respect to other heuristic methods for evolution of S-Boxes. First, the search space is highly reduced from $2^8!$ used in the optimization of 8-bit substitution boxes to $2^4!$ which is the search space of the 4-bit component requested by constructions **H**, **AA** and **AB** (notice that our proposal uses identical 4-bit components for h_i and π_i in case of construction based on butterfly structure). In addition, the structure of the final 8-bit substitution box is never modified by the optimization method, hence the result present a simple interpolation polynomial which may represent an advantage towards efficient masking. Finally, a large set of S-Boxes having good cryptographic properties can be quickly generated.

References

[1] Matsui M., "Linear cryptanalysis method for DES cipher", *Workshop on the Theory and Application of of Cryptographic Techniques*, Springer, 1993, 386–397.

[2] Biham E., Shamir A., "Differential cryptanalysis of DES-like cryptosystems", *Journal of Cryptology*, **4 1** (1991), 3–72.

[3] Armknecht F., "Improving fast algebraic attacks", *nternational Workshop on Fast Software Encryption*, Springer, 2004, 65–82.

[4] Kocher P., Jaffe J., Jun B., "Differential power analysis", *Annual International Cryptology Conference*, Springer, 1999, 388–397.

[5] Kocher P., "Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems", *Annual International Cryptology Conference*, Springer, 1996, 104–113.

[6] Suresh C., Rao J. R., Rohatgi P., "Template attacks", *International Workshop on Cryptographic Hardware and Embedded Systems*, Springer, 2002, 13–28.

[7] Brier E., Clavier C., Olivier F., "orrelation power analysis with a leakage model", *International Workshop on Cryptographic Hardware and Embedded Systems*, Springer, 2004, 16–29.

[8] Gierlichs B., Batina L., Tuyls P., Preneel B., "Mutual information analysis: A generic side-channel distinguisher", *International Workshop on Cryptographic Hardware and Embedded Systems*, Springer, 2008.

[9] Zhang X., Zheng Y., "Gac—the criterion for global avalanche characteristics of cryptographic functions", *The Journal of Universal Computer Science*, Springer, 1996, 320–337.

[10] Webster A. F., Tavares S. E., "On the design of S-Boxes", *Conference on the theory and application of cryptographic techniques*, Springer, 1985, 523–534.

[11] Adams C., Tavares S. E., "The structured design of cryptographically good s-boxes", *Journal of Cryptology*, **3 1** (1990), 27–41.

[12] Guilley S., Hoogvorst P., Pacalet R., "Differential power analysis model and some results", *Smart Card Research and Advanced Applications VI*, Springer, 2004, 127–142.

[13] Prouff E., "DPA attacks and s-boxes", *International Workshop on Fast Software Encryption*, Springer, 2005, 424–441.

[14] Chakraborty K., Sarkar S., Maitra S., Mazumdar B., Mukhopadhyay D., Prouff E., "Redefining the transparency order", *Designs, Codes and Cryptography*, **82 1-2** (2017), 95–115.

[15] Li H., Zhou Y., Ming J., Yang G., Jin C., "The Notion of Transparency Order, Revisited", *The Computer Journal*, **7** (2020).

[16] Picek S., Papagiannopoulos K., Ege B., Batina L., Jakobovic D., "Confused by confusion: Systematic evaluation of DPA resistance of various s-boxes", *International Conference on Cryptology in India*, Springer, 2014, 374–390.

[17] Carlet C., de Chérisey E., Guilley S., Kavut S., Tang D., "Intrinsic resiliency of s-boxes against side-channel attacks–best and worst scenarios", *EEE Transactions on Information Forensics and Security*, **16** (2020), 203–218.

[18] Dimitrov M. M., "On the design of chaos-based s-boxes", *IEEE Access*, **8** (2020), 117173–117181.

[19] Lu Q., Zhu C., Deng X., "An efficient image encryption scheme based on the LSS chaotic map and single s-box", *IEEE Access*, **8** (2020), 25664–25678.

[20] Ahmad M., Al-Solami E., Alghamdi A. M., Yousaf M. A., "Bijective s-boxes method using improved chaotic map-based heuristic search and algebraic group structures", *IEEE Access*, **8** (2020), 110397–110411.

[21] Ahmad M., Khaja I. A., Baz A., Alhakami H., Alhakami W., "Particle swarm optimization based highly nonlinear substitution boxes generation for security applications", *IEEE Access*, **8** (2020), 116132–116147.

[22] Daemen J., Rijmen V., *The design of Rijndael: AES, the advanced encryption standard, Second Edition*, Springer, 2020.

[23] Mangard S., Oswald E., Popp T., *Power analysis attacks: Revealing the secrets of smart cards*, Springer, 2008.

[24] Fomin D. B., "New classes of 8-bit permutations based on a butterfly structure", *Mat. Vopr. Kriptogr.*, **10**:2 (2019), 169–180.

[25] de la Cruz-Jiménez R. A., "Generation of 8-bit s-boxes having almost optimal cryptographic properties using smaller 4-bit s-boxes and finite field multiplication", *International Conference on Cryptology and Information Security in Latin America*, Springer, 2017, 191–206.

[26] Canteaut A., *Lecture notes on cryptographic boolean functions*, Inria (Paris, France), 2016.

[27] Carlet C., Crama Y., Hammer P. L., *Vectorial boolean functions for cryptography*, 2010.

[28] Chabaud F., Vaudenay S., "Links between differential and linear cryptanalysis", *Workshop on the Theory and Application of Cryptographic Techniques*, Springer, 1994, 356–365.

[29] Nyberg K., "On the construction of highly nonlinear permutations", *Workshop on the Theory and Application of Cryptographic Techniques*, Springer, 1992, 92–98.

[30] Budaghyan L., Carlet C., Pott A., "New classes of almost bent and almost perfect nonlinear polynomials", *IEEE Transactions on Information Theory*, **52 3** (2006), 1141–1152.

[31] Nyberg K., "Perfect nonlinear s-boxes", *Workshop on the Theory and Application of Cryptographic Techniques*, Springer, 1991, 378–386.

[32] Browning K. A., Dillon J. F., McQuistan M. T., Wolfe A. J., "An APN permutation in dimension six", *Finite Fields: theory and applications*, **518** (2010), 33–42.

[33] Armknecht F., Krause M., "Constructing single-and multi-output boolean functions with maximal algebraic immunity", *International Colloquium on Automata, Languages, and Programming*, Springer, 2006, 180–191.

[34] Millan W., "How to improve the nonlinearity of bijective s-boxes", *Australasian Conference on Information Security and Privacy*, Springer, 1998, 181–192.

[35] Millan W., Burnett L., Carter G., Clark A., Dawson E., "Evolutionary heuristics for finding cryptographically strong s-boxes", *International Conference on Information and Communications Security*, Springer, 1999, 263–274.

[36] Clark J. A., *Metaheuristic Search as a Cryptological Tool*, 2002, PhD. Thesis, University of York.

[37] Clark J. A., Jacob J. L., Stepney S., "The design of s-boxes by simulated annealing", *New Generation Computing*, **23 3** (2005), 219–231.

[38] Tesař P., "A new method for generating high non-linearity s-boxes", *Radioengineering*, **19 1** (2010), 23–26.

[39] Kazymyrov O. Kazymyrova V., Oliynykov R., "A method for generation of high-nonlinear s-boxes based on gradient descent", *Mat. Vopr. Kriptogr.*, **5**:2 (2014), 71–78.

[40] Picek S., *Applications of evolutionary computation to cryptology*, 2015, PhD. Thesis.

[41] Picek S., Cupic M., Rotim L., "A new cost function for evolution of s-boxes", *Evolutionary Computation*, **24 4** (2016), 695–718.

[42] Ivanov G., Nikolov N., Nikova S., "Cryptographically strong s-boxes generated by modified immune algorithm", *International Conference on Cryptography and Information Security in the Balkans*, Springer, 2015, 31–42.

[43] Ivanov G., Nikolov N., Nikova S., "Reversed genetic algorithms for generation of bijective s-boxes with good cryptographic properties", *Cryptography and Communications*, **8 2** (2016), 247–276.

[44] Isa H., Jamil N., Z'aba M., "Hybrid heuristic methods in constructing cryptographically strong s-boxes", 2016.

[45] Ahmad M., Bhatia D., Hassan Y., "A novel ant colony optimization based scheme for substitution box design", *Procedia Computer Science*, **57** (2015), 572–580.

[46] Wang Y., Zhang Z., Zhang L. Y., Feng J., Gao J., Lei P., "A genetic algorithm for constructing bijective substitution boxes with high nonlinearity", *Information Sciences*, 2020.

[47] Bolufé-Röhler A., Tamayo-Vera D., "Machine learning based metaheuristic hybrids for S-box optimization", *Journal of Ambient Intelligence and Humanized Computing*, 2020, 1–14.

[48] Djurasevic M., Jakobovic D., Picek S., "One property to rule them all? On the limits of trade-offs for S-boxes", *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, 2020, 1064–1072.

[49] Freyre-Echevarría A., Martínez-Díaz I., Legón-Pérez C. M., Sosa-Gómez G. Rojas O., "Evolving Nonlinear S-Boxes With Improved Theoretical Resilience to Power Attacks", *IEEE Access*, **8** (2020), 202728–202737.

[50] Freyre-Echevarría A., Alanezi A., Martínez-Díaz I., Ahmad M., Abd El-Latif A. A., Kolivand H., Razaq A., "An External Parameter Independent Novel Cost Function for Evolving Bijective Substitution-Boxes", *Symmetry*, **12 11** (2020).

[51] Eiben A. E., Smith J. E., *Introduction to evolutionary computing*, Springer, 2003.

Constructing Involutions Specifying Their Coordinate Functions

Daniel Humberto Hernández Piloto and Oliver Coy Puente

Institute of Cryptography, Havana University, Cuba
dhhernandez2410@gmail.com, o.coypuente@gmail.com

Abstract

Involutions, i.e, permutations such that its inverse is itself have an particular interest in Cryptography, because these components are used to decrease the cost of the implementation of decryption process. In this paper we propose some approaches to construct involutions of dimension $n+m$ by existing ones of dimension n, specifying their coordinate functions, where $m \in \mathbb{N}$ and $1 \leqslant m \leqslant n$. For the proposed constructions was calculated some cryptographic parameters as nonlinearity, algebraic degree, differential uniformity, lineal structure and number of fixed points.

Keywords: Permutations, involutions, vectorial boolean functions, coordinate functions, nonlinearity, algebraic degree, differential uniformity, lineal structure, fixed points.

1 Introduction

Substitutions on the finite field $\mathbb{F}_{2^n} = GF(2)[\theta]/z(\theta)$ have a significant impact in many applications such as design of symmetric cryptographic primitives, where $z(\theta)$ is an irreducible polynomial of degree n. Examples of this are the so-called S-boxes, which are used to guarantee one of two essential principles of these cryptographic primitives: the confusion of the input information's bits (see [1]). Exist a several works, where authors propose the construction of several substitutions classes (for example, [2, 3, 4, 5, 6, 7, 8, 9]).

An special kind of permutations on $\mathbb{F}_{2^n}$ are the *involutions*, permutations $F : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$ for which the following condition holds

$$F(F(\beta)) = \beta,$$

for any $\beta \in \mathbb{F}_{2^n}$. In many papers are proposed new methods for constructing permutations of these kinds (for example, [10, 12, 13]), which play a significant role in the design of cryptographic primitives that require the inverse transformations for the decryption of the information, such as SP-networks and others. Involutions find applications in coding theory (for example, [11]) and for constructing bent-functions (for example, [12, 13]). They have been

used frequently in block cipher designs, eg., AES [14], Anubis [15], Khazad [16] and PRINCE [17].

Let $V_n = \{0,1\}^n$, $n \in \mathbb{N}$ be the set of all binary vectors $x^{(n)} = (x_1, \ldots, x_n)$ of dimension n. It is well known that exist a bijective mapping between V_n and $\mathbb{F}_{2^n}$, defined by the correspondence

$$(x_1, x_2 \ldots, x_n) \leftrightarrow [x_1 \cdot \theta^{n-1} + x_2 \cdot \theta^{n-2} + \ldots + x_n]. \tag{1}$$

Using this mapping in what follows we make no difference between the elements of V_n and $\mathbb{F}_{2^n}$.

For a vectorial boolean function $F : V_n \to V_n$ the system of functions $f_i : V_n \to \{0,1\},\ i \in \overline{1,n}$ such that for any $x^{(n)} \in V_n$

$$F\left(x^{(n)}\right) = \left(f_1\left(x^{(n)}\right), \ldots, f_n\left(x^{(n)}\right)\right),$$

is called *coordinate functions* of the function F. In this case we will write $F = (f_1, \ldots, f_n) : V_n \to V_n$. It is evident that $F = (f_1, \ldots, f_n) : V_n \to V_n$ is an involution if and only if for any $x^{(n)} \in V_n$, $i \in \overline{1,n}$

$$f_i\left(F\left(x^{(n)}\right)\right) = x_i. \tag{2}$$

The construction of substitutions by their coordinate functions is a difficult task. Methods for constructing substitutions specifying their coordinate functions are proposed in the works [18, 19]. In this work we propose new methods for constructing involutions on V_n specifying their coordinate functions. In the section 2 our approaches are described. Some cryptographic properties of our construction are proved in the section 3. Results, obtained with the help of computational calculations are showed in the section 4. The conclusion of this work is given in the section 5.

2 Constructing involutions of dimension $n+m$ by existing ones of dimension n

Let be $F = (f_1, \ldots, f_n) : V_n \to V_n$ be a permutation, then the construction of the vectorial boolean function $G = (g_1, \ldots, g_{n+m}) : V_{n+m} \to V_{n+m}$ is defined as follows:

$$g_i\left(x^{(n+m)}\right) = \begin{cases} f_i\left(x^{(n)}\right), & i \in \overline{1,n}, \\ \varphi_{i-n}\left(x^{(i)}\right), & i \in \overline{n+1, n+m}, \end{cases} \tag{3}$$

where $\varphi_j : V_{n+j} \to \{0,1\}$ for any $j \in \overline{1,m}$ are arbitrary boolean functions.

Theorem 1. *The vectorial boolean function* $G = (g_1, \ldots, g_{n+m}) : V_{n+m} \to V_{n+m}$ *defined by the rule (3) is a permutation if and only if the boolean functions* φ_{i-n} *is linear on* x_i *for any* $i \in \overline{n+1, n+m}$.

Proof. Let $\varphi_{i-n}\left(x^{(i)}\right)$ be linear on x_i for any $i \in \overline{n+1, n+m}$, respectively. We need to prove that the vectorial boolean function $G = (g_1, \ldots, g_{n+m}) : V_{n+m} \to V_{n+m}$ defined by the rule (3) is a permutation, i.e. for any $a^{(n+m)} \in V_{n+m} \backslash \{0^{n+m}\}$ the boolean function $\bigoplus\limits_{j=1}^{n+m} a_j g_j(x^{(n+m)})$ is balanced, where $0^{n+m} = (0, \ldots, 0) \in V_{n+m}$. Consider 2 cases:

- case 1: let be $j \in \overline{1, m}$ such that $a_{n+j} \neq 0$, then $\bigoplus\limits_{j=1}^{n+m} a_j g_j(x^{(n+m)})$ is linear on x_{n+j}, which is equivalent to it is a balanced boolean function,
- case 2: for any $j \in \overline{1, m}$, $a_{n+j} = 0$, then $a^{(n)} \neq 0^{(n)}$ and $\bigoplus\limits_{j=1}^{n+m} a_j g_j(x^{(n+m)}) = \bigoplus\limits_{j=1}^{n} a_j f_j(x^{(n)})$, which is equivalent to it is a balanced boolean function.

Consider now that G is a permutation, we need to prove that for any $i \in \overline{n+1, n+m}$ the boolean functions $\varphi_{i-n}\left(x^{(i)}\right)$ is linear on x_i. We prove it without loss of generality for $m = 1$. As G is a permutation on V_{n+1}, then for any $y^{(n+1)} \in V_{n+1}$ exist a unique $x^{(n+1)} \in V_{n+1}$ such that $f_i\left(x^{(n)}\right) = y_i$ for all $i \in \overline{1, n}$ and $\varphi_1\left(x^{(n+1)}\right) = y_{n+1}$. Assume the contrary, and so suppose that $\varphi_1\left(x^{(n+1)}\right)$ is of the form $\varphi_1\left(x^{(n+1)}\right) = x_{n+1} h_1\left(x^{(n)}\right) \oplus h_2\left(x^{(n)}\right)$, where $h_1\left(x^{(n)}\right) \not\equiv 0, 1$, i.e., $\varphi_1\left(x^{(n+1)}\right)$ is not linear on the variable x_{n+1}. We select $y^{(n+1)} \in V_{n+1}$ such that $G(x^{(n+1)}) = y^{(n+1)}$, where $x^{(n+1)} = (\alpha_1, \ldots, \alpha_n, x_{n+1})$ for fixed $(\alpha_1, \ldots, \alpha_n) \in V_n$, and $h_1(\alpha_1, \ldots, \alpha_n) = 0$. Then

$$\varphi_1(\alpha_1, \ldots, \alpha_n, x_{n+1}) = h_2(\alpha_1, \ldots, \alpha_n) = y_{n+1}$$

does not depend of the x_{n+1} and consequently for $y^{(n+1)}$ exist $x^{(n+1)} = $ $= (\alpha_1, \ldots, \alpha_n, 0)$, $x^{(n+1)} = (\alpha_1, \ldots, \alpha_n, 1)$ such that $G(x^{(n+1)}) = y^{(n+1)}$ and hence G is not bijective. Consequently the function $\varphi_1\left(x^{(n+1)}\right)$ must be linear on the variable x_{n+1}. □

Using as base the algorithm 7.1 (page 228) of the work [9], which was used to create permutation over V_{n+1} specifying their coordinate functions, we propose a new method for constructing involutions on V_{n+m} by existing ones on V_n, where $1 \leqslant m \leqslant n$.

Construction 1: Let be $F = (f_1, \ldots, f_n) : V_n \to V_n$ be a permutation, then the construction of the vectorial boolean function $G = (g_1, \ldots, g_{n+m}) : V_{n+m} \to V_{n+m}$ is defined as follows:

$$g_i\left(x^{(n+m)}\right) = \begin{cases} f_i\left(x^{(n)}\right), & i \in \overline{1,n}, \\ x_i \oplus \psi_{i-n}\left(x^{(i-1)}\right), & i \in \overline{n+1,n+m}, \end{cases} \tag{4}$$

where $x^{(i-1)} = (x_1, \ldots, x_{i-1})$ and $\psi_j : V_{n+j-1} \to \{0,1\}$ for any $j \in \overline{1,m}$ are arbitrary boolean functions.

Theorem 2. *The vectorial boolean function $G = (g_1, \ldots, g_{n+m}) : V_{n+m} \to V_{n+m}$ defined by the rule (4) is an involution if and only if the permutation $F = (f_1, \ldots, f_n) : V_n \to V_n$ is an involution too and for any $j \in \overline{1,m}$, $x^{(n+j-1)} \in V_{n+j-1}$ holds*

$$\psi_j\left(x^{(n+j-1)}\right) = \psi_j\left(g_1\left(x^{(n+m)}\right), \ldots, g_{n+j-1}\left(x^{(n+m)}\right)\right). \tag{5}$$

Proof. Let be the vectorial boolean function $G = (g_1, \ldots, g_{n+m}) : V_{n+m} \to V_{n+m}$ defined by the rule (4) an involution, then from (2) we have that for any $i \in \overline{1,n+m}$

$$g_i\left(G\left(x^{(n+m)}\right)\right) = x_i.$$

This is equivalent to:

1. for any $i \in \overline{1,n}$
$$f_i\left(F\left(x^{(n)}\right)\right) = x_i$$
2. for any $i \in \overline{n+1,n+m}$
$$g_i\left(x^{(n+m)}\right) \oplus \psi_{i-n}\left(g_1\left(x^{(n+m)}\right), \ldots, g_{i-1}\left(x^{(n+m)}\right)\right) =$$
$$= x_i \oplus \psi_{i-n}\left(x^{(i)}\right) \oplus \psi_{i-n}\left(g_1\left(x^{(n+m)}\right), \ldots, g_{i-1}\left(x^{(n+m)}\right)\right) = x_i.$$

Then, from 1) and 2), respectively, we obtain that G is an involution if and only if F is an involution too and

$$\psi_j\left(x^{(n+j-1)}\right) = \psi_j\left(g_1\left(x^{(n+m)}\right), \ldots, g_{n+j-1}\left(x^{(n+m)}\right)\right)$$

for any $j \in \overline{1,m}$, $x^{(n+j-1)} \in V_{n+j-1}$. □

Let be $\overline{f} : V_n \to \{0,1\}$ the *logical negation* of the boolean function $f : V_n \to \{0,1\}$, i.e.

$$\overline{f}\left(x^{(n)}\right) = f\left(x^{(n)}\right) \oplus 1,$$

for any $x^{(n)} \in V_n$ (see [21]).

For a vectorial boolean function $F = (f_1, \ldots, f_{n_2}) : V_{n_1} \to V_{n_2},\ n_1, n_2 \in \mathbb{N},\ n_1 \geqslant n_2$, we will write that F *admit linear structures* (see [21]), if exist a vector $a^{(n_1)} \in V_n \backslash \{0^{(n_1)}\}$ such that

$$D_F^{a^{(n_1)}}(x^{(n_1)}) = F\left(x^{(n_1)}\right) \oplus F\left(x^{(n_1)} \oplus a^{(n_1)}\right) = b^{(n_2)} = const$$

for any $x^{(n_1)} \in V_{n_1}$, where $b^{(n_2)} \in V_{n_2}$. In this case vector $a^{(n_1)}$ is called $b^{(n_2)}$—*linear translator* (see [21]), of the vectorial boolean function F. Note that if $n_2 = 1$, then F is a boolean function.

Lemma 1. *Let $F = (f_1, \ldots, f_n) : V_n \to V_n$ be an involution, for the subset $s(k) = \{i_1, \ldots, i_k\} \subseteq \overline{1,n},\ 1 \leqslant i_1 < \ldots < i_k \leqslant n$ the vector $e^{(n)}(s(k)) \in V_n$ is defined as follow*

$$e_i^{(n)}(s(k)) = \begin{cases} 0, & i \in \overline{1,n} \backslash s(k), \\ 1, & i \in s(k), \end{cases}$$

for any $i \in \overline{1,n}$ and the vectorial boolean function $F' = (f_1', \ldots, f_n') : V_n \to V_n$ is defined by the rule

$$f_i'(x^{(n)}) = \begin{cases} f_i(x^{(n)}), & i \in \overline{1,n} \backslash s(k), \\ \overline{f_i}(x^{(n)}), & i \in s(k), \end{cases} \tag{6}$$

for any $i \in \overline{1,n}$. Then F' is an involution if and only if the vector $e^{(n)}(s(k))$ is a $e^{(n)}(s(k))$—linear translator of F.

Proof. Considering that $e^{(n)}(s(k))$ is a $e^{(n)}(s(k))$—linear translator of F we obtain that for any $x^{(n)} \in V_n$

$$F\left(x^{(n)} \oplus e^{(n)}(s(k))\right) \oplus e^{(n)}(s(k)) = F\left(x^{(n)}\right). \tag{7}$$

Since F is an involution, if x^n run all the vector space V_n, then $F(x^n)$ run all the vector space V_n too and we can rewrite the equality (7) as follow

$$F\left(F(x^{(n)}) \oplus e^{(n)}(s(k))\right) \oplus e^{(n)}(s(k)) = F\left(F(x^{(n)})\right) = x^{(n)}. \tag{8}$$

From (6) we obtain that $F'\left(x^{(n)}\right) = F\left(x^{(n)}\right) \oplus e^{(n)}(s(k))$, then (8) is equivalent to

$$F\left(F'\left(x^{(n)}\right)\right) \oplus e^{(n)}(s(k)) = F'\left(F'\left(x^{(n)}\right)\right) = x^{(n)},$$

i.e. F' is an involution. □

Using the logical negation of a boolean function and with the help of the lemma 1, by existing involutions defined by the rule (4) we can describe new involutions.

Theorem 3. *Let $G = (g_1, \ldots, g_{n+m}) : V_{n+m} \to V_{n+m}$ an involution from the theorem 2 and $G' = (g'_1, \ldots, g'_{n+m}) : V_{n+m} \to V_{n+m}$ a vectorial boolean function designed by the rule (6) for a fixed vector $e^{(n+m)}(s(k)) \in V_{n+m}$, where $s(k) = \{i_1, \ldots, i_k\} \subseteq \overline{1, n+m},\ 1 \leqslant i_1 < \ldots < i_k \leqslant n+m$.*

1. *Let $k_1 \leqslant k,\ k_1 \in \mathbb{N}$ be a integer such that $s(k_1) = \{i_1, \ldots, i_{k_1}\} \subseteq \overline{1, n}$. The vectorial boolean function G' is an involution if and only if the vector $e^{(n)}(s(k_1))$ is a $e^{(n)}(s(k_1))$—linear translator of F and for any $j \in \overline{1, m}$*

$$\psi_j\left(x^{(n+j-1)}\right) = \psi_j\left(g'_1\left(x^{(n+m)}\right), \ldots, g'_{n+j-1}\left(x^{(n+m)}\right)\right), \qquad (9)$$

2. *Let be $i_1 > n$. The vectorial boolean function G' is an involution if and only if for any $j \in \overline{1, m}$ holds (9).*

Proof. To prove the point *1* of the theorem consider that G' is an involution, i.e. for any $x^{(n+m)} \in V_{n+m}$

$$G'\left(G'\left(x^{(n+m)}\right)\right) =$$

$$= \left(F'\left(F'\left(x^{(n)}\right)\right), g'_{n+1}\left(G'\left(x^{(n+m)}\right)\right), \ldots, g'_{n+m}\left(G'\left(x^{(n+m)}\right)\right)\right) =$$

$$= \left(x^{(n)}, x_{n+1}, \ldots, x_{n+m}\right),$$

which is true if and only if F' is an involution and for any $j \in \overline{1, m}$ $g'_{n+j}\left(G'\left(x^{(n+m)}\right)\right) = x_{n+j}$. From the lemma 1 we obtain that F' is an involution if and only if the vector $e^{(n)}(s(k_1))$ is a $e^{(n)}(s(k_1))$—linear translator of F; and

$$g'_{n+j}\left(G'\left(x^{(n+m)}\right)\right) = g'_{n+j}\left(x^{(n+m)}\right) \quad \oplus$$

$$\oplus \quad \psi_j\left(g'_1\left(x^{(n+m)}\right), \ldots, g'_{n+j-1}\left(x^{(n+m)}\right)\right) =$$

$$= x_{n+j} \oplus \psi_j\left(x^{(n+j-1)}\right) \oplus \psi_j\left(g'_1\left(x^{(n+m)}\right), \ldots, g'_{n+j-1}\left(x^{(n+m)}\right)\right) = x_{n+j}$$

if and only if $\psi_j\left(x^{(n+j-1)}\right) = \psi_j\left(g'_1\left(x^{(n+m)}\right), \ldots, g'_{n+j-1}\left(x^{(n+m)}\right)\right)$. The point *2* of the theorem can be proved analogously to the prove of the point *1*. □

2.1 Some examples of the system of boolean functions $\psi_1, \ldots, \psi_m$

Of great interest is the description of boolean functions $\psi_j : V_{n+j-1} \to \{0,1\}$ for which the vectorial boolean function $G : V_{n+m} \to V_{n+m}$ defined by the rule (4) is an involution. An example of a class of these functions is described in the following proposition.

Proposition 1. *Let $F = (f_1, \ldots, f_n) : V_n \to V_n$ be an involution, $\psi_1 : V_n \to \{0,1\}$ a boolean function such that for any $x^{(n)} \in V_n$ $\psi_1\left(x^{(n)}\right) = \psi_1\left(F\left(x^{(n)}\right)\right)$ and for $j \in \overline{2,m}$ the boolean functions $\psi_j : V_{n+j-1} \to \{0,1\}$ are defined as*

$$\psi_j\left(x^{(n+j-1)}\right) = h_j^1\left(x^{(n+j-1)}\right) \cdot \psi_{j-1}\left(x^{(n+j-2)}\right) \oplus h_j^2\left(x^{(n+j-1)}\right), \qquad (10)$$

where for any $i \in \{1,2\}$ $h_j^i : V_{n+j-1} \to \{0,1\}$. Then the vectorial boolean function $G = (g_1, \ldots, g_{n+m}) : V_{n+m} \to V_{n+m}$ defined by the rule (4) is an involution if and only if for any $i \in \{1,2\}$ and $j \in \overline{2,m}$

$$h_j^i\left(x^{(n+j-1)}\right) = h_j^i\left(g_1\left(x^{(n+m)}\right), \ldots, g_{n+j-1}\left(x^{(n+m)}\right)\right).$$

Proof. Suppose that for any $i \in \{1,2\}$ and $j \in \overline{2,m}$

$$h_j^i\left(x^{(n+j-1)}\right) = h_j^i\left(g_1\left(x^{(n+m)}\right), \ldots, g_{n+j-1}\left(x^{(n+m)}\right)\right),$$

then from the theorem 2 we have that is sufficient to show that for any $j \in \overline{1,m}$ holds (5). Let's show this by induction on $j \in \overline{1,m}$. For $j = 1$ the equation (5) follows from the condition $\psi_1\left(x^{(n)}\right) = \psi_1\left(F\left(x^{(n)}\right)\right)$. Suppose that (5) is true for any $j \in \overline{2,m-1}$. For $j = m$ we obtain that

$$\begin{aligned}
&\psi_m\left(g_1\left(x^{(n+m)}\right), \ldots, g_{n+m-1}\left(x^{(n+m)}\right)\right) \quad = \\
= \quad & h_m^1\left(g_1\left(x^{(n+m)}\right), \ldots, g_{n+m-1}\left(x^{(n+m)}\right)\right) \quad \times \\
\times \quad & \psi_{m-1}\left(g_1\left(x^{(n+m)}\right), \ldots, g_{n+m-2}\left(x^{(n+m)}\right)\right) \quad \oplus \\
\oplus \quad & h_m^2\left(g_1\left(x^{(n+m)}\right), \ldots, g_{n+m-1}\left(x^{(n+m)}\right)\right) \quad = \\
= \quad & h_m^1\left(x^{(n+m-1)}\right) \cdot \psi_{m-1}\left(x^{(n+m-2)}\right) \oplus h_m^2\left(x^{(n+m-1)}\right) = \psi_m\left(x^{(n+m-1)}\right).
\end{aligned}$$

□

Corollary 1. *Under the conditions of the proposition 1, if*

$$h_j^1\left(x^{(n+j-1)}\right) = \overline{x}_{n+j-1} \quad \textit{and} \quad h_j^2\left(x^{(n+j-1)}\right) = x_{n+j-1},$$

then the vectorial boolean function $G = (g_1, \ldots, g_{n+m}) : V_{n+m} \to V_{n+m}$ *is an involution.*

In the proposition 1 was described a subclass of the involutions class defined by the theorem 2, where the boolean functions $\psi_j : V_{n+j-1} \to \{0,1\}$, $j \in \overline{2,m}$ are constructed recursively, based on the boolean function $\psi_1 : V_n \to \{0,1\}$. Some examples of the boolean function ψ_1 are given in the corollary 2.

Let $\mathsf{FixP}\,(F)$ be the set of all *fixed points* of the vectorial boolean function $F : V_n \to V_n$, i.e.

$$\mathsf{FixP}\,(F) = \left\{x^{(n)} \in V_n : \ F\left(x^{(n)}\right) = x^{(n)}\right\}.$$

Corollary 2. *Let* $G = (g_1, \ldots, g_{n+m}) : V_{n+m} \to V_{n+m}$ *be a vectorial boolean function defined by the rule (4) and for any* $x^{(n)} \in V_n$ *let* $\psi_1 : V_n \to \{0,1\}$ *be a boolean function defined by one of the following rules:*

1. $\psi_1\left(x^{(n)}\right) = \begin{cases} \varepsilon, & x^{(n)} \in \mathsf{FixP}\,(F), \\ \overline{\varepsilon}, & x^{(n)} \notin \mathsf{FixP}\,(F); \end{cases}$

2. $\psi_1\left(x^{(n)}\right) = h\left(\pi\left(x^{(n)}\right)\right) \oplus h\left(\pi\left(F\left(x^{(n)}\right)\right)\right);$

3. $\psi_1\left(x^{(n)}\right) = h\left(\pi\left(x^{(n)}\right)\right) \cdot h\left(\pi\left(F\left(x^{(n)}\right)\right)\right);$

4. $\psi_1\left(x^{(n)}\right) = h\left(\pi\left(x^{(n)}\right) \oplus \pi\left(F\left(x^{(n)}\right)\right)\right);$

5. $\psi_1\left(x^{(n)}\right) = h\left(z^{(n)}\right);$

for any boolean function $h : V_n \to \{0,1\}$, *where* $\varepsilon \in \{0,1\}$,

$$z^{(n)} = \tau^{-1}\left(\tau\left(\pi\left(x^{(n)}\right)\right) \otimes \tau\left(\pi\left(F\left(x^{(n)}\right)\right)\right)\right),$$

$\pi = (\pi_1, \ldots, \pi_n) : V_n \to V_n$ *is an arbitrary permutation,* $\tau : V_n \to \mathbb{F}_{2^n}$ *is the correspondence defined by the rule (1), and by "*$\otimes$*" is denoted the multiplication of two elements on the finite field* $\mathbb{F}_{2^n}$. *If for any* $j \in \overline{2,m}$ *the boolean function* ψ_j *is designed by the rule (10), then* G *is an involution.*

2.2 Constructing linear involutions of the form (4)

Consider now a particular case of the Construction 1 for constructing linear involutions. The set of all matrices on $\mathbb{F}_2$ of dimension $n \times n$ is denoted by $\mathbb{F}_{2_{n\times n}}$, the set of all invertible matrices from $\mathbb{F}_{2_{n\times n}}$ is denoted by $\mathbb{F}^*_{2_{n\times n}}$ and for the identity and zero matrices of $\mathbb{F}_{2_{n\times n}}$ we use the notations $\mathcal{I}_n$ and $\mathcal{O}_{n,n}$, respectively, for any $n \in \mathbb{N}$. By $x^{(n)\downarrow}$ is denoted the transpose of the vector $x^{(n)} \in V_n$, i.e $x^{(n)\downarrow} = (x_1, \ldots, x_n)^\top$.

Construction 1.1: Let be $\mathcal{U} = (u_{ij})_{n,n} \in \mathbb{F}^*_{2_{n\times n}}$ and the permutation $F = (f_1, \ldots, f_n) : V_n \to V_n$ is defined by the rule

$$F(x^{(n)}) = \left(\mathcal{U} \cdot \ x^{(n)\downarrow}\right)^\top$$

where $x^{(n)} \in V_n$ and for any $i \in \overline{1,n}$

$$f_i(x^{(n)}) = \bigoplus_{j=1}^{n} u_{ij}x_j.$$

Then the construction of the vectorial boolean function $G = (g_1, \ldots, g_{n+m}) : V_{n+m} \to V_{n+m}$ is defined as follows:

$$g_i\left(x^{(n+m)}\right) = \begin{cases} \bigoplus\limits_{j=1}^{n} u_{ij}x_j, & i \in \overline{1,n}, \\ x_i \oplus \bigoplus\limits_{j=1}^{i-1} c_{ij}x_j, & i \in \overline{n+1, n+m}, \end{cases} \tag{11}$$

where $\mathcal{C} = (c_{ij})_{m,n+m} \in \mathbb{F}_{2_{m\times n+m}}$.

Is easy to see that the vectorial boolean function $G = (g_1, \ldots, g_{n+m}) : V_{n+m} \to V_{n+m}$ defined by the rule (11) can be expressed as the multiplication of a matrix $\mathcal{M} \in \mathbb{F}_{2_{n+m\times n+m}}$ by the vector $x^{(n+m)} \in V_{n+m}$, i.e.

$$G\left(x^{(n+m)}\right) = \left(\mathcal{M} \cdot x^{(n+m)\downarrow}\right)^\top,$$

where

$$\mathcal{M} = \left(\begin{array}{c|c} \mathcal{U}_{n,n} & \mathcal{O}_{n,m} \\ \hline \mathcal{V}_{m,n} & \mathcal{W}_{m,m} \end{array}\right)_{n+m\times n+m}, \ \mathcal{C} = (\mathcal{V}_{m,n}|\mathcal{W}_{m,m})_{m\times n+m}$$

$\mathcal{V} = (v_{ij})_{m,n} \in \mathbb{F}_{2_{m\times n}}$ and $\mathcal{W} = (w_{ij})_{m,m} \in \mathbb{F}^*_{2_{m\times m}}$ is a lower triangular matrix of the form

$$\mathcal{W} = \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ w_{21} & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ w_{m1} & \cdots & \cdots & w_{m,m-1} & 1 \end{pmatrix}_{m\times m}$$

Theorem 4. *The vectorial boolean function $G = (g_1, \ldots, g_{n+m}) : V_{n+m} \to V_{n+m}$ defined by the rule (11) is an involution if and only if the permutation $F = (f_1, \ldots, f_n) : V_n \to V_n$ is an involution, $\mathcal{V} = \mathcal{W} \cdot \mathcal{V} \cdot \mathcal{U}$ and $\mathcal{W}^2 = \mathcal{I}_m$.*

Proof. The vectorial boolean function $G = (g_1, \ldots, g_{n+m}) : V_{n+m} \to V_{n+m}$ defined by the rule (11) is an involution if and only if $\mathcal{M}^2 = \mathcal{I}_n$, and this is equivalent to

$$\left(\begin{array}{c|c} \mathcal{U}^2_{n,n} & \mathcal{O}_{n,m} \\ \hline \mathcal{V}_{m,n} \cdot \mathcal{U}_{n,n} \oplus \mathcal{W}_{m,m} \cdot \mathcal{V}_{m,n} & \mathcal{W}^2_{m,m} \end{array}\right)_{n+m \times n+m} = \left(\begin{array}{c|c} \mathcal{I}_n & \mathcal{O}_{n,m} \\ \hline \mathcal{O}_{m,n} & \mathcal{I}_m \end{array}\right)_{n+m \times n+m},$$

which is true if and only if F is an involution, $\mathcal{V} = \mathcal{W} \cdot \mathcal{V} \cdot \mathcal{U}$ and $\mathcal{W}^2 = \mathcal{I}_m$. □

Corollary 3. *Let be $G = (g_1, \ldots, g_{n+m}) : V_{n+m} \to V_{n+m}$ a linear involution from the theorem 4 and the vectorial boolean function $H = (h_1, \ldots, h_{n+m}) : V_{n+m} \to V_{n+m}$ is defined as follows:*

$$H\left(x^{(n+m)}\right) = G\left(x^{(n+m)}\right) \oplus d^{(n+m)}$$

where $d^{(n+m)} \in V_{n+m}$. Then H is an involution if and only if $d^{(n+m)} \in \mathsf{FixP}(G)$.

Remark 1. *Is easy to see that in theorem 4 and corollary 3 are described all linear and affin involutions possible which we can construct using the construction designed by the rule (4).*

3 On some cryptographic properties of the constructed involutions

Exist several cryptographic properties for the characterization of vectorial boolean functions from different points of view (see [21]) with particular interest for the construction of cryptographic primitive. In this section our purpose is to describe some cryptographic properties for our approaches, such that the nonlinearity, the minimum algebraic degree, the differential uniformity, lineal structures and the number of fixed points.

3.1 On the fixed points of the proposed involutions

Proposition 2. *Any vector of the form $\left(x^{(n)}, y^{(m)}\right) \in V_{n+m}$ is a fixed point of the involution G from theorem 2 if and only if $x^{(n)} \in \mathsf{FixP}(F)$, $\psi_1\left(x^{(n)}\right) = 0$, $y^{(m)} \in \Gamma_G\left(x^{(n)}\right)$, where $\Gamma_G\left(x^{(n)}\right)$ is the set of all vectors $\gamma^{(m)} \in$*

V_m such that for all its subvectors $\gamma^{(j-1)} \in V_{j-1}$ the boolean functions $\psi_j\left(x^{(n)}, \gamma^{(j-1)}\right) = 0,\ j \in \overline{2,m}$, i.e.,

$$\Gamma_G\left(x^{(n)}\right) = \left\{\gamma^{(m)} \in V_m : \psi_j\left(x^{(n)}, \gamma^{(j-1)}\right) = 0,\ j \in \overline{2,m},\ \gamma^{(j-1)} \in V_{j-1}\right\}.$$

When the last condition is met

$$|\mathsf{FixP}\,(G)\,| = \sum_{x^{(n)} \in \mathsf{FixP}(F)} \left|\Gamma_G\left(x^{(n)}\right)\right| \leqslant 2^m \cdot |\mathsf{FixP}\,(F)\,|.$$

Proof. If $x^{(n)} \in \mathsf{FixP}\,(F)$ then for any $i \in \overline{1,n},\ y^{(m)} \in V_m$

$$g_i\left(x^{(n)}, y^{(m)}\right) = f_i\left(x^{(n)}\right) = x_i$$

and

$$g_{n+j}\left(x^{(n)}, y^{(m)}\right) = x_{n+j}$$

if and only if $\psi_1\left(x^{(n)}\right) = 0$ and for any $j \in \overline{2,m}$ $\psi_j\left(x^{(n)}, y^{(j-1)}\right) = 0$, i.e. $y^{(m)} \in \Gamma_G\left(x^{(n)}\right)$, which implies that the vector $\left(x^{(n)}, y^{(m)}\right) \in V_{n+m}$ is a fixed point of G. Then we obtain that the number of fixed points of G can be expressed as

$$|\mathsf{FixP}\,(G)\,| = \sum_{x^{(n)} \in \mathsf{FixP}(F)} \left|\Gamma_G\left(x^{(n)}\right)\right|.$$

Is not difficult to see that $\left|\Gamma_G\left(x^{(n)}\right)\right| \leqslant 2^m$, then holds

$$|\mathsf{FixP}\,(G)\,| \leqslant 2^m \cdot |\mathsf{FixP}\,(F)\,|.$$

□

Corollary 4. *Let $G = (g_1, \ldots, g_{n+m}) : V_{n+m} \to V_{n+m}$ be an involution from the theorem 2 and*

1. *if $G' = (g'_1, \ldots, g'_{n+m}) : V_{n+m} \to V_{n+m}$ is an involution defined by the rule (6) from the point 1 of the theorem 3, then*

$$|\mathsf{FixP}(G')| \leqslant 2^m \cdot \left(2^n - |\mathsf{FixP}(F)|\right),$$

2. *if $G' = (g'_1, \ldots, g'_{n+m}) : V_{n+m} \to V_{n+m}$ is an involution defined by the rule (6) from the point 2 of the theorem 3, then*

$$|\mathsf{FixP}\,(G')| \leqslant 2^m \cdot |\mathsf{FixP}\,(F)| - |\mathsf{FixP}\,(G)|.$$

In particular, if G is an involution with the maximum possible number of fixed points for the involutions defined by (4), (i.e. $2^m \cdot |\mathsf{FixP}\,(F)|$); then the involution defined by (6) have not fixed points.

Proof.

1. It is easy to see that if $x^{(n)} \in \mathsf{FixP}(F)$, then $x^{(n)} \notin \mathsf{FixP}(F')$, which means that $\mathsf{FixP}(F) \bigcap \mathsf{FixP}(F') = \varnothing$ and $|\mathsf{FixP}(F')| \leqslant 2^n - |\mathsf{FixP}(F)|$. Then from the proposition 2 we obtain that

$$|\mathsf{FixP}(G')| \leqslant 2^m \cdot |\mathsf{FixP}(F')| \leqslant 2^m \cdot (2^n - |\mathsf{FixP}(F)|)\,.$$

2. It is easy to see that for any $x^{(n)} \in \mathsf{FixP}(F)$, $y^{(m)} \in V_m$ we have $y^{(m)} \in \Gamma_G\left(x^{(n)}\right)$ if and only if $y^{(m)} \notin \Gamma_{G'}\left(x^{(n)}\right)$, which is equivalent to $\Gamma_G\left(x^{(n)}\right) \bigcap \Gamma_{G'}\left(x^{(n)}\right) = \varnothing$. Then $\left|\Gamma_{G'}\left(x^{(n)}\right)\right| \leqslant 2^m - \left|\Gamma_G\left(x^{(n)}\right)\right|$ and

$$|\mathsf{FixP}(G')| = \sum_{x^{(n)} \in \mathsf{FixP}(F)} \left|\Gamma_{G'}\left(x^{(n)}\right)\right| \leqslant \sum_{x^{(n)} \in \mathsf{FixP}(F)} \left(2^m - \left|\Gamma_G\left(x^{(n)}\right)\right|\right) =$$

$$= 2^m \cdot |\mathsf{FixP}(F)| - |\mathsf{FixP}(G)|\,.$$

□

For linear and affin involutions described in the section 2.2 we obtain the followings results on their fixed points. The *rang* of a matrix $\mathcal{A} \in \mathbb{F}_{2_{n\times m}}$ is denoted by $\mathsf{rang}(\mathcal{A})$.

Proposition 3. *Let be $H = (h_1, \ldots, h_{n+m}) : V_{n+m} \to V_{n+m}$ an affin involution from the corollary 3. Then*

$$|\mathsf{FixP}(H)| = \begin{cases} 0, & r \neq r_1, \\ 2^{n+m-r}, & r = r_1, \end{cases}$$

where $r = \mathsf{rang}(\mathcal{M} \oplus \mathcal{I}_{n+m})$ and $r_1 = \mathsf{rang}\left(\mathcal{M} \oplus \mathcal{I}_{n+m} | d^{(n+m)\downarrow}\right)$.

Proof. Is not difficult to see that if $y^{(n+m)} \in \mathsf{FixP}(H)$, then the vector $y^{(n+m)}$ is a solution of the system of linear equations

$$(\mathcal{M} \oplus \mathcal{I}_{n+m}) \cdot x^{(n+m)\downarrow} = d^{(n+m)\downarrow},$$

which have 2^{n+m-r} if $r = r_1$ and don't have solutions if $r \neq r_1$ (see [20]). □

Remark 2. *For the affin involution H from the corollary 3 we obtain that the vector $d^{(n+m)} \in \mathsf{FixP}(G)$ should be chosen that $r \neq r_1$, to obtain an involution without fixed points.*

3.2 On the linear structures and differential uniformity of the proposed involutions

Lemma 2. *If the boolean functions $f : V_n \to \{0,1\}$ is linear on x_i, $i \in \overline{1,n}$, then the vector $e_i^{(n)} = \left(0^{(i-1)}, \varepsilon, 0^{(n-i)}\right) \in V_n$ is ε—linear translator of the boolean function f.*

Proof. If the boolean functions $f : V_n \to \{0,1\}$ is linear on x_i, then exist a boolean function $\hat{f} : V_{n-1} \to \{0,1\}$ such that

$$f\left(x^{(n)}\right) = x_i \oplus \hat{f}\left(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n\right)$$

and

$$f\left(x^{(n)}\right) \oplus f\left(x^{(n)} \oplus e_i^{(n)}\right) =$$

$$= f\left(x^{(n)}\right) \oplus x_i \oplus \varepsilon \oplus \hat{f}\left(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n\right) = \varepsilon.$$

□

Consider the *differential uniformity* of $G : (g_1, \ldots, g_{n+m}) : V_{n+m} \to V_{n+m}$ denoted by $\delta(G)$ and defined as

$$\delta(G) = \max_{\substack{a^{(n+m)} \in V_{n+m} \setminus \{0^{(n+m)}\}, \\ b^{(n+m)} \in V_{n+m}}} \delta_G\left(a^{(n+m)}, b^{(n+m)}\right),$$

where $0^{(n+m)} = (0, \ldots, 0) \in V_{n+m}$ and

$$\delta_G\left(a^{(n+m)}, b^{(n+m)}\right) = \left|\{x^{(n+m)} \in V_{n+m} : D_G^{a^{(n)}}(x^{(n+m)}) = b^{(n+m)}\}\right|.$$

With the help of the lemma 2 we can find the differential uniformity of the described permutations in the theorem 1.

Proposition 4. *Let $G = (g_1, \ldots, g_{n+m}) : V_{n+m} \to V_{n+m}$ be a permutation from the theorem 1. Then G admit linear structures, that is, $\delta(G) = 2^{n+m}$.*

Proof. From the theorem 1 and lemma 2 we have that $\left(0^{(n+m-1)}, 1\right) \in V_{n+m}$ is 1—linear translator of the boolean function $\varphi_m : V_{n+m} \to \{0,1\}$, and this implies that for any $x^{(n+m)} \in V_{n+m}$

$$G\left(x^{(n+m-1)}, \overline{x}_{n+m}\right) \oplus G\left(x^{(n+1)}\right) =$$

$$= \left(0^{(n+m-1)}, \varphi_m\left(x^{(n+m-1)}, \overline{x}_{n+m}\right) \oplus \varphi_m\left(x^{(n+m)}\right)\right) = \left(0^{(n+m-1)}, 1\right),$$

from we obtain the statement of the proposition. □

Remark 3. *The proposition 4 implies that for an involution G from the theorem 2, $\delta(G) = 2^{n+m}$ too.*

3.3 On the minimum algebraic degree of the proposed involutions

Another cryptographic property, which is studied during the construction of vectorial boolean functions is the *minimum algebraic degree* (see [21]), defined by the formula

$$\mathsf{deg}\,(G) = \min_{a^{(n+m)} \in V_{n+m} \setminus \{0^{(n+m)}\}} \mathsf{deg}\left(\left\langle a^{(n+m)}, G\right\rangle\right),$$

for $G : V_{n+m} \to V_{n+m}$. When G is an involution defined by the rule (4) we obtain the higher bound of $deg\,(G)$.

Proposition 5. *Let $G = (g_1, \ldots, g_{n+m}) : V_{n+m} \to V_{n+m}$ be a permutation from the theorem 1. Then*

$$\mathsf{deg}\,(G) \leqslant n - 1.$$

Proof. It is sufficient to see that $\mathsf{deg}\,(G) \leqslant \mathsf{deg}\,(F)$, which is not greater than $n-1$ because F is an permutation too. □

Remark 4. *The proposition 5 implies that for an involution G from the theorem 2, $\mathsf{deg}\,(G) \leqslant n - 1$ too.*

3.4 On the nonlinearity of the proposed involutions

The Walsh transform of a boolean function $f : V_n \to \{0, 1\}$ (see [21]), is defined by the formula

$$\mathsf{W}_f^{b^{(n)}} = \sum_{x^{(n)} \in V_n} (-1)^{f\left(x^{(n)}\right) \oplus \left\langle b^{(n)}, x^{(n)}\right\rangle}$$

for any $b^{(n)} \in V_n$, where $\left\langle b^{(n)}, x^{(n)}\right\rangle = \bigoplus_{i=1}^{n} b_i x_i$.

Consider now the *nonlinearity* (see [21]), of G denoted by $\mathsf{N}\,(G)$ and defined as

$$\mathsf{N}\,(G) = \min_{a^{(n+m)} \in V_{n+m} \setminus \left\{0^{(n+m)}\right\}} \mathsf{N}\left(\left\langle a^{(n+m)}, G\right\rangle\right),$$

where by $\mathsf{N}\left(\left\langle a^{(n+m)}, G\right\rangle\right)$ is denoted the nonlinearity of the boolean function $\left\langle a^{(n+m)}, G\right\rangle = \bigoplus_{i=1}^{n+m} a_i g_i\left(x^{(n+m)}\right)$, which is defined by the formula:

$$\mathsf{N}\left(\left\langle a^{(n+m)}, G\right\rangle\right) = 2^{n+m-1} - \frac{1}{2} \max_{b^{(n+m)} \in V_{n+m}} \left|\mathsf{W}_{\left\langle a^{(n+m)}, G\right\rangle}^{b^{(n+m)}}\right|.$$

Lemma 3. *Let $G = (g_1, \ldots, g_{n+m}) : V_{n+m} \to V_{n+m}$ be an involution from the theorem 2, where $\psi_1 : V_n \to \{0,1\}$ is chosen from the point 1 of the corollary 2. If $\mathsf{FixP}(F) = \varnothing$, then $\mathsf{N}(G) = 0$.*

Proposition 6. *Let $G = (g_1, \ldots, g_{n+m}) : V_{n+m} \to V_{n+m}$ be an involution from the theorem 2. Then*

$$\mathsf{N}(G) = 2 \cdot \min_{a^{(n+m)} \in V_{n+m} \setminus \left\{0^{(n+m)}\right\}} \mathsf{N}\left(\Psi_{F,\psi}^{a^{(n+m)}}\right),$$

where $\psi = (\psi_1, \ldots, \psi_m) : V_{n+m-1} \to V_m$ and the boolean function $\Psi_{F,\psi}^{a^{(n+m)}} : V_{n+m-1} \to \{0,1\}$ is defined by the rule

$$\Psi_{F,\psi}^{a^{(n+m)}}\left(x^{(n+m-1)}\right) = \bigoplus_{i=1}^{n} a_i f_i\left(x^{(n)}\right) \oplus \bigoplus_{j=1}^{m} a_{n+j}\psi_j\left(x^{(n+j-1)}\right).$$

Proof. From (4) we have that

$$\mathsf{W}_{\left\langle a^{(n+m)}, G\right\rangle}^{b^{(n+m)}} = \sum_{x^{(n+m)} \in V_{n+m}} (-1)^{\bigoplus_{i=1}^{n} a_i f_i\left(x^{(n)}\right) \oplus \bigoplus_{j=1}^{m} a_{n+j} g_{n+j}\left(x^{(n+m)}\right) \oplus \left\langle b^{(n+m)}, x^{(n+m)}\right\rangle} =$$

$$= \sum_{x^{(n+m)} \in V_{n+m}} (-1)^{\bigoplus_{i=1}^{n} a_i f_i\left(x^{(n)}\right) \oplus \bigoplus_{j=1}^{m} a_{n+j}\left(x_{n+j} \oplus \psi_j\left(x^{(n+j-1)}\right)\right) \oplus \left\langle b^{(n+m)}, x^{(n+m)}\right\rangle}.$$

Denote by $c^{(n+m)} \in V_{n+m}$ the vector of the form $\left(b^{(n)}, a_{n+1} \oplus b_{n+1}, \ldots, a_{n+m} \oplus b_{n+m}\right)$. Then

$$\mathsf{W}_{\left\langle a^{(n+m)}, G\right\rangle}^{b^{(n+m)}} = \left(1 + (-1)^{c_{n+m}}\right) \times$$

$$\times \sum_{x^{(n+m-1)} \in V_{n+m-1}} (-1)^{\bigoplus_{i=1}^{n} a_i f_i\left(x^{(n)}\right) \oplus \bigoplus_{j=1}^{m} a_{n+j}\psi_j\left(x^{(n+j-1)}\right) \oplus \left\langle c^{(n+m-1)}, x^{(n+m-1)}\right\rangle} =$$

$$= \left(1 + (-1)^{c_{n+m}}\right) \cdot \mathsf{W}_{\Psi_{F,\psi}^{a^{(n+m)}}}^{c^{(n+m-1)}},$$

which implies that

$$\max_{b^{(n+m)} \in V_{n+m}} \left|\mathsf{W}_{\left\langle a^{(n+m)}, G\right\rangle}^{b^{(n+m)}}\right| = 2 \cdot \max_{c^{(n+m-1)} \in V_{n+m-1}} \left|\mathsf{W}_{\Psi_{F,\psi}^{a^{(n+m)}}}^{c^{(n+m-1)}}\right|,$$

where

$$\Psi_{F,\psi}^{a^{(n+m)}}\left(x^{(n+m-1)}\right) = \bigoplus_{i=1}^{n} a_i f_i\left(x^{(n)}\right) \oplus \bigoplus_{j=1}^{m} a_{n+j}\psi_j\left(x^{(n+j-1)}\right).$$

Then we obtain that $\mathsf{N}\left(\left\langle a^{(n+1)}, G\right\rangle\right) = 2 \cdot \mathsf{N}\left(\Psi_{F,\psi}^{a^{(n+m)}}\right)$ and

$$\mathsf{N}(G) = 2 \cdot \min_{a^{(n+m)} \in V_{n+m} \setminus \left\{0^{(n+m)}\right\}} \mathsf{N}\left(\Psi_{F,\psi}^{a^{(n+m)}}\right).$$

□

4 Some computational results

With the help of the software *SageMath* (see [22]), for mathematical and cryptographic research, were calculated the basic cryptographic properties studied in this work for our approaches.

4.1 Non-linear involutions constructed with our approaches

Let be $G = (g_1, \ldots, g_{n+m}) : V_{n+m} \to V_{n+m}$ a involution from the theorem 2, for which the system of boolean functions $\psi_2, \ldots, \psi_m$ is chosen so that for any $j \in \overline{2,m}$ the boolean function ψ_j is defined by the rule (10) and the boolean functions $h_j^1,\ h_j^2$ are chosen from the corollary 1.

In the table 1 are showed these properties in the case when the boolean function $\psi_1 : V_n \to \{0,1\}$ is chosen from the points *1—5* of the corollary 2, respectively. To implement the experiments, several involutions were chosen as F. For the point *1.* of this corollary was chosen $\varepsilon = 1$ and for the points *2—5* of this corollary, several boolean functions were chosen as h and the permutation π was generated random; the best results were obtained when

$$F(\beta) = \beta^{2^n-2}, \quad h\left(x^{(n)}\right) = \begin{cases} \bigoplus\limits_{j=1}^{k} x_j x_{j+k} \oplus x_n, & n \text{ is odd}, \\ \bigoplus\limits_{j=1}^{k} x_j x_{j+k}, & n \text{ is even}, \end{cases}$$

for any $\beta \in \mathbb{F}_{2^n}$, where $k = \begin{cases} \frac{n-1}{2}, & n \text{ is odd}, \\ \frac{n}{2}, & n \text{ is even}. \end{cases}$ We will denote by ψ_1^i the boolean function ψ_1 from the point i of the corollary 2.

ψ_1	n	m	N		deg		$\lvert\mathsf{FixP}(G)\rvert$	$\delta(G)$
			ψ_1	G	ψ_1	G		
ψ_1^1	4	4	2	16	3	2	0	256
	5	3	2	8	4	3	0	256
	6	2	2	4	5	4	0	256
	7	1	2	4	6	5	0	256
ψ_1^2	4	4	4	24	3	2	4	256
	5	3	8	40	4	3	4	256
	6	2	22	64	5	4	4	256
	7	1	50	88	6	5	4	256
ψ_1^3	4	4	3	24	4	3	2	256
	5	3	11	42	5	4	2	256
	6	2	24	40	5	4	4	256
	7	1	43	86	7	6	2	256
ψ_1^4	4	4	4	24	3	2	4	256
	5	3	10	40	4	3	4	256
	6	2	22	64	5	4	4	256
	7	1	50	88	6	5	4	256
ψ_1^5	4	4	3	24	4	3	2	256
	5	3	11	42	5	4	2	256
	6	2	24	64	4	4	0	256
	7	1	49	90	7	6	2	256

Table 1: Some cryptographic properties of G.

4.2 Linear involutions constructed with our approaches

In the section 2.2 were described several approaches for the construction of linear involutions of the form (4), using they matrix representation. Using the notations of this section, in the table 2 are showed some examples of matrices on $\mathbb{F}_2$ that satisfy the conditions of the theorem 4 and the corollary 3, for constructing linear and affin involutions, respectively.

Based on the statements of the proposition 3 and remark 2 we compute the matrices $\mathcal{U}_{n,n}$, $\mathcal{W}_{m,m}$ and the vector $d^{(n+m)}$ so that the involution G has as few fixed points as possible, and at the same time the involution H don't have.

n	m	$\mathcal{U}_{n,n}$	$\mathcal{V}_{m,n}$	$\mathcal{W}_{m,m}$	$\mathcal{M}_{n+m,n+m}$	$d^{(n+m)}$	$\lvert\mathsf{FixP}(G)\rvert$	$\lvert\mathsf{FixP}(H)\rvert$
3	3	$\begin{pmatrix}1&0&0\\0&0&1\\0&1&0\end{pmatrix}$	$\begin{pmatrix}0&0&0\\1&0&0\\0&0&0\end{pmatrix}$	$\begin{pmatrix}1&0&0\\1&1&0\\0&0&1\end{pmatrix}$	$\left(\begin{array}{ccc\|ccc}1&0&0&0&0&0\\0&0&1&0&0&0\\0&1&0&0&0&0\\\hline0&0&0&1&0&0\\1&0&0&1&1&0\\0&0&0&0&0&1\end{array}\right)$	$\begin{pmatrix}0\\0\\0\\\hline0\\0\\1\end{pmatrix}$	16	0
3	4	$\begin{pmatrix}1&0&0\\0&0&1\\0&1&0\end{pmatrix}$	$\begin{pmatrix}0&1&1\\1&1&0\\0&0&0\\1&1&1\end{pmatrix}$	$\begin{pmatrix}1&0&0&0\\1&1&0&0\\0&0&1&0\\0&0&1&1\end{pmatrix}$	$\left(\begin{array}{ccc\|cccc}1&0&0&0&0&0&0\\0&0&1&0&0&0&0\\0&1&0&0&0&0&0\\\hline0&1&1&1&0&0&0\\1&1&0&1&1&0&0\\0&0&0&0&0&1&0\\1&1&1&0&0&1&1\end{array}\right)$	$\begin{pmatrix}1\\0\\0\\\hline1\\0\\1\\0\end{pmatrix}$	16	0
4	4	$\begin{pmatrix}1&0&0&0\\0&1&0&0\\1&1&1&0\\1&1&0&1\end{pmatrix}$	$\begin{pmatrix}1&1&0&0\\0&1&1&0\\1&1&0&0\\1&1&1&0\end{pmatrix}$	$\begin{pmatrix}1&0&0&0\\1&1&0&0\\0&0&1&0\\0&0&1&1\end{pmatrix}$	$\left(\begin{array}{cccc\|cccc}1&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0\\1&1&1&0&0&0&0&0\\1&1&0&1&0&0&0&0\\\hline1&1&0&0&1&0&0&0\\0&1&1&0&1&1&0&0\\1&1&0&0&0&0&1&0\\1&1&1&0&0&0&1&1\end{array}\right)$	$\begin{pmatrix}0\\0\\0\\1\\\hline0\\0\\0\\0\end{pmatrix}$	32	0

Table 2: Some involutions from the Construction 1.1. and their fixed points

5 Conclusions

Constructing permutations by specifying their coordinate functions could be useful when implementing these transformations by using a bit-slicing approach. In this paper we consider th construction of involutions specifying their coordinate functions. We have presented a new construction of involutions of dimension $n + m$ by existing ones of dimension m. For proposed construction are described several subclasses with particular practical interest in terms of an effective implementations as is the case, for example, the recursive system of boolean functions $\psi_1, \ldots, \psi_m$ from the proposition 1. Was also considered a particular case of the presented construction for constructing all linear and affin involutions, which can be created with our approaches.

Several cryptographic properties such that nonlinearity, algebraic degree, differential uniformity, lineal structure and number of fixed points have been calculated in this work for the constructed involutions. For the involutions G from the theorem 2 are described in the point *2* of theorem 4 a method for constructing involutions without fixed points when $|\mathsf{FixP}(G)|$ reaches its maximum value. In the case of presented affin involutions are given the necessary conditions to obtain involutions of this kind without fixed points too. Was proved that proposed involutions admit linear structure, which implies that their differential uniformity will be 2^{n+m}. Practical results of these properties are showed too.

There are several questions about the construction suggested in this work which are left as future work, for example, the existences of non-recursive systems of boolean functions $\psi_1, \ldots, \psi_m$ for which the vectorial boolean functions defined by the rule (4) are involutions; and the existences of systems of boolean functions $\psi_1, \ldots, \psi_m$ for which the nonlinearity of the resulting involutions in the finite field $\mathbb{F}_{2^8}$ is greater than 90.

References

[1] Shannon C.E., "Communication theory of secrecy systems", *The Bell system technical journal 28.4*, 1949, 656–715.

[2] Menyachikhin A.V., "Spectral-linear and spectral-differential methods for generating S-boxes having almost optimal cryptographic parameters", *Mathematical Aspect of Cryptography, 8(2).*, 2017, 97–116.

[3] Fomin D.B., "New Classes of 8-bit Permutations Based on a Butterfly Structure.", *Mathematical Aspect of Cryptography, 10(2).*, 2019, 169–180.

[4] Carlet C., "On the higher order nonlinearities of Boolean functions and S-boxes, and their generalizations", *International Conference on Sequences and Their Applications*, 2008, 345–367.

[5] De La Cruz Jiménez R.A., "Generation of 8-bit S-boxes having almost optimal cryptographic properties using smaller 4-bit S-boxes and finite field multiplication", *International Conference on Cryptology and Information Security in Latin America*, 2017, 191–206.

[6] De La Cruz Jiménez R.A., "On some methods for constructing almost optimal S-Boxes and their resilience against side-channel attacks", *IACR Cryptol. ePrint Arch.*, 2018, 618.

[7] De La Cruz Jiménez R.A., "A method for constructing permutations, involutions and orthomorphisms with strong cryptographic properties", *Prikladnaya Diskretnaya Matematika. Supplement, 12.*, 2019, 145–151.

[8] Hernández Piloto D.H., "2-Transitivity degree for one class of substitutions over finite fields", *Prikladnaya Diskretnaya Matematika, 4*, 2019, 19–26.

[9] Chuan-Kun W., Dengguo F., and others, "Boolean functions and their applications in cryptography", 2016.

[10] Charpin P., Mesnager S., Sarkar S., "Involutions over the Galois field $\mathbb{F}_2$", *Transaction on information theory, 62(4)*, 2016, 2266–2276.

[11] Bulter B.K., "Error Floors of LDPC Codes and Related Topics", 2013.

[12] Coulter R.S., Mesnager S., "Bent functions from involutions over $\mathbb{F}_2$", *Transaction on information theory, 64(2)*, 2018, 2979–2986.

[13] Luo G., Cao W., Mesnager S., "Several new classes of self-dual bent functions derived from involutions", *Cryptography and Communications, 11(6)*, 2019, 1261–1273.

[14] Standard, NIST-FIPS, *Announcing the Advanced Encryption Standard (AES)*, Federal Information Processing Standards Publication 197(1-51), 2001.

[15] Barreto P., *The Anubis block cipher*, NESSIE, 2000.

[16] Barreto, P.S.L.M, Rijmen, *The Khazad legacy-level block cipher*, Primitive submitted to NESSIE 97, 2000.

[17] Borghoff J. and others, "PRINCE — a low-latency block cipher for pervasive computing applications", *International conference on the theory and application of cryptology and information security.*, 2012, 208–225.

[18] Nikonov V.G.1, Sarantsev A.V., "Methods of compact realization of bijective mappings by regular systems of one-type boolean functions", *Discrete and continuous models and applied computational science 2(1)*, 2003, 94–105.

[19] Coy Puente O., "On a construction method for permutations over finite fields", *Computational nanotechnology, 2*, 2019, 85–89.

[20] Glukhov M.M., Elizarov V.P., Nechaev A.A., *Algebra: Uchebnik. - izdanie vtoroe, ispravlennoe i dopolnennoe. [Algebra: Textbook. - second edition, revised and supplemented.]*, Izdatelstvo Lan, Sankt-Peterburg, Moscow, Krasnodar, 2015, In Russian.

[21] Logachev O. A., Salnikov A. A., Smyshlyaev S. V., Yashenko V. V., *Boolean functions in coding theory and cryptology.*, MCCME, Moscow, 2004, In Russian.

[22] *http://www.sagemath.org. — Sage Mathematics Software Version 8.1*, 2018.

Symmetric Cryptography Analysis

On the Impossibility of an Invariant Attack on Kuznyechik

Denis Fomin

Higher School of Economics, Russia
dfomin@hse.ru

Abstract

Currently numerous cryptographic systems are based on SP-networks. These primitives are supposed to be secure but recent investigations show that some attacks are possible. The aim of this work is to study the security of the Russian standardized block cipher Kuznyechik against invariant attacks. We study the known decompositions of its permutation and show the ways of constructing invariant subsets. A new approach to invariant attacks are presented and proved that there is no subsets based on S-Box properties that are invariant under the round functions of Kuznyechik.

Keywords: Kuznyechik, block cipher, invariant attack, nonlinear invariant, decomposition, S-Box, permutation

1 Introduction

Invariant attacks are one of the most known approaches to studing the cryptographic algorithm security based on its structural properties. Modern cryptographic primitives have round based structure and several algorithms have been broken using this type of attack [1, 2, 3].

A lot of researches focused on the cryptographic properties of the Russian standardized block cipher Kuznyechik, [4, 5, 6]. At the same there are currently no known practical attacks on it. Authors [4] have suggested that recently founded decompositions of the permutation of Kuznyechik may lead to some attacks on it. In this work we propose a new approach to generalizing invariant attacks based on the S-Box properties of the algorithm and analyse the resistance of Kuznyechik block cipher on it.

2 Preliminaries

Let $\mathbb{F}_q$ be a finite field of characteristic 2 with $q = 2^p$ elements, $\mathbb{F}_q^n$ — an n-dimensional vector space over $\mathbb{F}_q$. The additive group $(\mathbb{F}_q, \oplus)$ is homomorphic

to the group $(\mathbb{F}_2^p, \oplus)$ with exclusive-or operator $\oplus$. By $\mathrm{GL}_m(q)$ we denote a group of $n \times n$ invertible matrices over $\mathbb{F}_q$.

Block cipher design is based on Shannon's principles of confusion and diffusion [7]. A function $F\colon \mathbb{F}_q^m \to \mathbb{F}_q^m$ of key-alternating substitution-permutation networks (SP-networks or SPN) is composed of a layer of substitution boxes (S-boxes), and a layer of bit permutations. Let

$$F_K(x) = F(x) \oplus K = \mathrm{X}[K]\left(F\left(x\right)\right)$$

be a round function (incuding the key addition), $F(x) = \mathrm{L} \circ \mathrm{S}(x)$, where

- $\mathrm{S}\colon \mathbb{F}_q^m \to \mathbb{F}_q^m$, $\mathrm{S}(x) = \mathrm{S}(x_1, \ldots, x_m) = (\pi(x_1), \ldots, \pi(x_m))$;
- $\mathrm{L}\colon \mathbb{F}_q^m \to \mathbb{F}_q^m$, $\mathrm{L}(x) = x \cdot L$, $L \in \mathrm{GL}_m(q)$, $L = (l_{i,j})_{m\times m}$, $l_{i,j} \in \mathbb{F}_q^*$.

Such an SP-network will be denoted as SPN*.

According to [3] the core idea of nonlinear invariant attack is to find a function $g\colon \mathbb{F}_q^m \to \mathbb{F}_2$ such that there exists many keys K:

$$g\left(F_K(x)\right) = g(x \oplus k) \oplus c = g(x) \oplus g(k) \oplus c \;\forall x \in \mathbb{F}_q^m.$$

In particular, if there exists a subset $\mathcal{G}$ of $\mathbb{F}_q^m$ such that

$$\{F_K(x),\ x \in \mathcal{G}\} = \mathcal{G} \text{ (or for the simplicity } F_K(\mathcal{G}) = \mathcal{G}). \tag{1}$$

for a lot of keys K, the function g is the indicator function of the subset $\mathcal{G}$. This idea can be generalized as follows. Let $\mathcal{G} \subset \mathbb{F}_q^m$, $r \in \mathbb{N}$ and

$$F_{K_{i+r}} \circ \ldots \circ F_{K_i}(\mathcal{G}) = \mathcal{G}$$

for a set of vector of keys $\{(K_i, \ldots, K_{i+r})\}$. The set $\mathcal{G}$ can be used to implement an invariant attack. The problem is to find a way to construct such a subset. The easiest way to solve it is to use the invariants of functions S and L. This paper proposes a different approach, which consists in constructing an invariant for the round transformation, which, in general, is not an invariant of S or L.

Let us present some of the results of the work [8] that are necessary for further exposition. Let $\mathcal{A}$ and $\mathcal{B}$ be a pair of families of sets

$$\mathcal{A} = \{A_1, A_2, \ldots, A_{e_a}\},\ A_i \subseteq \mathbb{F}_q,$$

$$\mathcal{B} = \{B_1, B_2, \ldots, B_{e_b}\},\ B_i \subseteq \mathbb{F}_q$$

and for any $i \in \{1, \ldots, e_a\}$ there exists $j \in \{1, \ldots, e_b\}$ such that $\pi\left(A_i\right) \subseteq B_j$.

Consider the families $\mathcal{A}^m$ and $\mathcal{B}^m$ that are the Cartesian product of families $\mathcal{A}$ and $\mathcal{B}$ correspondingly. Then for any element $A_{i_1} \times \ldots \times A_{i_m} \in \mathcal{A}^m$, there exists an element $B_{j_1} \times \ldots \times B_{j_m} \in \mathcal{B}^m$ such that

$$\mathrm{S}\left(A_{i_1} \times \ldots \times A_{i_m}\right) = \left(\pi(A_{i_1}) \times \ldots \times \pi(A_{i_m})\right) \subseteq B_{j_1} \times \ldots \times B_{j_m}.$$

Suppose that set $\mathcal{G}$ is a subsets of the family $\mathcal{A}^m$ and $r = 0$. That means that there exists a key K such that the following diagram is true:

$$A_{i_1} \times \ldots \times A_{i_m} \xrightarrow{\mathrm{S}} \underbrace{B_{j_1} \times \ldots \times B_{j_m}}_{\in \mathcal{B}^m} \xrightarrow{\mathrm{L}} \underbrace{C}_{\in \mathcal{C}} \xrightarrow{\mathrm{X}[K]} \underbrace{A_{i_1} \times \ldots \times A_{i_m}}_{\in \mathcal{A}^m}. \qquad (2)$$

The obvious consequence of this diagram is the following

Proposition 1 ([8]). *Let $F\colon \mathbb{F}_q^m \to \mathbb{F}_q^m$ be a round function of a key-alternating SPN*. If there exists a key K such that the diagram (2) is true, then the family*

$$C = \mathrm{LS}\left(A_{i_1} \times \ldots \times A_{i_m}\right)$$

has the form $C_{l_1} \times \ldots \times C_{l_m}$, where C_{l_j}, $j \in \{1, \ldots, m\}$ is a subset of a $\mathbb{F}_q$.

Using the same idea we can generalised this approach for $r \geq 0$. Let $\mathfrak{G} = (V, E)$ be an oriented graph, with vertices

$$V = \left\{ A_{i_1} \times \ldots \times A_{i_m} \middle| \, A_{i_j} \subseteq \mathbb{F}_q, j \in \{1, \ldots, m\} \right\}.$$

An edge $\left(A_{i'_1} \times \ldots \times A_{i'_m}, A_{i''_1} \times \ldots \times A_{i''_m}\right)$ is in E if and only if there exists a key K such that

$$F_K\left(A_{i'_1} \times \ldots \times A_{i'_m}\right) = A_{i''_1} \times \ldots \times A_{i''_m}.$$

The generalization of an invariant attack is possible if there exists a cycle in $\mathfrak{G}$. If diagram (2) is true then there exists a loop in $\mathfrak{G}$, if $|E| = 0$ then attack is impossible. If there exists a cycle of length $r + 1$ in $\mathfrak{G}$ then the following diagram is true:

$$\begin{aligned} A_{i_1} \times \ldots \times A_{i_m} \xrightarrow{\mathrm{S}} B_{j_1} \times \ldots \times B_{j_m} \xrightarrow{\mathrm{L}} C_{l_1} \times \ldots \times C_{l_k} \xrightarrow{\mathrm{X}[K_i]} \\ \xrightarrow{\mathrm{X}[K_i]} A_{o_1} \times \ldots \times A_{o_k} \xrightarrow{\mathrm{X}[K_{i+r}]} \ldots \xrightarrow{\mathrm{X}[K_{i+r}]} A_{i_1} \times \ldots \times A_{i_m}. \end{aligned}$$

Then $A_{i_1} \times \ldots \times A_{i_m} \in \mathcal{G}$ and

$$F_{K_{i+r}} \circ \ldots \circ F_{K_i}\left(A_{i_1} \times \ldots \times A_{i_m}\right) = A_{i_1} \times \ldots \times A_{i_m}.$$

Using the graph representation and the fact that $L \in \mathrm{GL}_m(q)$ it's easy to prove

Proposition 2 ([8]). *Let* $F\colon \mathbb{F}_q^m \to \mathbb{F}_q^m$ *be a round function of a key-alternating SPN*,* $A' = A_{i'_1} \times \ldots \times A_{i'_m}$ *and* $A'' = A_{i''_1} \times \ldots \times A_{i''_m}$ *be two vertices of the same cycle of graph* $\mathfrak{G}$*,*

$$B' = \mathrm{S}\,(A')\,,\ C' = \mathrm{LS}\,(A')\,,\ B'' = \mathrm{S}\,(A'')\,,\ C'' = \mathrm{LS}\,(A'')\,.$$

Then

- $B' = B_{j'_1} \times \ldots \times B_{j'_m},\ B'' = B_{j''_1} \times \ldots \times B_{j''_m} \in \mathcal{B}^m,$
- $C' = C_{l'_1} \times \ldots \times C_{l'_m},\ C'' = C_{l''_1} \times \ldots \times C_{l''_m} \in \mathcal{A}^m,$
- $\left|A_{i'_1}\right| = \ldots = \left|A_{i'_m}\right| = \left|B_{j'_1}\right| = \ldots = \left|B_{j'_m}\right| = \left|C_{l'_1}\right| = \ldots = \left|C_{l'_m}\right|,$
- $\left|A_{i'_1}\right| = \left|A_{i''_1}\right|.$

Using these cardinalities relations it is possible to show the algebraic structure of vertices in cycles of $\mathfrak{G}$.

Theorem 1 ([8]). *Let* $F\colon \mathbb{F}_q^m \to \mathbb{F}_q^m$ *be a round function of a key-alternating SPN*,* $A_{i_1} \times \ldots \times A_{i_m}$ *is a vertex on a cycle of graph* $\mathfrak{G}$*,*

- $\mathrm{S}\,(A_{i_1} \times \ldots \times A_{i_m}) = B_{j_1} \times \ldots \times B_{j_m},$
- $\mathrm{L}(B_{j_1} \times \ldots \times B_{j_m}) = C_{l_1} \times \ldots \times C_{l_m}.$

Then

1. A_{i_z}, B_{j_z}, C_{l_z} *are some cosets of* $(\mathbb{F}_q, \oplus)$, $z = \{1, \ldots, m\}$*;*
2. *for any* $z \in \{1, \ldots, m\}$ *there exists* $c \in \mathbb{F}_q$*:* $\pi(c \oplus C_{l_z})$ *is a coset of* $(\mathbb{F}_q, \oplus)$*.*

This theorem sets up the way of finding the invariant subset $\mathcal{G}$. The first we need is to enumerate pairs (A_i, B_i) of coset of $(\mathbb{F}_q, \oplus)$ such that $\pi(A_i) = B_i$.

In this work we analyze Kuznyechik block cipher that is known to be an SPN* and prove that $|E| = 0$. To prove this fact, let us first prove the following theorem.

Theorem 2. *Let* $F\colon \mathbb{F}_q^m \to \mathbb{F}_q^m$ *be a round function of a key-alternating SPN*,* $A_{i_1} \times \ldots \times A_{i_m}$ *is a vertex on a cycle of graph* $\mathfrak{G}$*,* $B_{j_1} \times \ldots \times B_{j_m} = \mathrm{S}\,(A_{i_1} \times \ldots \times A_{i_m})$*. For any* $z \in \{1, \ldots, m\}$ A_{i_z}, $B_{j_z} = \mathsf{B}_{j_z} \oplus b_{j_z}$ *is a coset of* $(\mathbb{F}_q, \oplus)$*,* B_{j_z} *is a subgroup, and*

$$U_z \ = \ \underbrace{\{0\} \times \ldots \times \{0\} \times \mathsf{B}_{j_z}}_{z} \times \{0\} \times \ldots \times \{0\}.$$

Then the set $W_z = \mathrm{L}(U_z)$ *has the form*

$$W_z = W_{z_1} \times \ldots \times W_{z_m},$$

where W_{z_h} *is a coset of* $(\mathbb{F}_q, \oplus)$ *such that there exists a constant* $c_h \colon \pi(W_{z_h} \oplus c_h)$ *is a coset of* $(\mathbb{F}_q, \oplus)$, $h = \{1, \ldots, m\}$.

Proof. Let's consider the set

$$B_{j_1} \times \ldots \times B_{j_m} = \mathsf{B}_{j_1} \times \ldots \times \mathsf{B}_{j_m} \oplus (b_{j_1}, \ldots, b_{j_m}).$$

Without loss of generality consider $h = 1$.

$$\begin{aligned} \mathrm{L}(B_{j_1} \times \ldots \times B_{j_m}) &= \mathrm{L}(\mathsf{B}_{j_1} \times \ldots \times \mathsf{B}_{j_m}) \oplus \mathrm{L}(b_{j_1}, \ldots, b_{j_m}) = \\ &= \mathrm{L}(\{0\} \times \mathsf{B}_{j_2} \times \ldots \times \mathsf{B}_{j_m}) \oplus \mathrm{L}(b_{j_1}, \ldots, b_{j_m}) \oplus \mathrm{L}(U_1) = \\ &= \mathrm{L}(\{0\} \times \mathsf{B}_{j_2} \times \ldots \times \mathsf{B}_{j_m}) \oplus \mathrm{L}(b_{j_1}, \ldots, b_{j_m}) \oplus W_1. \quad (3) \end{aligned}$$

U_1 is a subgroup of $\mathbb{F}_q^m$ then $\mathrm{L}(U_1) = W_1$ is a subgroup of $\mathbb{F}_q^m$ too. Moreover, let

$$W_1 = \left\{ \left(w_{1,1}^{(j)}, \ldots, w_{1,m}^{(j)} \right) \middle| \, j \in \{1, \ldots, |U_1|\} \right\}.$$

Then for any $z = \{1, \ldots, m\}$ $W_{1,z} = \left\{ w_{1,z}^{(j)} \middle| \, j \in \{1, \ldots, |U_1|\} \right\}$ is a subgroup of $\mathbb{F}_q$ because $L = (l_{i,j})_{m \times m}$, $l_{i,j} \in \mathbb{F}_q^*$.

According to the theorem 1

$$\mathrm{L}(B_{j_1} \times \ldots \times B_{j_m}) = C_{l_1} \times \ldots \times C_{l_m} = \mathsf{C}_{l_1} \times \ldots \times \mathsf{C}_{l_m} \oplus (c_{l_1}, \ldots, c_{l_m}), \quad (4)$$

where C_{l_z} is a subgroup of $(\mathbb{F}_q, \oplus)$, $c_{l_z} \in \mathbb{F}_q$, $z \in \{1, \ldots, m\}$. From equations (3) and (4) it follows that the set W_1 is a subset of

$$\mathsf{C}_{l_1} \times \ldots \times \mathsf{C}_{l_m} \oplus (c_{l_1}, \ldots, c_{l_m}) \oplus \mathrm{L}(b_{j_1}, \ldots, b_{j_m})$$

because

$$(0, \ldots, 0) \in \mathrm{L}(\{0\} \times \mathsf{B}_{j_2} \times \ldots \times \mathsf{B}_{j_m}).$$

At the same time $|W_{1,z}| = |\mathsf{C}_{l_1}|$ from which it follows that $\mathsf{C}_{l_1} = W_{1,z}$. Using the theorem 1 this theorem is proven. □

3 Kuznyechik permutation properties

Increased attention has been paid to the permutation of the Russian startedized algorithm Kuznyechik [9] in recent years. Its first decomposition was found by Alex Biryukov, Leo Perrin, and Aleksei Udovenko [10]. In this

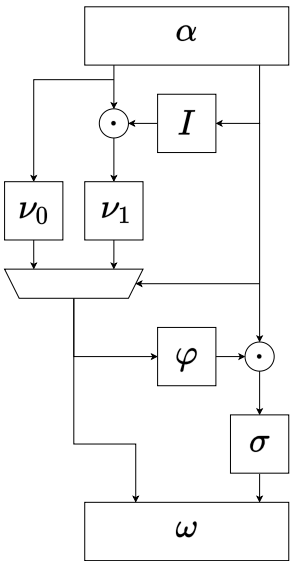

Figure 1: BPU-decomposition, [10]

work we call it BPU-decomposition. Some other interesting properties were found in [11, 4]. BPU-decomposition has a rather simple design (see fig. 1) which can be used for efficient implementation on various platforms [12].

The following algorithm was found by [10] to implement the S-Box of Kuznyechik. Let $\mathbb{F}_{2^4} = GF(2^4, \cdot, \oplus) = GF(2)[y]/(f(y))$ be a finite field with 2^4 elements and irreducible polynomial $f(y) = y^4 \oplus y^3 \oplus 1$. Every element $x \in \mathbb{F}_{2^8}$ can be considered as a concatenation of $l, r \in \mathbb{F}_{2^4}$ using bit representation of x:

$$x = (x_1, \ldots, x_8) = (l \| r),\ l = (x_1, \ldots, x_4),\ r = (x_5, \ldots, x_8).$$

Using this bijection algorithm from [10] can be presented as follows:

1. $(l \parallel r) := \alpha(l \parallel r)$,
2. **if** $r = 0$, **then** $l := \nu_0(l)$, **else** $l := \nu_1(l \cdot I(r))$;
3. $r := \sigma(r \cdot \varphi(l))$,
4. **return** $(l \parallel r) := \omega(l \parallel r)$,

where nonlinear transformations ν_0, ν_1, I, σ, φ are given in the following table (we consider that elements of $\mathbb{F}_{2^4}$ can be shown in hexadecimal repre-

sentation):

I	0,	1,	c,	8,	6,	f,	4,	e,	3,	d,	b,	a,	2,	9,	7,	5
ν_0	2,	5,	3,	b,	6,	9,	e,	a,	0,	4,	f,	l,	8,	d,	c,	7
ν_1	7,	6,	c,	9,	0,	f,	8,	1,	4,	5,	b,	e,	d,	2,	3,	a
φ	b,	2,	b,	8,	c,	4,	1,	c,	6,	3,	5,	8,	e,	3,	6,	b
σ	c,	d,	0,	4,	8,	b,	a,	e,	3,	9,	5,	2,	f,	1,	6,	7

and linear transformations α and ω are the following:

$$\alpha = \begin{pmatrix} 0&0&0&0&1&0&0&0\\ 0&1&0&0&0&0&0&1\\ 0&1&0&0&0&0&1&1\\ 1&1&1&0&1&1&1&1\\ 1&0&0&0&1&0&1&0\\ 0&1&0&0&0&1&0&0\\ 0&0&0&1&1&0&1&0\\ 0&0&1&0&0&0&0&0 \end{pmatrix}, \quad \omega = \begin{pmatrix} 0&0&0&0&1&0&1&0\\ 0&0&0&0&0&1&0&0\\ 0&0&1&0&0&0&0&0\\ 1&0&0&1&1&0&1&0\\ 0&0&0&0&1&0&0&0\\ 0&1&0&0&0&1&0&0\\ 1&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&1 \end{pmatrix}.$$

According to Theorem 1 , we must look for coset A_i, which are maped by the S-Box to some coset B_i. Let us show that the BPU-decomposition allows us to extract such cosets.

Proposition 3. *For S-Box π of Kuznyechik there exist two pairs of subgroup $(\mathsf{A}_i, \mathsf{B}_i)$*

- $\mathsf{A}_1 = \left\{ \alpha^{-1}(0\mathrm{xd} \cdot x \| x) \middle| \, x \in \mathbb{F}_{2^4} \right\}$, $\mathsf{B}_1 = \{ \beta(0\|y) | \, y \in \mathbb{F}_{2^4} \}$,
- $\mathsf{A}_2 = \left\{ \alpha^{-1}(x\|0) \middle| \, x \in \mathbb{F}_{2^4} \right\}$, $\mathsf{B}_2 = \{ \beta(y\|0) | \, y \in \mathbb{F}_{2^4} \}$,

such that there exist $a, b \in \mathbb{F}_2^8$: $\pi(\mathsf{A}_i \oplus a) = \mathsf{B}_i \oplus b$.

Proof. Let $\widehat{\pi}$ be an affine-equivalent permutation of π (see fig. 2):

$$\widehat{\pi}(x) = \omega^{-1}\left(\pi\left(\alpha^{-1}(x)\right) \oplus (0\mathrm{x}2\|0\mathrm{xc})\right).$$

If we show that for

- $\mathsf{A}'_1 = \{ (0\mathrm{xd} \cdot x \| x) | \, x \in \mathbb{F}_{2^4} \}$, $\mathsf{B}'_1 = \{ (0\|y) | \, y \in \mathbb{F}_{2^4} \}$,
- $\mathsf{A}'_2 = \{ (x\|0) | \, x \in \mathbb{F}_{2^4} \}$, $\mathsf{B}'_2 = \{ (y\|0) | \, y \in \mathbb{F}_{2^4} \}$,

the equation $\widehat{\pi}(\mathsf{A}'_i) = \mathsf{B}'_i$ is true for every $i \in \{1, 2\}$, we'll proof the proposition.

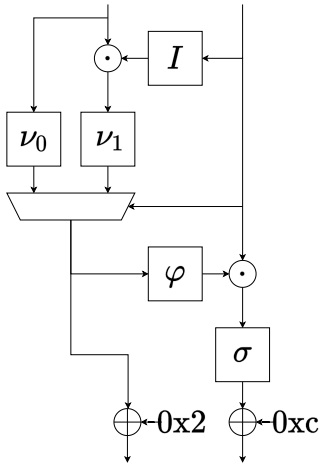

Figure 2: Decomposition of $\widehat{\pi}$

Without loss of generality, let's consider the case $i = 1$ (fig. 3) the case $i = 2$ can be considered similarly (fig. 4).

If x is not equal to 0, then $x \cdot 0\text{xd} \cdot x^{-1} = 0\text{xd}$ is a constant and $\nu_1(0\text{xd}) \oplus 0\text{x2} = 0\text{x0}$.

It's obvious that $\widehat{\pi}$ maps the set $\{(x \cdot 0\text{xd} \| x),\ x \in \mathbb{F}_{2^4}^*\}$ to $\{(0\|y),\ y \in \mathbb{F}_{2^4}^*\}$ because of the facts: $\varphi(0\text{x2}) \neq 0$, $x_1 \cdot 0\text{x2} = x_2 \cdot 0\text{x2} \Leftrightarrow x_1 = x_2$, σ is a bijection and $\sigma(0) = 0\text{xc}$.

If x is equal to 0 then $\widehat{\pi}(0\|0) = (0\|0)$.

□

The proved proposition only indicates that such cosets exist, but does not prove that others do not exist. To enumerate them all, consider an algorithm that works for any permutation. Let $\text{span}(S)$ be a linear span of a set S. Using ideas from [13] the following algorithm can be proposed:

Algorithm 1.

1. $i := 0$
2. **for every** $a, b \in \mathbb{F}_q$:
   (a) $\mathsf{A}_i \leftarrow \{0\}$;
   (b) $\mathsf{B}_i \leftarrow \text{span}(\pi(\mathsf{A}_i \oplus a) \oplus b)$;
   (c) $\mathsf{A}_i \leftarrow \text{span}\left(\pi^{-1}(\mathsf{A}_i \oplus b) \oplus a\right)$;
   (d) **if** $\mathsf{A}_i = \text{span}(\mathsf{A}_i)$ **then:**
      – **if** $|\mathsf{A}_i| \neq 2^8$, **print**$(A_i = \mathsf{A}_i \oplus a, B_i = \mathsf{B}_i \oplus b)$, $i \leftarrow i + 1$;

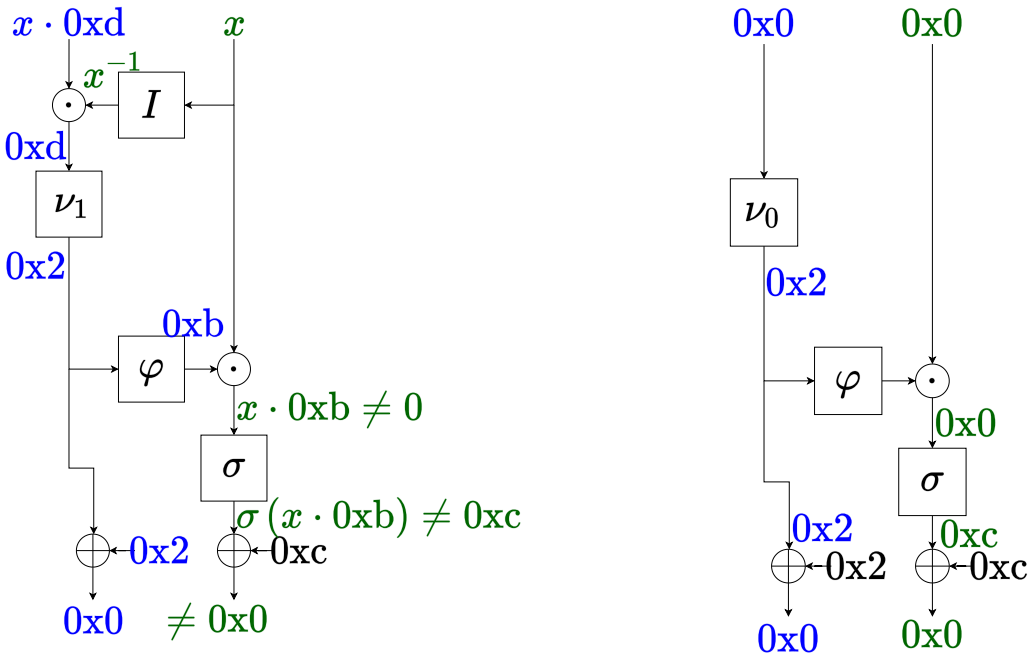

Figure 3: $\widehat{\pi}$ maps A'_1 to B'_1

– **for every** $x \in \mathbb{F}_2^8 \backslash \mathsf{A}_i$: $\mathsf{A}_i \leftarrow \mathrm{span}\,(\mathsf{A}_i \cup x)$, **go to step** (2.b);

Proposition 4. *The algorithm 1 is correct.*

Proof. It's obvious that if there exists a coset $A_i \subset \mathbb{F}_2^8$ such that a permutation π maps it into coset $B_i \subset \mathbb{F}_2^8$ then algorithm 1 will print it. □

Definition 1. *A pair of sets* $(\mathsf{A}_i, \mathsf{B}_i)$ *is I pair of sets for a permutation* $\pi\colon \mathbb{F}_q \to \mathbb{F}_q$ *if there exist* $a, b \in \mathbb{F}_q$ *such that*

$$\pi(\mathsf{A}_i \oplus a) = \mathsf{B}_i \oplus b.$$

Subspaces A_i *and* B_i *are called LI and RI sets for* π *correspondingly.*

In proprosition 3 we found two I pairs of sets $(\mathsf{A}_i, \mathsf{B}_i)$ for permutation π and every set consists of 16 elements. Using the algorithm 1 one can find such pairs of sets of any size. We implemented it and found:

– 2 I pairs $(\mathsf{A}_i, \mathsf{B}_i)$, $|\mathsf{A}_i| = |\mathsf{B}_i| = 16$;

– 1 943 I pairs $(\mathsf{A}_i, \mathsf{B}_i)$, $|\mathsf{A}_i| = |\mathsf{B}_i| = 4$;

– 2 730 I pairs $(\mathsf{A}_i, \mathsf{B}_i)$, $|\mathsf{A}_i| = |\mathsf{B}_i| = 2$.

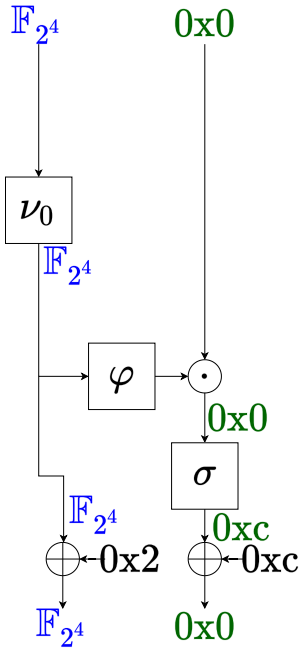

Figure 4: Invariant subspace of $\widehat{\pi}$

4 Impossibility attack details

Using the theorem 2 we can propose the following approach to prove the impossibility of the invariant attack. Let $(\mathsf{A}_i, \mathsf{B}_i)$ is an I pair for permutation π. Consider

$$B_i^{(j)} = \underbrace{\{0\} \times \ldots \times \{0\}}_{j-1} \times \mathsf{B}_i \times \{0\} \times \ldots \times \{0\},$$

$$\mathrm{L}\left(B_i^{(j)}\right) = C_i^{(j)} = \left\{ \left(c_{i,k}^{(j,1)}, \ldots, c_{i,k}^{(j,m)} \right),\ k = 1, \ldots, |\mathsf{B}_i| \right\}.$$

And from the theorem 2 it follows that every set

$$C_i^{(j,l)} = \left\{ c_{i,k}^{(j,l)},\ k = 1, \ldots, |\mathsf{B}_i| \right\}$$

must be A_d — a subset of an LI set for π. Then

$$\exists\, c_1, c_2 \in \mathbb{F}_{2^4} \colon \pi\left(\mathsf{A}_d \oplus c_1\right) \oplus c_2$$

is a subgroup of $(\mathbb{F}_q, \oplus)$. Using computer calculation and the ideas above we proved the following

Proposition 5. *Let π is a permutation* L *is a linear and* S *is a nonlinear transform of Kuznyechik algorithm, Then for every I pair $(\mathsf{A}_i, \mathsf{B}_i)$, $|\mathsf{B}_i| > 1$,*

for permutation π *and for every* $j = \{1, \ldots, m\}$*, there exist* $l = \{1, \ldots, m\}$ *such that* $C_i^{(j,l)}$ *is not a subset of any subgroup* A_d *such that*

$$\exists\ c_1, c_2 \in \mathbb{F}_{2^4} \colon \pi\left(\mathsf{A}_d \oplus c_1\right) \oplus c_2$$

is a subgroup of $(\mathbb{F}_q, \oplus)$.

Let's consider the most interesting example and take into account I pair of sets $(\mathsf{A}_i, \mathsf{B}_i)$ from the proposition 3:

- $\mathsf{A}_1 = \{0\text{x}00, 0\text{x}05, 0\text{x}22, 0\text{x}27, 0\text{x}49, 0\text{x}4\text{c}, 0\text{x}6\text{b}, 0\text{x}6\text{e}, 0\text{x}8\text{b}, 0\text{x}8\text{e}, 0\text{xa}9, 0\text{xac}, 0\text{xc}2, 0\text{xc}7, 0\text{xe}0, 0\text{xe}5\}$, $\mathsf{B}_1 = \{0\text{x}00, 0\text{x}01, 0\text{x}0\text{a}, 0\text{x}0\text{b}, 0\text{x}44, 0\text{x}45, 0\text{x}4\text{e}, 0\text{x}4\text{f}, 0\text{x}92, 0\text{x}93, 0\text{x}98, 0\text{x}99, 0\text{xd}6, 0\text{xd}7, 0\text{xdc}, 0\text{xdd}\}$;
- $\mathsf{A}_2 = \{0\text{x}00, 0\text{x}01, 0\text{x}0\text{a}, 0\text{x}0\text{b}, 0\text{x}44, 0\text{x}45, 0\text{x}4\text{e}, 0\text{x}4\text{f}, 0\text{x}92, 0\text{x}93, 0\text{x}98, 0\text{x}99, 0\text{xd}6, 0\text{xd}7, 0\text{xdc}, 0\text{xdd}\}$, $\mathsf{B}_2 = \{0\text{x}00, 0\text{x}02, 0\text{x}04, 0\text{x}06, 0\text{x}10, 0\text{x}12, 0\text{x}14, 0\text{x}16, 0\text{x}20, 0\text{x}22, 0\text{x}24, 0\text{x}26, 0\text{x}30, 0\text{x}32, 0\text{x}34, 0\text{x}36\}$;

It's the largest LI and RI sets for π. We also can mention that $\mathsf{B}_1 = \mathsf{A}_2$. If we consider

$$B_1^1 = \mathsf{B}_1 \times \{0\} \times \ldots \times \{0\}$$

then $C_1^{1,1} = \mathsf{B}_1 = \mathsf{A}_2$ because according to the [9] the linear transform of Kuznyechik is based on LFSR with the least feedback coefficient equals to $e \in \mathbb{F}_2^8$. At the same time $C_1^{1,2} \neq \mathsf{A}_1 \oplus a$ nor $C_1^{1,2} \neq \mathsf{A}_2 \oplus a$ for any $a \in \mathbb{F}_{2^8}$ that means that A_{i_1} in $\mathcal{G}$ is not $\mathsf{A}_1 \oplus c$ for any $c \in \mathbb{F}_{2^8}$. Much simpler with

$$B_2^1 = \mathsf{B}_2 \times \{0\} \times \ldots \times \{0\}.$$

In that case $C_2^{1,1} = \mathsf{B}_2 \neq \mathsf{A}_1$ and $C_2^{1,1} = \mathsf{B}_2 \neq \mathsf{A}_1$.

5 Conclusion

We presented a new approach to invariant attacks based on the S-box properties of an SPN*. Kuznyechik is an SPN* since it has a linear layer based on MDS-matrix. Using a computer calculation we enumerated all I pairs for permutation π of Kuznyechik algorithm and proved the impossibility of a generalised invariant attack.

References

[1] Gregor Leander, Mohamed Ahmed Abdelraheem, Hoda AlKhzaimi, and Erik Zenner., "A Cryptanalysis of PRINTcipher: The Invariant Subspace Attack.", Lecture Notes in Computer Science *In Phillip Rogaway, editor,* CRYPTO, **6841** (2011), 206–221.

[2] Gregor Leander, Brice Minaud, and Sondre Rønjom., "A Generic Approach to Invariant Subspace Attacks: Cryptanalysis of Robin, iSCREAM and Zorro. *IACR Cryptology ePrint Archive*, 2015:68", 2015 `https://eprint.iacr.org/2015/68`.

[3] Yosuke Todo, Gregor Leander, and Yu Sasaki., "Nonlinear Invariant Attack – Practical Attack on Full SCREAM, iSCREAM, and Midori64. Cryptology ePrint Archive, Report 2016/732 `https://eprint.iacr.org/2016/732`", 2016.

[4] Léo Perrin., "Partitions in the S-Box of Streebog and Kuznyechik. *IACR Cryptology ePrint Archive*, 2019:92", 2019 `https://eprint.iacr.org/2019/92`.

[5] Vitaly Kiryukhin., "An algorithm for bounding non-minimum weight differentials in 2-round LSX-ciphers. Cryptology ePrint Archive, Report 2020/1208", 2020 `https://eprint.iacr.org/2020/1208`.

[6] Riham AlTawy and Amr M. Youssef., "A Meet in the Middle Attack on Reduced Round Kuznyechik. Cryptology ePrint Archive, Report 2015/096", 2015 `https://eprint.iacr.org/2015/096`.

[7] Henk C. A. van Tilborg, editor., "*Encyclopedia of Cryptography and Security.*", *Springer*, 2005.

[8] D.I. Trifonov and D.B. Fomin., "Invariant Subspaces in SPN Block Cipher.", Applied Discrete Mathematics,, **??**:?? (2021), ??–?? Manuscript submitted for publication..

[9] "GOST R 34.12-2015 Information technology. Cryptographic data security. Block ciphers.", 2015.

[10] Alex Biryukov, Léo Perrin, and Aleksei Udovenko., "Reverse-Engineering the S-Box of Streebog, Kuznyechik and STRIBOBr1.", Lecture Notes in Computer Science *In Marc Fischlin and Jean-Sébastien Coron, editors,* EUROCRYPT (1), **9665** (2016), 372–402.

[11] Léo Perrin and Aleksei Udovenko., "Exponential S-Boxes: a Link Between the S-Boxes of BelT and Kuznyechik/Streebog.", IACR Trans. Symmetric Cryptol., **2** (2016), 99–124.

[12] Alexander Smirnov Denis Fomin Olga Avraamova, Vladimir Serov and Vasily Shokhov., "A Compact Bit-sliced Representation of Kuznechik S-box. In *CTCrypt'20*", 2020.

[13] Gregor Leander., "On Invariant Attacks.", 2019 Invited talk..

Algebraic Cryptanalysis of Round-reduced Lightweight Ciphers Simon and Speck

Aleksandr Kutsenko[1,2], Natalia Atutova[1,2], Darya Zyubina[1,2], Ekaterina Maro[3], and Stepan Filippov[4]

[1]Sobolev Institute of Mathematics, Novosibirsk, Russia
[2]Novosibirsk State University, Novosibirsk, Russia
[3]Southern Federal University, Taganrog, Russia
[4]Saint Petersburg State University, Saint Petersburg, Russia
alexandrkutsenko@bk.ru, atutova.n@yandex.ru, zyubinadarya@gmail.com, eamaro@sfedu.ru, filippowstepan@yandex.ru

Abstract

This paper presents algebraic attacks on Simon and Speck, two families of lightweight block ciphers having LRX- and ARX-structures respectively. They were presented by the U.S. National Security Agency in 2013 and later standardized by ISO as a part of the RFID air interface standard. We algebraically encode the ciphers and try to solve the underlying systems with different SAT solvers, methods based on the linearization and for the first time apply to these ciphers the approaches that use the sparsity of the considered systems of equations. The linearization parameters in systems of equations for both of the ciphers are estimated. A comparison of the efficiency of the used methods is provided.

Keywords: algebraic cryptanalysis, block cipher, lightweight, Simon, Speck

Lightweight cryptography is a research direction of current interest. This is due to the fact that the impact and the usage of RFID tags, FPGAs, smartcards, mobile phones, sensor networks and other cryptographic algorithms for resource-constrained devices continuously grows and becomes more and more important. Lightweight cryptographic primitives are designed to be both efficient and secure for limited resources. In this case the problem of obtaining the trade-off between the security and efficiency, measured by different metrics, appears.

There were developed a number of lightweight block and stream ciphers, hash functions with a purpose of obtaining the aforementioned trade-off. For example, lightweight block ciphers designs include, but are not limited to, HIGHT [1], KATAN [2], KLEIN [3], Piccolo [4] and PRESENT [5].

In 2013, the NSA introduced the specifications of lightweight block cipher families SIMON and SPECK that were claimed to be flexible enough to provide excellent performance in both hardware and software environments. SIMON has been optimized for performance on hardware devices, and SPECK for performance in software. But it was emphasized that both families performed exceptionally well in both hardware and software, providing the platform flexibility required by future applications. As of October 2018, the Simon and Speck ciphers have been standardized by ISO as a part of the RFID air interface standard, International Standard ISO/29167-21 (Information technology — Automatic identification and data capture techniques — Part 21: Crypto suite SIMON security services for air interface communications) and International Standard ISO/29167-22 (Information technology — Automatic identification and data capture techniques — Part 22: Crypto suite SPECK security services for air interface communications), that makes them available for use by commercial entities.

There are no specific cryptanalytic results nor analysis provided in the specification document. However, later there appeared a couple of works related to the cryptanalysis of these ciphers. Mostly differential attacks are under consideration. For instance, in paper [6] differential cryptanalysis of round-reduced SIMON and SPECK was considered. The attacks on up to slightly more than half the number of rounds were described and the drawback of the intensive optimizations in these ciphers was concluded.

The considered ciphers are representatives of LRX- and ARX- structures of block ciphers, the core of them is the explicit usage of nonlinear algebraic operations instead of S-boxes. It leads to the problem of algebraic analysis of these ciphers. Algebraic analysis of SIMON was made by Raddum in [7]. Combined algebraic and truncated differential cryptanalysis on reduced-round SIMON appeared in paper of Courtois et al. [8]. The resistance of SIMON-64/128 with respect to algebraic attacks was studied by using a SAT solver and ElimLin algorithm. In article [9] the usage of SAT-solvers for algebraic cryptanalysis of ARX-structures was discussed. Recently, in paper [10] the attack on up to 13 rounds with 8 chosen plaintexts by fixing 4 and 6 key bits for Simon-32/64 and Simon-64/128 was presented.

In current work we study and compare the efficiency of different types of algebraic attacks on round-reduced SIMON and SPECK. The analysis is provided via different SAT solvers usage as well as methods of solving systems of polynonomial equations, based on the linearization routine. The methods that exploit the sparsity of the systems of equations (The Raddum-Semaev description of the system and the Algorithm) are also considered. This is the

first attempt to analyze the resiliense of SPECK cipher to different algebraic attacks outside the SAT-solvers usage. The conclusion on the obtained results is given.

1 SIMON and SPECK families of ciphers

1.1 General description of SIMON

SIMON is a family of lightweight block ciphers for an optimal hardware performance, presented in [11]. Simon has structure of classical Feistel scheme, in each round $2n$-bit input of the round is divided into two n-bit halves. Each round of SIMON applies a non-linear, non-bijective round function $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ to the left half L of the state. The output of F is added using XOR to the right half R along with a round key k, and the two halves are swapped. The round function F is defined as

$$F(x) = (S^8(x) \odot S^1(x)) \oplus S^2(x), \quad x \in \mathbb{F}_2^n,$$

where $S^j(x)$ denotes left rotation of $x \in \mathbb{F}_2^n$ by j positions and the symbol $\odot$ is for binary operation AND.

We introduce a new variable for each output of the bitwise operation $\odot$, then to describe T rounds we get $n(T-2)$ quadratic equations in $n(T-2)+k$ unknowns. Where n is a word size, T is a number of rounds and k is a key length.

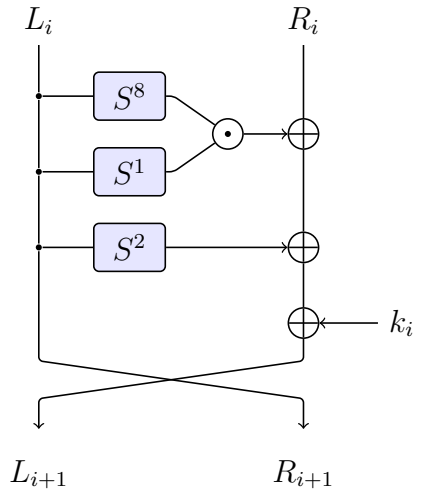

Figure 1: Round function of SIMON

The key schedule of SIMON is described as a function that operates on two, three or four n-bit word registers, depending on the size of the general key. It performs two rotations to the right: $S^{-3}(x)$ and $S^{-1}(x)$ and XOR the

results together with a fixed constant $c = 2^n - 4$ and five constant sequences depending on the version of the specification. These constant sequences are obtained by using three square matrices of order 5 over the field $\mathbb{F}_2$, and a linear feedback shift register where the first two are of period 31 and the last three have the period 62. The general secret key consists of m key words, each of n bits length, where $m \in \{2, 3, 4\}$.

1.1.1 Key schedules

The first m keys are set, each consisting of n bits. The sequence of keys is calculated recursively ($c = 2^n - 4$ is a constant, and z_j is a fixed periodical sequence, exact value see in [11]). The value of m depends on the values of the block size $2n$ and the number of rounds T (Table 1)

$$k_{i+m} = \begin{cases} c \oplus (z_j)_i \oplus k_i \oplus \left(I \oplus S^{-1}\right) S^{-3} k_{i+1}, & \text{for } m = 2, \\ c \oplus (z_j)_i \oplus k_i \oplus \left(I \oplus S^{-1}\right) S^{-3} k_{i+2}, & \text{for } m = 3, \\ c \oplus (z_j)_i \oplus k_i \oplus \left(I \oplus S^{-1}\right) (S^{-3} k_{i+3} \oplus k_{i+1}), & \text{for } m = 4. \end{cases}$$

Block size $2n$	**Key size** mn	**Word size** n	**Key words** m	**const seq**	**Rounds** T
32	**64**	16	4	z_0	32
48	**72**	24	3	z_0	36
	96		4	z_1	36
64	**96**	32	3	z_2	42
	128		4	z_3	44
96	**96**	48	2	z_2	52
	144		3	z_3	54
128	**128**	64	2	z_2	68
	192		3	z_3	69
	256		4	z_4	72

Table 1: SIMON parameters

1.2 General description of SPECK

SPECK is a family of lightweight block ciphers for excellent performance in both hardware and software, but have been optimized for performance on microcontrollers. This family was also presented in paper [11]. In each round

$2n$-bit input of the round is divided into two n-bit halves. Each round of Speck applies a non-linear round function is defined as

$$R_k(x, y) \to \big(\big(S^{-\alpha}(x) + y\big) \oplus k, S^{\beta}(y) \oplus \big(S^{-\alpha}(x) + y\big) \oplus k\big),$$

where $S^j(x)$ denotes left rotation (if $j > 0$) by j positions and right rotation (if $j < 0$) of $x \in \mathbb{F}_2^n$, the symbol «+» is an addition modulo 2^n. The parameters have following values: $\alpha = 7$ and $\beta = 2$ if $n = 16$ (block size is equal to 32) and $\alpha = 8$ and $\beta = 3$ otherwise.

On the first round there will be only $2n$ equations because initially we set two n-bit words. On the next encryption rounds, $2 \cdot (8n - 3)$ equations are added each time (for $m = 1$ $7n - 3$ equations are added on key schedules and $8n - 3$ on a round function) and $3n$ unknowns. Starting from the second round, 3 unknowns are added due to the key schedule. When constructing a system of equations, we substitute the input and output cipher before the first round and after the last (L_0, R_0, L_n, R_n), so the number of unknowns is reduced by $4n$. The final formulas for the number of equations and the number of unknowns are

$$e = \begin{cases} (7n - 3)(T - 1) + (8n - 3)(T - 1) + 2n, & \text{for } m = 1, \\ 2(8n - 3)(T - 1) + 2n, & \text{for } m = 2, 3, 4, \end{cases}$$

$$u = \begin{cases} n(5T - 4), & \text{for } m = 1, \\ n(6T - 5), & \text{for } m = 2, 3, 4. \end{cases}$$

where e is a number of equations, u is a number of variables, n is a word size, T is a number of rounds.

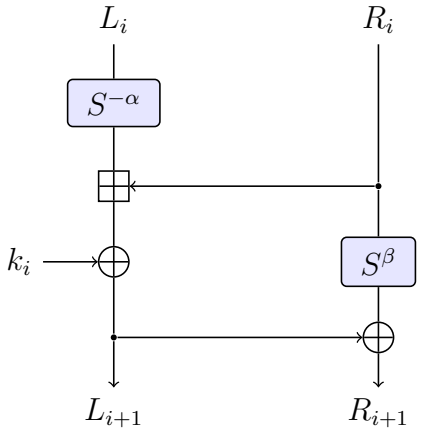

Figure 2: Round function of Speck

1.2.1 Key schedules

The SPECK key schedules use the round function to generate round keys. Let $K = (l_{m-2}, \ldots, l_0, k_0)$ be a key for a SPECK, where $l_i, k_0 \in \mathbb{F}_2^n$, $m \in \{2, 3, 4\}$. The value of m depends on the values of the block size $2n$ and the number of rounds T (Table 2). Keys k_i and l_i are defined as

$$l_{i+m-1} = (k_i + S^{-\alpha} l_i) \oplus i,$$

$$k_{i+1} = S^{\beta} k_i \oplus l_{i+m-1}$$

Block size $2n$	**Key size** mn	**Word size** n	**Key words** m	**Rot** α	**Rot** β	**Rounds** T
32	**64**	16	4	7	2	22
48	**72**	24	3	8	3	22
	96		4			23
64	**96**	32	3	8	3	26
	128		4			27
96	**96**	48	2	8	3	28
	144		3			29
128	**128**	64	2	8	3	32
	192		3			33
	256		4			34

Table 2: SPECK parameters

Conducting cryptanalysis on a small number of rounds (such as 3 and 4) with selecting standard characteristics (Table 2) is not sensible since the keys are not built on the basis of the original ones and there will be no connection between them. Therefore, in this work we consider the cipher with $m = 1$ for $T \in \{3, 4\}$.

1.2.2 Addition modulo 2^n

The round function of the Speck cipher is nonlinear, this property in SPECK is provided by the addition operation modulo 2^n, which is the part of the encryption algorithm. It is possible to obtain a redefined system of $6n - 3$ linearly independent algebraic equations that completely describe the operation under consideration [12]. One of them will be linear while the rest

are quadratic.

$$\begin{cases} w_0 x_{i+\alpha} = x_\alpha x_{i+\alpha} \oplus y_0 x_{i+\alpha} \ , \ i = \overline{1, n-1} \\ w_0 y_i = x_\alpha y_i \oplus y_0 y_i \ , \ i = \overline{0, n-1} \\ w_0 w_i = x_\alpha w_i \oplus y_0 w_i \ , \ i = \overline{0, n-1} \\ w_1 x_\alpha = x_{1+\alpha} x_\alpha \oplus y_1 x_\alpha \oplus x_\alpha y_0 \\ w_1 y_0 = x_{1+\alpha} y_0 \oplus y_1 y_0 \oplus x_\alpha y_0 \\ w_i = x_{i+\alpha} \oplus y_i \oplus x_{i-1+\alpha} \oplus y_{i-1} \oplus x_{i-1+\alpha} y_{i-1} \oplus x_{i-1+\alpha} w_{i-1} \oplus \\ y_{i-1} w_{i-1}, i = \overline{2, n-1} \\ w_i(x_{i-1+\alpha} \oplus y_{i-1}) = x_{i-1+\alpha} x_{i+\alpha} \oplus x_{i-1+\alpha} y_i \oplus x_{i-1+\alpha} \oplus x_{i-1+\alpha} w_{i-1} \oplus \\ x_{i+\alpha} y_{i-1} \oplus y_{i-1} y_i \oplus y_{i-1} \oplus y_{i-1} w_{i-1} \ , \ i = \overline{2, n-1} \\ w_i(x_{i-1+\alpha} + w_{i-1}) = x_{i-1+\alpha} x_{i+\alpha} \oplus x_{i-1+\alpha} y_i \oplus x_{i-1+\alpha} \oplus x_{i+\alpha} w_{i-1} \oplus y_i w_{i-1} \oplus \\ x_{i-1+\alpha} w_{i-1} \ , \ i = \overline{2, n-1} \\ w_0 = x_{\alpha \ mod \ n} \oplus y_0 \\ w_1 = x_{1+\alpha} \oplus y_1 \oplus x_\alpha y_0 \end{cases}$$

2 Attacks based on linearization

2.1 Pure linearization

The idea of this method is to assign every monomial from the initial system with a new variable. The system after the assignment becomes a linear one. The obtained system is solved via different methods, for instance Gaussian elimination, and solutions of linear system are checked for being solutions for the initial nonlinear system of equations.

The efficiency of linearization depends on the rank r of the system whereas the number of different monomials in the initial system defines the number of variables n' in the system of linear equations. The set of solutions is not empty, so it is $2^{n'-r} > 0$, hence in order to estimate the performance one should analyze the bounds for the values of n' and r.

The analysis of this attack (see [13]) shows that the rank of the system is expected to be sufficiently large if $m \approx n^2/2$. Estimation of the required number of operations and time complexity of the attack can be provided by taking into account the number of different monomials in the system of equations that describe the considered cipher and varying its rank.

2.1.1 Number of different monomials in Simon's system of equations

Considering the encryption algorithm, we can estimate the number of monomials for each round. With the introduction of new variables, the estimate is $6nT$, where n is the word length, T is the number of rounds. The estimate was obtained based on the fact that for each operation new variables are introduced (xor, plus, plus, addition with key) and taking into account the re-designation when replacing L_{i+1} and R_{i+1}.

In addition, an estimation of the number of variables without reassignment (introduction of new variables) was carried out in order to assess the effectiveness of the linearization method. When analyzing a small number of rounds without introducing new variables, it was noticed that every four rounds, the number of variables decreases when added using R_i. Thus, a recurrence relation was obtained for the number of variables, taking into account the decrease every four rounds. Let $P(T)$ be the number of variables on the T-th round, where n is a word size.

$$P(T) = \begin{cases} 4 \cdot n, & T = 1 \\ 7 \cdot n, & T = 2 \\ n\big(P^2(T-2) + P(T-2) + 1\big), & \text{if } T \text{ is divisible by 4} \\ n\big(P^2(T-2) + P(T-1) + P(T-2) + 1\big), & \text{otherwise.} \end{cases}$$

In practice, an estimate for the number of variables with $n = 16$, $T = 32$ was found, excluding the key stage, it is about 2^{68}. Thus, we found that changing the variables significantly reduces the amount of computation.

For the case when new variables are introduced on every round and the degree is at most 2, the formula without monomials that come from the key schedule equations (all equations are linear), is

$$M \leqslant 6nT,$$

where M is a number of monomials, n is a word size, T is a number of rounds.

Block size $2n$	**Word size** n	**Rounds** T	**Num. of monomials**	**Rank of the linearized system**	**Num. of unknowns with key schedule**
32	16	32	$\approx 2^{11.6}$	$\approx 2^{8.9}$	$\approx 2^{9}$
48	24	36	$\approx 2^{12.3}$	$\approx 2^{9.7}$	$\approx 2^{9.7}$
64	32	42	$\approx 2^{13}$	$\approx 2^{10.3}$	$\approx 2^{10.4}$
		44	$\approx 2^{13}$	$\approx 2^{10.4}$	$\approx 2^{10.4}$
96	48	52	$\approx 2^{13.9}$	$\approx 2^{11.2}$	$\approx 2^{11.3}$
		54	$\approx 2^{13.9}$	$\approx 2^{11.3}$	$\approx 2^{11.3}$
128	64	68	$\approx 2^{14.7}$	$\approx 2^{12}$	$\approx 2^{12.1}$
		69	$\approx 2^{14.7}$	$\approx 2^{12}$	$\approx 2^{12.1}$
		72	$\approx 2^{14.8}$	$\approx 2^{12.1}$	$\approx 2^{12.2}$

Table 3: Parameters of Simon's system of equations

By using these data the estimates of cryptographic strength with respect to linearization can be made. In fact it defined by the complexity of the search in the set of the solutions of obtained system of equations and the complexity of obtaining the solutions by Gauss elimination. The complexity of the search is about $2^{n'-r}$, where n' is the number of monomials (variables in lonearized system) and r is the rank of linearized system.

For real specifications, for instance for 32 rounds block size and 32 rounds number of monomials (variables in linearized system) is about $2^{11.5}$ while the rank is about 2^9. For 96 block size and 52 rounds the number of monomials (variables in linearized system) is about 2^{14} while the rank is about 2^{11} It is clear that the obtained estimates are unfeasible in comparison with brute force.

2.1.2 Number of different monomials in Speck's system of equations

The main method of withholding degree is the introduction of new variables for the output bits of nonlinear operations. In this case the degree will then not exceed 2. New variables are introduced with each new round: cipher text (x_i, y_i), key (k_i, l_i), variables describing addition modulo 2^n.

In the system of equations that describes an addition modulo 2^n (section 1.2.2), there are total $5(7n-8)$ monomials. In practice, it was found out that the unique monomials in the system of equations of addition modulo 2^n is at most $25n - 18$. As a result, the number of unique monomials per Speck round is at most $28n - 18$ per each round.

The final formula for estimating the number of monomials, excluding such ones that come from the key schedule equations (all equations are linear), is

$$M \leqslant (28n - 18)T,$$

where M is the number of monomials, n is the word size, T is the number of rounds.

Block size $2n$	Rounds T	Num. of monomials	Rank of the system without key sch.	Rank of the system with key schedule	Num. of unknowns without key sch.	Num. of unknowns with key schedule
32	22	$\approx 2^{13.2}$	$\approx 2^{11.4}$	$\approx 2^{12.4}$	$\approx 2^{9.95}$	$\approx 2^{11}$
48	22	$\approx 2^{13.8}$	$\approx 2^{12}$	$\approx 2^{13}$	$\approx 2^{10.5}$	$\approx 2^{11.6}$
	23	$\approx 2^{13.9}$	$\approx 2^{12.1}$	$\approx 2^{13.1}$	$\approx 2^{10.6}$	$\approx 2^{11.7}$
64	26	$\approx 2^{14.5}$	$\approx 2^{12.7}$	$\approx 2^{13.7}$	$\approx 2^{11.2}$	$\approx 2^{12.3}$
	27	$\approx 2^{14.5}$	$\approx 2^{12.8}$	$\approx 2^{13.7}$	$\approx 2^{11.3}$	$\approx 2^{12.3}$
96	28	$\approx 2^{15.2}$	$\approx 2^{13.4}$	$\approx 2^{14.4}$	$\approx 2^{11.9}$	$\approx 2^{13}$
	29	$\approx 2^{15.2}$	$\approx 2^{13.4}$	$\approx 2^{14.4}$	$\approx 2^{12}$	$\approx 2^{13}$
128	32	$\approx 2^{15.8}$	$\approx 2^{14}$	$\approx 2^{15}$	$\approx 2^{12.6}$	$\approx 2^{13.6}$
	33	$\approx 2^{15.8}$	$\approx 2^{14.1}$	$\approx 2^{15.1}$	$\approx 2^{12.6}$	$\approx 2^{13.6}$
	34	$\approx 2^{15.9}$	$\approx 2^{14.1}$	$\approx 2^{15.1}$	$\approx 2^{12.6}$	$\approx 2^{13.6}$

Table 4: Parameters of Speck's system of equations

As well as in the previous section by using obtained estimates the complexity of the linearization attack can be analyzed. For real specifications, for instance for 32 rounds block size and 32 rounds number of monomials (variables in linearized system) is about 2^{13} while the rank is about 2^9. For 96 block size and 28 rounds the number of monomials (variables in linearized system) is about 2^{15} while the rank is about 2^{13} It is clear that the obtained estimates are unfeasible in comparison with brute force as well.

2.2 XL-attack

This attack was introduced in [14, 15]. It takes a system of m polynomial equations in n unknowns, of degree d and outputs its solution or solutions, if the equations have sufficient rank.

- 1: Select degree $D > d$. Usually $D = d + 1$.
- 2: Make a list S of all monomials of degree $D - d$ or less, including the monomial 1, which has degree 0.
- 3: Multiply all equations by every element of S. (Since there were m equations before this step, there are $m|S|$ equations after it).
- 4: Linearize the system.
- 5: Solve the obtained system via linear algebra.

For the case $d = 2$ and $D = d + 1$ the analysis of this attack (see [13]) shows that the uique solution is likely to be found if $m \approx n^2/6$.

2.3 ElimLin

The ElimLin algorithm appeared in [16] (see also its analysis in [17]). Its point is the search of hidden linear equations existing in the ideal generated by the given system of equations. This algorithm is composed of two sequential steps:

1 : Gaussian Elimination: Discover all the linear equations in the linear span of initial equations.

2 : Substitution: Variables are iteratively eliminated.

In more details it can be described as follows.

INPUT: A system of degree 2 polynomial equations. OUTPUT: Either, a solution or solutions to the system, if the equations have sufficient rank, or if not, then a reduced system of equations in fewer variables than the original, to be solved by some other method.

1 : D is an empty set.

2 : Linearize the system of equations.

3 : Perform Gaussian Elimination to result in Reduced Row Echelon Form.

4 : Let ℓ be the number of all-linear equations found.

  1 : If $\ell = 0$, STOP.

  2 : If $\ell > 0$.

    1 : For $i = 1, 2, \ldots, \ell$

      1 : Move all the variables and constants, but one, to one side of the equal sign.

      2 : Substitute this redefinition of a variable into the other equations, thus eliminating one variable.

      2 : Substitute this redefinition of a variable into the other definitions in D.

      2 : Add the definition to D.

    2 : Goto Step 3, "Perform Gaussian Elimination."

2.4 Results

In the Table 5 we give the results for the pure linearization, the XL-method and the ElimLin method that allow to compare SIMON and SPECK from that perspective. For XL-method the value of the resulting degree D was chosen to 3.

A search on the key space key is 2^{16} (when $n = 16$, $m = 1$). As we can see in the table 5 the linearization method from round 4 and 5 onwards gives worse results than a brute force attack. Using the pure linearization method for T at least 4 and XL-method for at least 5 rounds (cipher SIMON) does not improve the search for a solution in comparison with brute force.

	Simon parameters	**Number of equations**	**Number of variables**	**Number of monomials**	**Number of solutions**
Pure linearization	$T = 3$, $m = 1$	48	32	48	4, only one corresponds to the key
XL-method	$T = 3$, $m = 1$	1584	32	992	1
Pure linearization	$T = 4$, $m = 1$	64	48	80	65536
XL-method	$T = 4$, $m = 1$	3136	48	2616	256, only one corresponds to the key
Pure linearization	$T = 5$, $m = 1$	80	64	112	2^{32}
XL-method	$T = 5$, $m = 1$	5200	64	5008	2^{336}
	Speck parameters				
Pure linearization	$T = 3$, $m = 1$	500	176	1236	—
XL-method	$T = 3$, $m = 1$	88500	176	185216	—

Table 5: Results for attaks based on linearization

	Parameters	**(Equations, Linear equations)**	**(Equations, Linear equations after ElimLin applied)**
Simon	$T = 3$, $m = 1$	(48, 32)	(48, 32)
Simon	$T = 5$, $m = 1$	(80, 32)	(80, 48)
Speck	$T = 3$, $m = 1$	(500, 132)	(307, 137)
Speck	$T = 5$, $m = 2$	(1032, 296)	(654, 297)

Table 6: Results for ElimLin

3 Attack based on SAT solvers

3.1 SAT

The Boolean satisfiability problem (SAT) is a decision problem, in which for an arbitrary Boolean formula the question is whether there exists such assignment of variables that the formula has value True. This problem is known to be NP-hard.

SAT solvers are a powerful computational tools to test the hardness of certain problems, they have successfully been used to test hardness assump-

tions [18]. There are several examples of the usage of SAT solvers in a scope of algebraic cryptanalysis. The first SAT-based cryptanalysis was provided by Massacci et al. in [19]. In that work the Data Encryption Standard (DES) was attacked with a usage of DPLL-based SAT solvers.

SAT-based cryptanalysis implies two stages: on the first stage a SAT encoding is provided, for instance the translation of the given ANF system to CNF. There are some tools for converting cryptographic tasks into CNF: Grain-of-Salt [20], URSA [21], SAW [22], Transalg [23], Bosphorus [24]. We use anf2cnf [25] convertor from PolyBoRi library integrated at Sage. On the second stage the obtained SAT instance is solved using SAT solving algorithm. For cryptographic systems often applied such SAT-solvers as CryptoMiniSat [26] and Lingeling (with its parallel versions Plingeling and Treengeling) [27].

For addition information about overview and state-of-art on SAT solvers and their applications to cryptanalysis we recommended to refer to paper [23].

3.2 Results

In this section the results on the usage of SAT solvers for attack on reduced-round versions of Simon and Speck ciphers are given. We apply SAT solvers CryptoMiniSat (in Sage ver. 6.10) and Lingeling, Plingeling, Treengeling at PC with following features: Core i5-4690 CPU 3.5 GHz (x4), 12Gb RAM.

Choosing the tools for solving SAT problem was make in favor of Lingeling family solvers and CryptoMiniSat based on rating SAT Competition 2018 [28], 2020 [29]. The CryptoMiniSat solver was originally developed for solving SAT problems related to cryptographic structures and has been widely used in scientific literature for analyzing methods based on SAT solving. Plingeling and CryptoMiniSat solvers were included in the top-3 parallel tracks (only for SAT) SAT Competition 2018, which is presumably about the effectiveness of their subsequent use on multiprocessor systems.

Experimental result of SAT solving for 3 to 10 round Simon and 3 to 6 round Speck are presented at Tables 7 and 8. Two ANF generation form for Simon were examined: all round keys are independent variables and all round keys are represented by key schedule algorithm.

Simon parameters	Num. of equations	Num. of unknowns	SAT parameters	SAT	Time (RAM)
$T = 3$, $m = 1$ (with round key)	80	80	96 lit., 432 clause	CryptoMiniSat (SageMath) Lingeling Plingeling Treengeling	0.17 sec. 0.01 sec., 0.1 MB 1.1 sec., 0.7 MB 0.50 sec., 0.05 MB
$T = 5$, $m = 2$ (with round key)	128	128	192 lit., 1136 clause	CryptoMiniSat (SageMath) Lingeling Plingeling Treengeling	8.43 sec. 0.9 sec., 2.0 MB 2.9 sec., 21.0 MB 2.36 sec., 10 MB
$T = 5$, $m = 2$ (key schedule)	80	80	176 lit., 1710 clause	CryptoMiniSat (SageMath) Lingeling Plingeling Treengeling	15.79 sec. 1.4 sec., 2.0 MB 2.2 sec., 15.4 MB 0.86 sec., 3 MB
$T = 7$, $m = 2$ (with round key)	192	192	320 lit., 2064 clause	CryptoMiniSat (SageMath) Lingeling Plingeling Treengeling	287.31 sec. 3687.9 sec., 45.9 MB 212.7 sec., 103.3 MB 681.14 sec., 77 MB
$T = 7$, $m = 2$ (key schedule)	112	112	320 lit., 3632 clause	CryptoMiniSat (SageMath) Lingeling Plingeling Treengeling	101.23 sec. 1867.2 sec., 38.0 MB 229.5 sec., 99.2 MB 389.84 sec., 62 MB
$T = 8$, $m = 2$ (with round key)	224	224	384 lit., 2528 clause	CryptoMiniSat (SageMath) Lingeling Plingeling Treengeling	- 69811.9 sec., 120.5 MB 4775.5 sec., 260.3 MB 12702.81 sec., 182 MB
$T = 8$, $m = 2$ (key schedule)	128	128	368 lit., 4448 clause	CryptoMiniSat (SageMath) Lingeling Plingeling Treengeling	51533.67 sec. 845.4 sec., 26.6 MB 1188.8 sec., 169.2 MB 4426.12 sec., 95 MB
$T = 9$, $m = 2$ (key schedule)	144	144	480 lit., 6448 clause	CryptoMiniSat (SageMath) Lingeling Plingeling Treengeling	- >260174.3 sec., >180.7 MB 47799.2 sec., 620.3 MB 24547.91 sec., 172 MB
$T = 10$, $m = 2$ (key schedule)	160	160	560 lit., 8096 clause	CryptoMiniSat (SageMath) Lingeling Plingeling Treengeling	- - 17554.9 sec., 458.8 MB 60776.91 sec., 234 MB
$T = 11$, $m = 2$ (key schedule)	176	176	640 lit., 9648 clause	CryptoMiniSat (SageMath) Lingeling Plingeling Treengeling	- -

Table 7: Results for SAT solvers on Simon

Speck parameters	Num. of equations	Num. of unknowns	SAT parameters	SAT	Time (RAM)
$T = 3$, $m = 1$	500	176	1460 lit., 11020 clause	CryptoMiniSat (SageMath) Plingeling Treengeling Lingeling	0.56 sec. 0.9 sec., 9.6 MB 0.97 sec., 4 MB 0.2 sec., 1.9 MB
$T = 4$, $m = 2$	782	320	2492 lit., 17380 clause	CryptoMiniSat (SageMath) Plingeling Treengeling Lingeling	21.4 sec. 3.0 sec., 17.3 MB 8.25 sec., 15 MB 61.4 sec., 14.8 MB
$T = 5$, $m = 2$	1032	416	3312 lit., 23184 clause	CryptoMiniSat (SageMath) Plingeling Treengeling Lingeling	- - 14448.17 sec., 278 MB -
$T = 6$, $m = 2$	1282	512	4132 lit., 28988 clause	CryptoMiniSat (SageMath) Plingeling Treengeling Lingeling	- - 123353.82 sec., 546 MB -

Table 8: Results for SAT solvers on Speck

4 The Raddum–Semaev Method

4.1 The representation of the system os equations

This approach to solving sparse polynomial systems of equations over $\mathbb{F}_2$ was introduced by Hårvard Raddum and Igor Semaev, its general description was presented in [30]. The analysis and some properties one can find in paper [31].

Its core is the following. To i-th equation $f_i(x) = 0$ from the initial system of equations a subset if variables $X_i \subseteq X$ and the list $L_i \subseteq \mathbb{F}_2^{|X_i|}$ of vectors are associated. The set X_i is the set of all variables from which the Boolen function f essentially depends. The list L_i consists of all configurations that are in fact solutions of the equation $f_i(x) = 0$ (it is expected that the cardinality of $|L_i|$ is about $2^{|X_i|-1}$). Every pair $\big(X_i, L_i\big)$ can be considered as a single vertex in a graph. This set of vertices is said to be upper set [13]. The other type of vertices (lower set) is defined by the pairs $\big(X_i \cap X_j, L'_{ij}\big)$ each of which is obtained via the intersection of variables ffrom i-th and j-th equations, whereas the list L'_{ij} is a set of all possible combinations for tha variables from $X_i \cap X_j$ that is a space $\mathbb{F}_2^{|X_i \cap X_j|}$. The edges are drawn from the vertex $\big(X_i \cap X_j, L'_{ij}\big)$ to every of vertices $\big(X_i, L_i\big)$ and $\big(X_j, L_j\big)$. If there is a pair of vertices with the same intersection that is already considered in the graph, two edges are added instead of introducing the new vertex.

The sparsity in variables plays an important role since the lists L_{ij} comprise all possible combinations from the intersections of two particular equatuions. Here, we discuss a form of sparsity when only a limited numberof variables actually appear in each equation. If this number is large the computationsl cost can be nonfeasible. Together with that, all solutions of the equations from the initial system should be considered.

4.2 Agreeing-Gluing Algorithm

The processing and the search of the solution is performed via the so called Agreeing procedure. This routine takes two adjacent vertices and updates their lists by removing vectors that have different subvectors for common variables. It starts chain-reaction with another vertices that were agreed before such update, so the algorithm proceeds them again that leads to the reducing of their lists.

In practise it is often the case when all vertices are in agreement state while there are still a lot of redundant configarations in their lists, that makes the search of the solution hard from this point. For such situations a Gluing

procedure is performed. For two pairs $\big(X_1, L_1\big)$ and $\big(X_2, L_2\big)$ two sets of variables $Z = X_1 \cup X_2$ and $Y = X_1 \cap X_2$ are defined by the rule $U = \{a_1, b, a_2\}$ with $(a_1, b) \in L_1$, $(b, a_2) \in L_2$, $a_i = X_i \setminus Y$ and b belongs to Y. Then the vector (a_1, b, a_2) is the gluing of (a_1, b) and $\{b, a_2\}$. After the gluing the new vertex is not agreed with its neighbours so the Agreement procedure can start.

There is also another technique used for re-starting the Agreement procedure that is known as Splitting. Its idea is that the list of the vertex is splitted into two parts one of which is temporarily discarded. If there is no solution at the end of the work of the Algorithm, the another partition is considered.

The criteria for stop is the situation when there is an only one item in every list, but in practise it is enough to have small number of vectors in the lists after the Agreeing-Gluing Algorithm.

As results for the usage of this Algorithm to attack SIMON and SPECK we give only maximal number of rounds for which the Algorithm finished in feasible time. It is worth mention that time complexity depends heavily on the heuristics used to start the Agreement process whether it is (partial) Splitting or Gluing. The choice of vertices for Gluing can also comprise some analysis of current state of the graph.

4.3 SIMON

The number of variables in each equation non-trivially depends on the number of rounds and keys. The Table 9 shows the dependence of the maximal number of variables in the equation on the number of rounds and keys.

$m = 2$	**Num. of rounds (T)**	5	6	7	8	9	10	11	12
	Max. num. of variables	9	11	12	12	14	17	18	18
	Num. of rounds (T)	13	14	15	16	17	18	19	20
	Max. num. of variables	18	18	18	18	22	25	26	26
$m = 4$	**Num. of rounds (T)**	5	6	7	8	9	10	11	12
	Max. num. of variables	6	10	18	21	23	25	28	31

Table 9: Number of variables for each equation for SIMON

It shows that for $T \geqslant 16$ it becomes rather costly to perform the Agreeng-Gluing algorithm.

Within current work the Agreeing-Gluing Algorithm was run on SIMON for up to 9 rounds.

Simon parameters	**Num. of equations**	**Num. of unknowns**	**Upper set Lower set**
$T = 7$, $m = 2$	112	112	112 800
$T = 8$, $m = 2$	128	128	128 1072
$T = 9$, $m = 2$	144	144	144 1600

Table 10: Parameters for the Raddum-Semaev Algorithm on Simon

4.4 Speck

By introducing of new variables on each round of Speck cipher, the number of different variables on each round does not increase. The maximum number of variables that occur in a single equation is 6. Furthermore, the number of equations and the number of variables on each round can be represented as a Table 11 for $m = 1$ and as a Table 12 for $m = 2, 3, 4$.

Number of variables	**Number of equations**
6	$2(T-1)(2n-4)$
5	$2(T-1)n$
4	$6(T-1)n + (T-2)n$
3	$2(n+1)(T-1)$
2	$3n$

Table 11: Number of variables for each equation for Speck, $m = 1$

Number of variables	**Number of equations**
6	$2(T-1)(2n-4)$
5	$2(T-1)n$
4	$6(T-1)n + (T-2)n$
3	$2(n+1)(T-1)$
2	$(T-1)n + 3n$

Table 12: Number of variables for each equation for Speck, $m = 2, 3, 4$

The Agreeing-Gluing Algorithm was run on Speck for up to 6 rounds.

Speck parameters	**Num. of equations**	**Num. of unknowns**	**Upper set Lower set**
$T = 3$, $m = 1$	500	176	500 558
$T = 4$, $m = 2$	782	320	782 749
$T = 5$, $m = 2$	1032	416	1032 1005
$T = 6$, $m = 2$	1282	512	1282 1229

Table 13: Parameters for the Raddum-Semaev Algorithm on Speck

5 Conclusion

The goal of current work was to analyze and compare the efficiency of different types of algebraic attacks under the same conditions on two instances of LRX- and ARX- ciphers that are based on the explicit usage of logical operations. This is the first attempt to estimate the resilience of the cipher Speck to algebraic cryptanalysis via different methods.

Experimental results show that algebraic analysis techniques is perspective way for modern cipher's robustness analysis (especially for lightweight ciphers). Two approaches at algebraic analysis as linearization methods and reduction to SAT-problem for Simon and Speck ciphers are presented. The usage of the the Raddum–Semaev Algorithm was also analyzed.

The results of algebraic analysis show that including of extra nonlinear operation (like addition modulo 2^n) leads to an extremely increase of time and memory complexity of algebraic attack. Therefore observed methods more efficiently applicable for Simon cryptanalysis then for Speck encryption algorithm. At the same time the sparsity of the system for Speck seems to be extremely lower than for Simon that leads to the idea that the usage of techniques that exploit sparsity is a goal worth pursuing.

Further directions of research are: theoretical complexity assessments of algebraic analysis for full-round Simon and Speck ciphers, experimental usage of other ANF-to-CNF converters and efficient SAT-solvers, observe and develop methods to combine linearization and SAT techniques to improve efficiency of analysis. The usage and comparison of other methods of solving systems of Boolean equations is also a direction for the future research.

Acknowledgments. The work is supported by the Mathematical Center

in Akademgorodok under agreement No. 075-15-2019-1613 with the Ministry of Science and Higher Education of the Russian Federation and Laboratory of Cryptography JetBrains Research.

The authors cordially thank Sergey Agievich and Natalia Tokareva for the advice and the attention to work and also the anonymous referees for their valuable comments and suggestions which led to the improvement of this paper.

References

[1] Hong D. et al., "HIGHT: A New Block Cipher Suitable for Low-Resource Device", *Lecture Notes in Computer Science*, CHES 2006: Cryptographic Hardware and Embedded Systems — CHES 2006, **4249**, 2006, 46–59.

[2] De Canniére C., Dunkelman O., Knezevic M., "KATAN and KTANTAN — A Family of Small and Efficient Hardware-Oriented Block Ciphers", *Lecture Notes in Computer Science*, CHES 2009: Cryptographic Hardware and Embedded Systems — CHES 2009, **5747**, 2009, 272–288.

[3] Gong Z., Nikova S., Law Y.W., "KLEIN: A New Family of Lightweight Block Ciphers", *Lecture Notes in Computer Science*, RFIDSec 2011: RFID. Security and Privacy, **7055**, 2012, 1–18.

[4] Shibutani K., Isobe T., Hiwatari H., Mitsuda A., Akishita T., Shirai T., "*Piccolo*: An Ultra-Lightweight Blockcipher", *Lecture Notes in Computer Science*, CHES 2011: Cryptographic Hardware and Embedded Systems — CHES 2011, **6917**, 2011, 342–357.

[5] Bogdanov A. et al., "PRESENT: An Ultra-Lightweight Block Cipher", *Lecture Notes in Computer Science*, CHES 2007: Cryptographic Hardware and Embedded Systems — CHES 2007, **4727**, 2007, 450–466.

[6] Abed F., List E., Lucks S., Wenzel J., "Differential Cryptanalysis of Round-Reduced Simon and Speck", *Lecture Notes in Computer Science*, FSE 2014: Fast Software Encryption, **8540**, 2015, 525–545.

[7] Raddum H., "Algebraic Analysis of the Simon Block Cipher Family", *Lecture Notes in Computer Science*, LATINCRYPT 2015: Progress in Cryptology — LATINCRYPT 2015, **9230**, 2015, 157–169.

[8] Courtois N, Mourouzis T, Song G, Sepehrdad P, Susil P., "Combined Algebraic and Truncated Differential Cryptanalysis on Reduced-round Simon", 11th International Conference on Security and Cryptography, 2014, 399–404.

[9] Andrzejczak M., Dudzic W., "SAT Attacks on ARX Ciphers with Automated Equations Generation", *Infocommunications Journal*, **11**:4 (2019), 2–7.

[10] Yeo S.L., Le D.-P., Khoo K., "Improved algebraic attacks on lightweight block ciphers", *Jour. of Cryptogr. Engineering*, **11** (2021), 1–19.

[11] Beaulieu R., Shors D., Smith J., Treatman-Clark S., Weeks B., Wingers L., "The Simon and Speck Families of Lightweight Block Ciphers", *NSA Research Directorate*, 2013.

[12] Courtois N. T.,Debraize B., "Algebraic Description and Simultaneous Linear Approximations of Addition in Snow 2.0", *Lecture Notes in Computer Science*, ICICS 2008: Information and Communications Security, **5308**, 2008, 328–344.

[13] Bard G., *Algebraic Cryptanalysis*, Springer, 2009, 356 pp.

[14] Courtois N., Shamir A., Patarin J., Klimov A., "Efficient algorithms for solving overdefined systems of multivariate polynomial equations", *Lecture Notes in Computer Science*, EUROCRYPT 2000, **1807**, ed. B. Preneel, 2000, 392–407

[15] Courtois N., "The security of cryptographic primitives based on multivariate algebraic problems", MQ, MinRank, IP, HFE. Ph.D. thesis, Paris VI (2001). Available at http://www.nicolascourtois.net/phd.pdf.

[16] Courtois, N., Bard, G.V., "Algebraic cryptanalysis of the data encryption standard", *Lecture Notes in Computer Science*, IMA International Conference on Cryptography and Coding Theory, **4887**, ed. S.D. Galbraith, 2007, 152–169.

[17] Courtois N., Sepehrdad P., Susil P., Vaudenay S., "Elimlin algorithm revisited", *Lecture Notes in Computer Science*, FSE 2012: Fast Software Encryption, **7549**, 2012, 306–325.

[18] Soos M., Nohl K., Castelluccia C., "Extending SAT Solvers to Cryptographic Problems", *Lecture Notes in Computer Science*, SAT 2009: Theory and Applications of Satisfiability Testing — SAT 2009], **5584**, 2009, 244–257.

[19] Massacci F., Marraro L., "Logical cryptanalysis as a SAT-problem: Encoding and analysis", *Journal of Automated Reasoning*, **24** (2000), 165–203.

[20] Grain of Salt https://www.msoos.org/grain-of-salt/.

[21] Janicic P., "URSA: A System for Uniform Reduction to SAT", *Logical Methods in Computer Science*, **8** (2012).

[22] Carter K., Foltzer A., Hendrix J., Huffman B., Tomb A., "SAW: The software analysis workbench", 2013 ACM SIGAda Annual Conference on High Integrity Language Technology (HILT), 2013, 15–18.

[23] Semenov A., Otpuschennikov I., Gribanova I., Zaikin O., Kochemazov S., "Translation of Algorithmic Descriptions of Discrete Functions to SAT with Applications to Cryptanalysis Problems", *Logical Methods in Computer Science*, **16**:1 (2020).

[24] Choo D., Soos M., Chai K.M.A., Meel K. S., "Bosphorus: Bridging ANF and CNF Solvers", 2019 Design, Automation, Test in Europe Conference Exhibition (DATE), 2019.

[25] "An ANF to CNF Converter using a Dense/Sparse Strategy", https://doc.sagemath.org/html/en/reference/sat/sage/sat/converters/polybori.html.

[26] Soos M., "The CryptoMiniSat 5 set of solvers at SAT competition 2016", *SAT Competition 2016 — Solver and Benchmark Descriptions*, 2016.

[27] Biere A., "CaDiCaL, Lingeling, Plingeling, Treengeling, YalSAT Entering the SAT Competition 2017", Proceedings of SAT Competition 2017 — Solver and Benchmark Descriptions, **B-2017-1 of Department of Computer Science Series of Publications B** (2017), 14–15.

[28] Heule M.J.H., Järvisalo M.J., Suda M. (eds.), "Proceedings of SAT Competition 2018 — Solver and Benchmark Descriptions", **B-2018-1** (2018).

[29] "Proceedings of SAT Competition 2020 — Solver and Benchmark Descriptions", **B-2020-1** (2020).

[30] Raddum H., Semaev I., "New technique for solving sparse equation systems", Cryptology ePrint Archive, 2006/475.

[31] Semaev I., "On solving sparse algebraic equations over finite fields", *Des. Codes Cryptogr.*, **49**:1–3 (2008), 47–60.

Algebraic and Probabilistic Aspects

Two Variants of Lempel-Ziv Criterion and Their Reasoning

Vladimir Mikhailov and Vasilii Kruglov

Steklov Mathematical Institute of the Russian Academy of Sciences, Russia
zpt@rambler.ru, mikh_vg@mail.ru

Abstract

Consider a random binary sequence $X_1, \ldots, X_n$ and the hypothesis H_0 that the elements of this sequence are independent and have equiprobable distributions on the set $\{0, 1\}$. In this paper we propose two goodness-of-fit criterions for the hypothesis H_0, these criterions are based on computation of Lempel-Ziv statistics. A sequence of the length $n = mT$ is divided into m blocks of (equal) length T, for these blocks we compute values of Lempel-Ziv statistics $W_1(T), \ldots, W_m(T)$. If the hypothesis H_0 is true, these values are independent and their distributions are equal, so we may construct goodness-of-fit tests for hypothesis H_0 based on these statistics via standard methods.

The first criterion is based on the statistic $\tilde{W}(2mT) = (W_1 + W_2 + \ldots + W_m) - (W_{m+1} + W_{m+2} + \ldots + W_{2m})$, the distribution of this statistic is symmetric about zero.

The statistic of the second criterion is the value $\tilde{\chi}^2(mT) = \max_{1 \le k \le m} \chi^2_k(T)$, where $\chi^2_1(T), \ldots, \chi^2_m(T)$ — values of chi-square statistics corresponding to $W_1(T), \ldots, W_m(T)$.

For both criterions we propose limit distributions of statistics, and for the first criterion we also obtain an estimation for the speed of convergence to the limit normal distribution,

To compute the distributions of statistics $W_k(T)$ we use the formulae proposed in [4] (these formulae may be found in the section 5).

Keywords: Lempel-Ziv, RNG testing, statistical criterion, computation.

1 Introduction

Lempel-Ziv critetion for a long time was a part of the NIST Statistical Test Suite — well-known collection of statistical criterions for testing the quality of random and pseudorandom sequences designed by the National Institute of Standards and Technology, USA (ref. [1], [2]).

For computation of Lempel-Ziv statistic a sequence $X_1, X_2, \ldots$ of elements of alphabet $\{0, 1\}$ is divided into sequences of digits (words) in such a way that any next word is the least word that is not equal to any of previous

words; the first word is the empty word. The statistic of Lempel-Ziv criterion is the amount $W(T)$ of words obtained in such a way for the sequence $X = (X_1, X_2, \ldots, X_T)$ of the length T.

Examples:

Binary sequence 011101101011 of 12 digits is be divided into 7 words $\varnothing, (0), (1), (11), (01), (10), (101)$ and remainder 1 that is not considered because it is equal to the third word.

Binary sequence 010101101010 of 12 digits is be divided into 7 words $\varnothing, (0), (1), (01), (011), (010), (10)$ without remainder.

The Lempel-Ziv criterion is a goodness-of-fit criterion for the hypothesis H_0 that digits of sequence X are independent and equiprobable distributed on $\{0, 1\}$, this criterion is defined as follows:

$$\{|W(T) - \mu(T)| < C\sigma(T)\} \Rightarrow H_0,$$
$$\{|W(T) - \mu(T)| \geq C\sigma(T)\} \Rightarrow H_1, \tag{1}$$

where C — critical level, H_1 — full alternative for hypothesis H_0, $\mu(T) = \mathbf{E}W(T)$ and $\sigma(T) = \sqrt{\mathbf{D}W(T)}$. It was proved (ref. [1], §3.10) that

$$\lim_{T\to\infty} \frac{\mathbf{E}W(T)}{T/\log_2 T} = 1$$

$$\mathbf{D}W(T) \sim \frac{T(C_D + \delta\log_2(T))}{\log_2^3 T} \text{ for } T \to \infty, \tag{2}$$

where $C_D = 0.26600\ldots$ is a constant and $\delta(\cdot)$ is a slowly varying function with zero mean value, $|\delta(\cdot)| < 10^{-6}$.

As was mentioned in [1], if the hypothesis H_0 is true, the distribution of random variable $(W(T) - \mu(T))/\sigma(T)$ for $T \to \infty$ converges to the standard normal distribution, so for a given significance level $\alpha > 0$ the critical level C is generally defined by the condition $2(1-\Phi(C)) = \alpha$, where $\Phi(x)$ — standard normal distribution function, and the probability to reject the hypothesis H_0 if it is true tends to α for $T \to \infty$.

Due to insufficient knowledge about speed of convergence of the distribution of statistic $W(T)$ to the limit distribution and about accuracy of estimations (2), it was recommended to use Lempel-Ziv criterion only for long binary sequences, i.e. for $T \geq 10^6$ (ref. [1]). This shortcoming was mentioned as one of reasons for removing this criterion from NIST Suite (ref. [3]).

The narrowness of the interval that contains the main mass of distribution of statistic $W(T)$ was also mentioned as a shortcoming of Lempel-Ziv criterion (ref. [3]).

Obviously, there may be some other compression-based statistical criterions. As an example, some interesting experimental results for criterions based on RAR and ARJ compression, along with theoretical results for appliance of universal codes for statistical criterions for random sequences of 0 and 1, may be found in [7].

As an advantages of criterions presented in this paper over standard Lempel-Ziv criterion we may mention their accuracy that is limited only by computational resources. Further, we propose limit distributions for statistics of these criterions and for the first statistic we have obtained explicit estimations for the speed of convergence to limit (normal) distribution. For reasoning of these criterions we significantly rely on numeric data obtained via the method of computation of probabilities of distribution of $W(T)$ proposed in [4] for the assumption that hypothesis H_0 is true.

2 Criterion with summation

Divide sample $X = (X_1, X_2, \ldots, X_{2mT})$ into $2m$ nonintersecting blocks of the length T and for each of these blocks compute value of Lempel-Ziv statistic $W(T)$. Denote computed values as $W_1, W_2, \ldots, W_{2m}$ and compute statistic

$$\tilde{W}(2mT) = (W_1 + W_2 + \ldots + W_m) - (W_{m+1} + W_{m+2} + \ldots + W_{2m}).$$

Formula for this statistic may be rewritten as

$$\tilde{W}(2mT) = \sum_{i=1}^{m} V_i(2T) = \sum_{i=1}^{m} (W_i(T) - W_{i+m}(T)),$$

in this way random value $\tilde{W}(2mT)$ is represented as the sum of m independent values $V_i(2T)$ that are identically distributed and

$$\mathbf{E}V_i(2T) = 0, \mathbf{E}\tilde{W}(2mT) = 0, \mathbf{D}\tilde{W}(2mT) = 2m\mathbf{D}W(T).$$

Goodness-of-fit criterion for hypothesis H_0 of independent and equiprobable distribution of the elements of sample X that is based on statistic $\tilde{W}(2mT)$ is defined as follows:

$$\left\{ |\tilde{W}(2mT)| < C\sqrt{\mathbf{D}\tilde{W}(2mT)\}} \right\} \Rightarrow H_0,$$

$$\left\{ |\tilde{W}(2mT)| \geq C\sqrt{\mathbf{D}\tilde{W}(2mT)} \right\} \Rightarrow H_1.$$

Hypothesis H_1 is the full alternative to hypothesis H_0, C is the critical level and for a given significance level $\alpha > 0$ critical level C is defined by the equality $2(1 - \Phi(C)) = \alpha$.

For given values of m and T distribution of $W_i(T)$ may be computed by formulae from section 5 and after that distribution of $V_i(2T)$ may be computed by formula

$$\mathbf{P}\{V_i(2T) = k\} = \sum_l \mathbf{P}\{W(T) = l\}\mathbf{P}\{W(T) = k - l\}.$$

Distribution of random variable $\tilde{W}(2mT)$ may be computed as the m-fold convolution of the distribution $V_i(2T)$.

For random variable V_{2T} and $T = 1000, 2000, \ldots, 8000$ we have computed values $\mathbf{E}|V_{2T}|$, $\mathbf{D}V_{2T}$, $\sigma(V_{2T}) = \sqrt{\mathbf{D}V_{2T}}$ and $\mathbf{E}|V_{2T}|^3$, these values are presented in table 3.

3 On the accuracy of normal approximation for $\tilde{W}(2mT)$.

The accuracy of normal approximation of the distribution of statistic $\tilde{W}(2mT)$ may be estimated via well-known Berry–Esseen inequality (inequality for constant C_1 may be found in [5]).

Theorem 1. *For distribution function of random variable $\tilde{W}(2mT)$ the following inequality is valid:*

$$\sup_{-\infty<x<\infty} \left| \mathbf{P}\left\{ \frac{\tilde{W}(2mT)}{\sqrt{\mathbf{D}\tilde{W}(2mT)}} < x \right\} - \Phi(x) \right| \le \frac{C_1 \mathbf{E}|V_1(2T)|^3}{(2m\mathbf{D}W(T)))^{3/2}}, \qquad (4)$$

where $C_1 \le 0.4774$.

Corollary 1. *If $m \to \infty$, then for any $-\infty < x < \infty$*

$$\mathbf{P}\left\{ \frac{\tilde{W}(2mT)}{\sqrt{\mathbf{D}\tilde{W}(2mT)}} < x \right\} \to \Phi(x).$$

In the final part of this paper in table 4 we present computed values of the right part of inequality (4) for $T = 1000, \ldots, 8000$ and $m = 1000, 2000$. These values show that for estimation of accuracy of normal approximation of $\tilde{W}(2mT)$ values of m are more important than values of T.

4 Criterion of chi-square type

Let us remind well-known fact of mathematical statistics (e.g. [6], §3.2, section 2).

Statement 1. *Let the support of random variable ξ be divided into intervals $\Delta_1, \ldots, \Delta_N$. Let simple hypothesis H_0 for the distribution of ξ be considered and $\mathbf{P}\{\xi \in \Delta_j\} = p_j^0$, $j = 1, \ldots, N$, if this hypothesis is true. For given sample $X_1, \ldots, X_n$ of values of random variable ξ and any $j = 1, \ldots, N$ compute values $v_j = \sum_{i=1}^{n} I_{\{X_i \in \Delta_j\}}$ and value*

$$\chi^2 = \sum_{j=1}^{N} \frac{(v_j - np_j^0)^2}{np_j^0}.$$

Then for $n \to \infty$ the distribution of random value χ^2 converges to chi-square distribution with $N-1$ degrees of freedom.

Denote the distribution function of chi-square distribution with $N-1$ degrees of freedom as $\chi^2_{N-1}(x)$. For a given significance level $\alpha \in (0,1)$ define critical level $C(N-1,\alpha)$ by equality

$$\chi^2_{N-1}(C(N-1,\alpha)) = \alpha. \tag{5}$$

For chi-square criterion the hypothesis H_0 is accepted if

$$\{\chi^2 \le C(N-1,\alpha)\}$$

and is rejected if

$$\{\chi^2 > C(N-1,\alpha)\}.$$

Now we propose goodness-of-fit criterion for the hypothesis of equiprobable Bernoulli distribution that is based on dividing samples into blocks and using chi-square criterion. For clearness of explaination we use explicit values of probabilities of distribution $W(T)$ for $T = 1000$ (ref. table 1).

Consider sample $X_1, \ldots, X_n$ of $n = mT$ digits, each digit is equal to zero or one. We divide this sample into m nonintersecting blocks of the length T and for each block compute value W_T. For $T = 1000$ we divide the set of possible values of $W(T)$ into $N = 5$ intervals

$$\Delta_1 = \{0, \ldots, 171\}, \Delta_2 = \{172\}, \Delta_3 = \{173\}, \Delta_4 = \{174\}, \Delta_5 = \{175, 176, \ldots\},$$

so, according to previously computed distribution of $W(T)$,

$$p_1^0 = 0.074603, p_2^0 = 0.245722, p_3^0 = 0.409858, p_4^0 = 0.236133, p_5^0 = 0.0336848.$$

Remark 1. *Probabilities $p_1^0, \ldots, p_5^0$ are given with the accuracy up to 6-th digit and the sum of these five values is equal to 1.0000008. During the computation these values were calculated with higher accuracy and for calculated values*

$$1 - p_1^0 - p_2^0 - p_3^0 - p_4^0 - p_5^0 = 2.64274 \cdot 10^{-019}.$$

For any of m obtained values $W_1(T), \ldots, W_m(T)$ we compute values $v_{k,1}(T), v_{k,2}(T), v_{k,3}(T), v_{k,4}(T), v_{k,5}(T)$ which are equal to amounts of values $W_k(T)$ that turned out to be in one of corresponding five intervals. We compute statistics

$$\chi_k^2(T) = \sum_{j=1}^{5} \frac{(v_{k,j} - np_j^0)^2}{np_j^0}$$

and statistic

$$\tilde{\chi}^2(mT) = \max_{1 \le k \le m} \chi_k^2(T).$$

For a given significance level $\alpha > 0$ we calculate the quantile $C(4, \alpha^{1/m})$ and define the criterion by the rules

$$\{\tilde{\chi}^2(mT) < C(N-1, \alpha^{1/m})\} \Rightarrow H_0,$$
$$\{\tilde{\chi}^2(mT) \ge C(N-1, \alpha^{1/m})\} \Rightarrow H_1,$$

where H_1 is the full alternative for main hypothesis H_0, $N = 5$.

Theorem 2. *Let the hypothesis H_0 be true, so random variables $X_1, \ldots, X_{mT}$ are independent and equiprobably distributed on $\{0, 1\}$. If parameters m and N are fixed and $T \to \infty$, then*

$$\mathbf{P}\{\tilde{\chi}^2(mT) < x\} \to 1 - \left(1 - \chi_{N-1}^2(x)\right)^m, x \in (-\infty, +\infty) \tag{6}$$

$$\mathbf{P}\{\tilde{\chi}^2(mT) \ge C(N-1, \alpha^{1/m})\} \to \alpha. \tag{7}$$

There $\chi_{N-1}^2(x)$ is the distribution function of the chi-square distribution with $N-1$ degrees of freedom and $C(N-1, \alpha)$ is the function of α defined in (5).

So, if the hypothesis H_0 is true and the size T of sample increases, then the probability to reject tends to α.

If value T increases, then the number of values such that random variable $W(T)$ is equal to this value with significant probability also increases. For example, for $T = 8000$ the set of possible values of $W(T)$ may be divided (ref. table 1) into $N = 7$ intervals

$$\Delta_1 = \{0, \ldots, 970\}, \Delta_2 = \{971\}, \Delta_3 = \{972\}, \ldots, \Delta_7 = \{976, 977, \ldots\},$$

so the distribution of statistic χ^2 converges to chi-square distribution with 6 degrees of freedom.

5 On computation of the probabilities of distribution $W(T)$

Lempel-Ziv algorithm sequentially compose a dictionary of words of 0 and 1. Consider value $S(n)$ that is equal to cumulative length of all words in a dictionary of n words.

Values of random variable $S(n)$ and statistic $W(T)$ are linked by equality

$$\big\{W(T) < n\big\} = \big\{S(n) > T\big\}.$$

Via this simple formula the computation of distribution of $W(T)$ may be implemented by significantly simpler computation of distribution of $S(n)$. Formulae for distributions of $S(n)$ are presented in the next theorem.

Theorem 3. *Let $X_1, X_2, \ldots$ be a sequence of independent random variables that are distributed on the set $\{0, 1\}$ with probabilities*

$$\mathbf{P}\{X_t = 1\} = p, \quad \mathbf{P}\{X_t = 0\} = 1 - p,$$

where $p \in (0, 1)$. Then for $r = 0, 1, \ldots, n(n-1)/2$

$$\mathbf{P}\{S(n+1) = n + r\} =$$

$$= \sum_{m=0}^{n} C_n^m p^m (1-p)^{n-m} \sum_{l=0}^{r} \mathbf{P}\{S(m) = l\}\mathbf{P}\{S(n-m) = r - l\}.$$

Initial and boundary values of probabilities $\mathbf{P}\{S(n+1) = r\}$ are defined by equalities

$$\mathbf{P}\{S(0) = 0\} = \mathbf{P}\{S(1) = 0\} = 1,$$

$$\mathbf{P}\{S(2) = 1\} = 1, \quad \mathbf{P}\{S(2) = 1 + r\} = 0, \text{ если } r \geq 1.$$

$$\mathbf{P}\{S(n+1) = r\} = 0, \quad r = 0, \ldots, n-1.$$

Proofs of these statements were given in [4].

6 Tables

All tables are provided for equiprobable distribution of random variables $\{X_1, \ldots, X_n\}$ on set $\{0, 1\}$.

Table 1 (in 4 parts). Probabilities of distributions of random variables $W(1000), \ldots, W(8000)$.

n	$T = 1000$	n	$T = 2000$
169	0.0007	300	0.0012
170	0.0089	301	0.0103
171	0.0648	302	0.0564
172	0.2457	303	0.1848
173	0.4098	304	0.3317
174	0.2361	305	0.2915
175	0.0330	306	0.1088
176	0.0006	307	0.0143
		308	0.0005
	8		9
$\mathbf{E}W(T)$	172.899	$\mathbf{E}W(T)$	304.220

n	$T = 3000$	n	$T = 4000$
420	0.0001	536	0.0005
421	0.0011	537	0.0036
422	0.0081	538	0.0193
423	0.0406	539	0.0710
424	0.1321	540	0.1747
425	0.2647	541	0.2753
426	0.3050	542	0.2633
427	0.1863	543	0.1439
428	0.0545	544	0.0417
429	0.0067	545	0.0058
430	0.0003	546	0.0003
	11		11
$\mathbf{E}W(T)$	425.627	$\mathbf{E}W(T)$	541.309

n	$T = 5000$	n	$T = 6000$
647	0.0001	756	0.0005
648	0.0014	757	0.0031
649	0.0081	758	0.0146
650	0.0338	759	0.0503
651	0.0996	760	0.1246
652	0.2013	761	0.2168
653	0.2693	762	0.2568
654	0.2285	763	0.1999
655	0.1172	764	0.0983
656	0.0343	765	0.0291
657	0.0053	766	0.0049
658	0.0004	767	0.0004
	12		12
$\mathbf{E}W(T)$	653.046	$\mathbf{E}W(T)$	761.811

n	$T = 7000$	n	$T = 8000$
862	0.0004	966	0.0002
863	0.0021	967	0.0013
864	0.0102	968	0.0065
865	0.0358	969	0.0239
866	0.0939	970	0.0668
867	0.1787	971	0.1395
868	0.2414	972	0.2137
869	0.2248	973	0.2344
870	0.1400	974	0.1796
871	0.0563	975	0.0935
872	0.0140	976	0.0321
873	0.0021	977	0.0070
874	0.0002	978	0.0009
	13		13
$\mathbf{E}W(T)$	868.213	$\mathbf{E}W(T)$	972.665

Probabilities that are not presented in table 1 are smaller than 0.0001. The string under the strings with expectations shows the number of such n that $\mathbf{P}\{W(T) = n\} \geq 0.0001$.

In table 2 we present computed values of $\mathbf{E}W(T)$ and $\mathbf{D}W(T)$ to compare them and main parts of formulae (2) recomended by NIST. We may note that for presented values of parameter T computed values considerably differ from

the main parts of values that are recommended by NIST.

Table 2.

T	$\mathbf{E}W_T$	$\frac{T}{\log_2 T}$	$\mathbf{D}W_T$	$\frac{0.266T}{\log_2^3 T}$
1000	172.899	100.343	0.96268	0.26874
2000	304.220	182.385	1.34154	0.40345
3000	425.627	259.723	1.65301	0.51781
4000	541.309	334.286	1.92859	0.62103
5000	653.046	406.910	2.18096	0.71686
6000	761.811	478.059	2.41656	0.80727
7000	868.213	548.025	2.63918	0.89348
8000	972.665	617.008	2.85136	0.97628

Table 3. Values $\mathbf{E}|V_{2T}|$, $\mathbf{D}V_{2T}$, $\sigma(V_{2T}) = \sqrt{\mathbf{D}V_{2T}}$ and $\mathbf{E}|V_{2T}|^3$ for $T = 1000, 2000, \ldots, 8000$

T	$\mathbf{E}\lvert V_{2T}\rvert$	$\mathbf{D}V_{2T}$	$\sigma(V_{2T})$	$\mathbf{E}\lvert V_{2T}\rvert^3$
1000	1.05493	1.92537	1.38758	4.29894
2000	1.26349	2.68309	1.63801	7.04866
3000	1.41194	3.30601	1.81824	9.62922
4000	1.53127	3.85718	1.96397	12.1292
5000	1.63290	4.36191	2.08852	14.5852
6000	1.72236	4.83312	2.19844	17.0151
7000	1.80281	5.27836	2.29747	19.4288
8000	1.87627	5.70272	2.38804	21.8339

Table 4. Values of the right part of inequality (4).

T	$m = 1000$	$m = 2000$
1000	2.42924e-005	8.58868e-006
2000	2.42123e-005	8.56035e-006
3000	2.41834e-005	8.55011e-006
4000	2.41720e-005	8.54608e-006
5000	2.41702e-005	8.54547e-006
6000	2.41755e-005	8.54733e-006
7000	2.41869e-005	8.55136e-006
8000	2.42042e-005	8.55748e-006

References

[1] Rukhin A. et al., *NIST Special Publication 800-22. A statistical test suite for random and pseudorandom number generators for cryptographic applications*, 2000.

[2] Rukhin A. et al., *NIST Special Publication 800-22r1a. A statistical test suite for random and pseudorandom number generators for cryptographic applications*, 2010.

[3] Doganaksoy A., Gologlu F., "On Lempel-Ziv Complexity of Sequences", *LNCS*, Sequences and Their Applications – SETA 2006, **4086**, ed. Gong G., Helleseth T., Song HY., Yang K., Springer, Berlin, Heidelberg, 2006.

[4] Mihailov V.G., "Formulae to Calculate Distributions of Lempel-Ziv Statistic and Relative Statistics", *OPPM Surveys in Applied and Industrial Mathematics*, **14**:3 (2007), 461–473.

[5] Tyurin I.S., "An improvement of the residual in the Lyapunov theorem", *Teor. Veroyatnost. i Primenen.*, **56**:4 (2011), 808–811.

[6] Ivchenko G.I., Medvedev Y.I., *Mathematical Statistics*, High School, Moscow, 1984, In Russian.

[7] Ryabko B.Ya., Monarev V.A., "Using information theory approach to randomness testing", *Journal of Statistical Planning and Inference*, **133**:1 (2003), 95–110.

Streebog Compression Function as PRF in Secret-key Settings

Vitaly Kiryukhin

JSC «InfoTeCS», Russia
LLC «SFB Lab», Russia
Vitaly.Kiryukhin@infotecs.ru

Abstract

Security of the many keyed hash-based cryptographic constructions (such as HMAC) depends on the fact that the underlying compression function $\mathsf{g}(H, M)$ is a pseudorandom function (PRF). This paper presents key-recovery algorithms for 7 rounds (of 12) of Streebog compression function. Two cases were considered, as a secret key can be used: the previous state H or the message block M. The proposed methods implicitly show that Streebog compression function has a large security margin as PRF in the above-mentioned secret-key settings.

Keywords: Streebog, PRF, truncated differentials, rebound, polytopic cryptanalysis.

1 Introduction

Hash function is one of the most commonly used cryptographic primitives. Usually, the following three security properties are expected from a non-keyed hash function:

1) preimage resistance (for a given value $\mathsf{Hash}(Msg)$ it is hard to obtain Msg);

2) second preimage resistance (for a given message Msg it is difficult to find a different Msg' such that $\mathsf{Hash}(Msg) = \mathsf{Hash}(Msg')$);

3) collision resistance (it is hard to construct a nontrivial message pair (Msg, Msg') such that $\mathsf{Hash}(Msg) = \mathsf{Hash}(Msg')$).

For hash functions based on the Merkle-Damgård scheme [3, 2], similar requirements are imposed on the underlying compression function $\mathsf{g}(H_{prev}, M) = H_{next}$ (where M is a fixed-length block of the hashed message, H_{prev} and H_{next} are the previous and the next internal states respectively).

Russian hash function Streebog [1], like many others, uses slightly modified Merkle-Damgård approach. Its compression function is based on a 12-rounds AES-like [22] block cipher in Miyaguchi-Preneel mode. The previous

internal state is transformed to 13 round keys for the block cipher. The internal state consists of 8×8 bytes ($n = 512$ bits). The output length of hash function can be either 512 or 256-bit.

Over recent years, Streebog (as well as its compression function and block cipher) was subjected to a thorough analysis by many experts. We cite papers devoted to the preimage [11, 12, 16, 9], the second preimage [7], various types of the collisions [10, 11, 12, 13, 14]. Many articles describe so-called «known-key» (and «chosen-key») distinguishers [8, 13, 12, 17, 18]. The latter demonstrate some non-random structural properties of the transformation (a compression function or a block cipher) by constructing the corresponding set of input-output pairs.

Keyless hash function is often used as part of the secret-key cryptoalgorithms. Some of the most well-known examples are HMAC and NMAC [6]. The security of such algorithms depends significantly on the fact that the compression function is a PRF. Let one of the arguments $\mathtt{g}(H, M)$ be a secret key and an adversary can adaptively choose blocks for the other input and observe outputs. It is clear that a simple key guessing with time-complexity about $t = 2^n$ can be used to distinguish between $\mathtt{g}(H, M)$ and a random function. In some cases, straightforward birthday-paradox distinguisher with data-complexity $q = 2^{n/2}$ can also be mounted. Is it possible to construct more efficient algorithms for a specific instance of $\mathtt{g}(H, M)$? In our paper we consider round-reduced Streebog compression function.

To the best of our knowledge, there is only one paper [15] on the subject[1]. The authors [15] utilize impossible differential properties to mount secret-state (secret-IV) recovery attacks on 6.75-rounds.

Next, we present key-recovery algorithms for 7-round Streebog compression function.

In section 3 we describe algorithm for the secret-state case. The proposed method is based on polytopic approach [5]. A naive algorithm for «generalized birthday problem» [23] is also an important part of the method.

In section 4 the second secret-message case is considered. The rebound technique [25] is used to obtain usable pairs of non-secret states. The truncated differential [20] method is then applied to recover the secret message.

Comparative characteristics of algorithms are presented in table 1. Note also that the initial data processing was not taken into account when calculating the complexity of attacks [15] (so $t < q$, «Time» is less than «Data»).

[1]For completeness, it is worth noting that key-recovery attack on HMAC-Streebog was presented in [24] as the extension of the generic state-recovery attack on HMAC with an arbitrary Merkle-Damgård hash-function. Data-complexity of attack [24] is significantly more than HMAC allowable «provable secure» bounds [6]. The attack also does not depend on the properties of the compression function.

Our results provide an additional argument showing that Streebog compression function (as a PRF) has a significant security margin.

Setting	Rounds	Time	Memory	Data	Description
secret H	6.75	$2^{399.5}$	2^{349}	2^{483}	[15]
	6.75	$2^{261.5}$	2^{205}	$2^{495.5}$	[15]
	7	2^{421}	2^{354}	2^{64}	Section 3
	12	2^{256}	2^{256}	2^{256}	birthday-paradox distinguisher
	12	2^{512}	$\sim$	2	key guessing
secret M	7	2^{240}	2^{20}	2^{113}	Section 4
	12	2^{512}	$\sim$	2	key guessing

Table 1: Attacks on Streebog compression functions in secret-key settings. «Time» (t) in g computations, «Memory» in n-bit blocks, «Data» (q) in chosen message-output pairs.

2 Definitions

Let $\mathbb{F}_{2^8}$ be a finite field. Each element of $\mathbb{F}_{2^8}$ can be interpreted as an integer or binary vector. Denote $v \times v$ matrix space over $\mathbb{F}_{2^8}$ by $\mathbb{F}_{2^8}^{v\times v}$ (we also use symbol $\mathbb{F}_{2^8}^{v}$ as a vector space). Elements from $\mathbb{F}_{2^8}^{v\times v}$ will be denoted by capital letters: A, B. Blocks of states and messages also belong to $\mathbb{F}_{2^8}^{v\times v}$. Elements of a matrix are indexed by $0 \le i, j \le v-1$ (for example, $a = A[0,0]$ is an element from the upper-left corner of the matrix). $A[i,\cdot]$ is i-th row of A, $A[\cdot, j]$ is j-th column of A.

Denote bitwise xor operation by symbol $\oplus$. This operation is defined naturally for all the objects under consideration.

Let us have a sequence of blocks

$$B_0, ..., B_d \in \mathbb{F}_{2^8}^{v\times v}, d \in \mathbb{N},$$

then we refer to $\boldsymbol{\Delta B} = B_0 \oplus B_1 \in \mathbb{F}_{2^8}^{v\times v}$ as a difference and to a sequence

$$\boldsymbol{\delta B} = (B_0 \oplus B_1, B_0 \oplus B_2, ..., B_0 \oplus B_d) \in \left(\mathbb{F}_{2^8}^{v\times v}\right)^d \tag{1}$$

as a d-difference. Differences are indicated in bold text: $\boldsymbol{\delta M}$, $\boldsymbol{\Delta K_4}$.

The d-difference $\boldsymbol{\delta B} \in \left(\mathbb{F}_{2^8}^{v\times v}\right)^d$ can also be interpreted as $v\times v$ «columns» of d bytes each: $\boldsymbol{\delta B} \in \left(\mathbb{F}_{2^8}^{d}\right)^{v\times v}$, $\boldsymbol{\delta B[i,j]} \in \mathbb{F}_{2^8}^{d}$. If $\boldsymbol{\Delta B[i,j]} \neq 0$ (resp. $\boldsymbol{\delta B[i,j]} \neq \mathbf{0}$) then we say that the position (i,j) is active, otherwise inactive, $0 \le i, j \le v-1$.

The differential (resp. polytopic) trail is the sequence of the differences (resp. d-differences) after each transformation in the cipher.

The transformations over $\mathbb{F}_{2^8}^{v\times v}$ (also over $\mathbb{F}_{2^8}^{v}$ and $\mathbb{F}_{2^8}$) are denoted by sans serif font: f, S, L. The notation LS indicates a composition of transformations,

where S applies first (the reverse order «left-to-right» is used on the figures). The inverse transformations are specified by f^{-1}.

Streebog

Streebog compression function $\mathsf{g}_N(H, M)$ employs AES-like XSPL-cipher E in the Miyaguchi-Preenel mode

$$\mathsf{g}_N(H, M) = \mathsf{E}(H \oplus N, M) \oplus H \oplus M = R, \text{ where}$$

$H \in \mathbb{F}_{2^8}^{v \times v}$ is the previous state of the hash function;
$M \in \mathbb{F}_{2^8}^{v \times v}$ is the message block;
$N \in \mathbb{F}_{2^8}^{v \times v}$ is the number of previously hashed bits;
$R \in \mathbb{F}_{2^8}^{v \times v}$ is the output (the next state of hash function).

The block cipher E consists of 12 rounds and a post-whitening key addition. Each round consists of four operations:

X – modulo 2 addition of an input block with a round key;

S – parallel application of the fixed bijective substitution s to each byte of the state;

P – transposition of the state;

L – parallel application of the linear transformation l to each row of the state. In [22], it was shown that l-transformation can be represented as the MDS matrix $\mathbb{L}$ over $\mathbb{F}_{2^8}^{8 \times 8}$.

The block cipher formula is

$$\mathsf{E}(K, M) = \mathsf{X}[K_{13}]\mathsf{LPSX}[K_{12}] \ldots \mathsf{LPSX}[K_2]\mathsf{LPSX}[K_1](M).$$

The state size consists of $n = 512$ bits ($v \times v = 8 \times 8$ bytes).

The key schedule uses round constants $C_i \in \mathbb{F}_{2^8}^{v \times v}$, $i = 1, 2, \ldots, 12$, and round keys $K_i \in \mathbb{F}_{2^8}^{v \times v}$, $i = 1, 2, \ldots, 13$ are derived from a master key K_0 as follows:

$$K_0 = H \oplus N,\ K_1 = \mathsf{LPS}(H \oplus N),\ K_{i+1} = \mathsf{LPS}(K_i \oplus C_i),\ i = 1, 2, \ldots, 12.$$

We also denote the intermediate states before X, S, P, L transformations in i-th round as X_i, Y_i, Z_i, W_i correspondingly ($X_1 = M$, $Y_1 = M \oplus K_1$, $Z_1 = \mathsf{S}(Y_1)$, $W_1 = \mathsf{P}(Z_1)$, etc.). The states in the key schedule are denoted in a similar way $HX_i = K_i$, HY_i, HZ_i, HW_i, where $H = HX_0$, $HX_1 = \mathsf{LPS}(H \oplus N)$ etc.

We define an r-round compression function with $r + 1$ round keys as:

$$\mathsf{g}(H, M) = (\mathsf{X}[K_{r+1}]\mathsf{LPSX}[K_r] \ldots \mathsf{LPSX}[K_1](M)) \oplus H \oplus M.$$

Next, we also assume that N is an arbitrary constant C_0.

3 State as a secret key

Let the state H be a secret. An adversary knows a message M and an output R.

$$\mathsf{g}(H, M) = \mathsf{E}(H, M) \oplus H \oplus M = R.$$

Hence, the analysis is reduced to the block cipher

$$\mathsf{E}(H, M) \oplus H = R \oplus M = \widetilde{R},$$

$$\mathsf{E}(H, M) \oplus H = \mathsf{X}[K_{r+1} \oplus H]\mathsf{LPSX}[K_r] \ldots \mathsf{LPSX}[K_1](M),$$

where the last round key is $\widetilde{K}_{r+1} = K_{r+1} \oplus H$.

A secure block cipher can be used as a secure PRF up to about $q = 2^{n/2}$ queries [19]. Thus, any algorithm that requires more message-output pairs can't be considered as a direct threat to a PRF. The generic limit of the time complexity $t = 2^n$ is defined by straightforward key guessing.

We propose the polytopic (multidimensional differential) based key-recovery algorithm against 7-rounds. The method consists of the following steps:

1. Choose structure of 2^{64} messages M;

2. Guess 64 bits of the first key K_1. Partially encrypt all messages up to the second L-transformation;

3. Choose about 2^7 blocks (of 2^{64}) and obtain d-difference $\boldsymbol{\delta W_2}$ with only one active S-box;

4. Propagate $\boldsymbol{\delta W_2}$ forward to $\boldsymbol{\delta W_5[0,0]}$ by guessing 136 bits of the intermediate states;

5. Propagate $\boldsymbol{\delta \widetilde{R}}$ backward to $\boldsymbol{\delta X_6[0,0]}$ by guessing 72 bits of the intermediate states (similarly and independently for $\boldsymbol{\delta X_6[0,1]}$,...,$\boldsymbol{\delta X_6[0,7]}$);

6. Check by using a naive algorithm for «generalized birthday problem» that $\boldsymbol{\delta W_5[0,0]}$ can be obtained via inverse linear transformation $\mathsf{l}^{-1}(\boldsymbol{\delta X_6[0,0]}, \ldots, \boldsymbol{\delta X_6[0,7]})$;

7. If the check failed in the previous step then go back to step 2 and try another bits of K_1. If the check is passed then the key bits and the state bits are guessed correctly.

Let's look at the steps in more detail.

The first and second steps are designed to bypass the first round (figure 1). We use the structure of 2^{64} messages. One column in each message takes all possible values ($M[\cdot, 0]$ in the picture). All other seven columns are arbitrary constants ($M[\cdot, 1]$, ..., $M[\cdot, 7]$ in the picture). For any values of K_1 and K_2, this will also be true for the columns in $W_2 = \mathsf{PSX}[K_2]\mathsf{LPSX}[K_1](M)$.

Guess column $K_1[\cdot, 0]$ and compute row $K_2[0, \cdot]$. In this case, all the values in column $W_2[\cdot, 0]$ are exactly known.

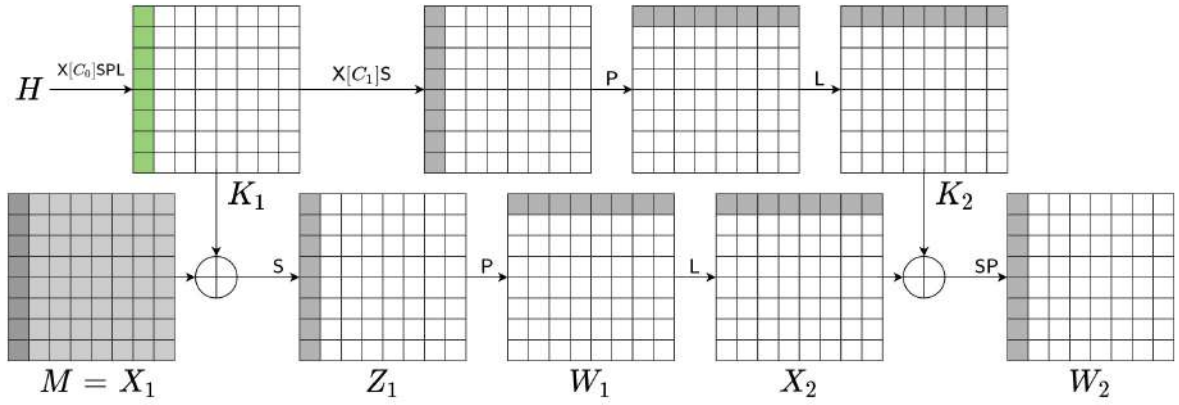

Figure 1: Steps 1-2. Guessed bits are highlighted with green. Computed or known values are denoted by gray cells. The formulas are given in reverse (left-to-right) notation.

Recall that for each of the 2^{64} states W_2 in the structure, columns $W_2[\cdot, 1]$, ..., $W_2[\cdot, 7]$ are unknown constants. It is easy to find such 2^7 states $W_2^{(0)}$,$W_2^{(1)}$,...,$W_2^{(d)}$, $d = 2^7 - 1$ that d-difference

$$\boldsymbol{\delta W_2} = (W_2^{(0)} \oplus W_2^{(1)}, W_2^{(0)} \oplus W_2^{(2)}, ..., W_2^{(0)} \oplus W_2^{(d)}) \in \left(\mathbb{F}_{2^8}^{v \times v}\right)^d$$

will have only one active byte $\boldsymbol{\delta W_2[0, 0]}$. In other words, we select states W_2 so that the bytes $W_2[1, 0]$,...,$W_2[7, 0]$ are also constants. We choose the corresponding outputs $\widetilde{R}$ and compute d-difference $\boldsymbol{\delta \widetilde{R}}$.

The difference (and d-difference) is unambiguously propagated through X, L and P transformations, but we have to guess the state bytes to propagate the difference through S. Obviously, zero difference remains the same after any transformation.

If the bytes from $K_1[\cdot, 0]$ are guessed correctly, then the trail from $\boldsymbol{\delta W_2}$ to $\boldsymbol{\delta \widetilde{R}}$ must exist. Otherwise, it's possible to check that there are no appropriate trails (or almost none).

The d-difference $\boldsymbol{\delta W_2}$ propagates through L and $\mathsf{X}[K_3]$ to $\boldsymbol{\delta Y_3}$, which contains eight active bytes $\boldsymbol{\delta Y_3[0, \cdot]}$ (see figure 2). Recall that this is true due to the MDS property of L [22]. We guess $Y_3[0, \cdot]$ and obtain 2^{64} possible d-differences $\boldsymbol{\delta Z_3}$. Next, $\boldsymbol{\delta Y_4}$ is computed by linear propagation through P, L, $\mathsf{X}[K_4]$. All byte positions in $\boldsymbol{\delta Y_4}$ are active.

By guessing only one column $Y_4[\cdot, 0]$ we obtain 2^{128} possible d-differences $\boldsymbol{\delta Z_4[\cdot, 0]}$. The remaining seven columns in $\boldsymbol{\delta Z_4}$ are active but unknown to us.

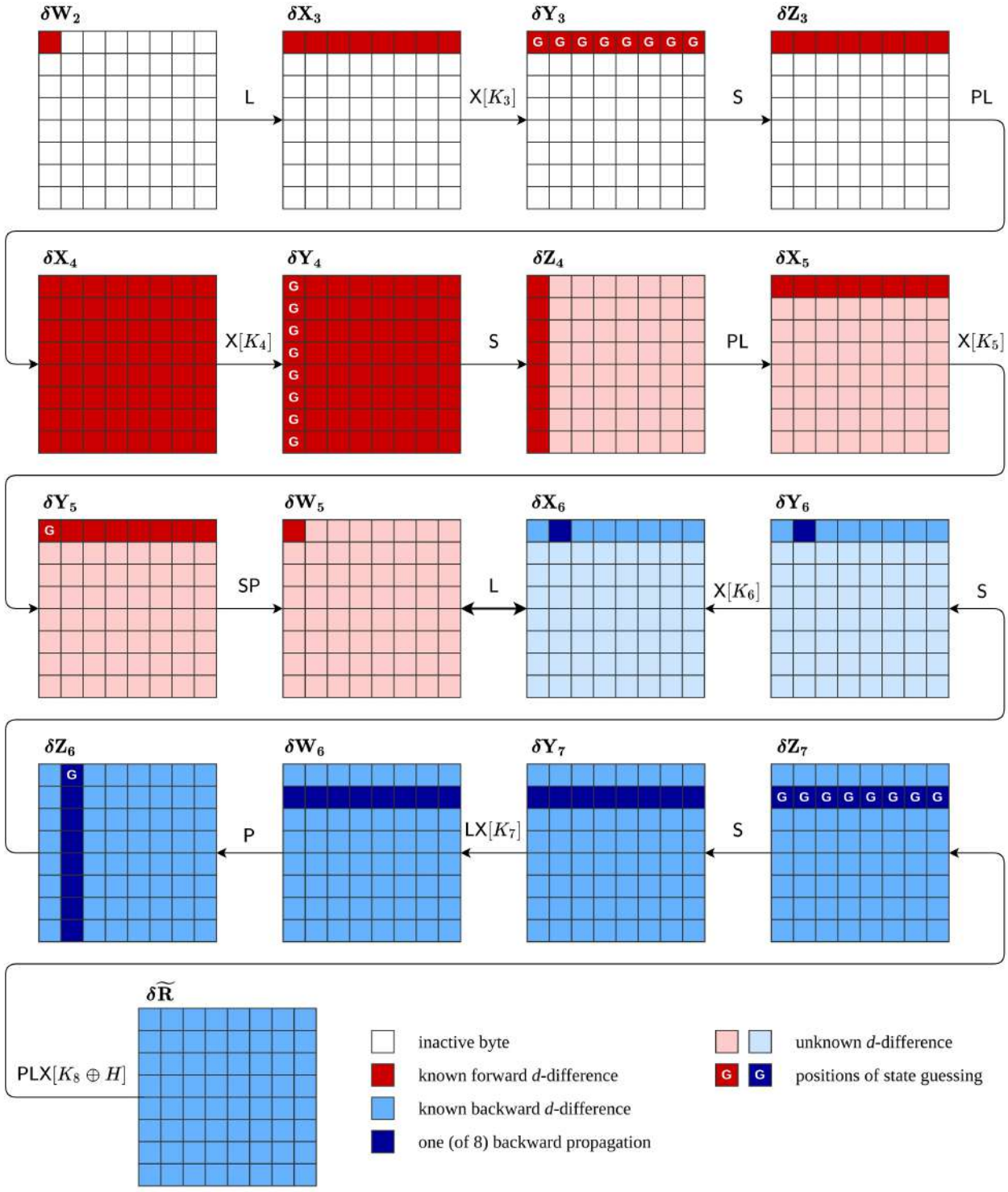

Figure 2: Steps 4-5. Forward and backward d-difference propagations.

The d-difference $\boldsymbol{\delta Y_5[0,\cdot]}$ is calculated in the same way for each $\boldsymbol{\delta Z_4[\cdot,0]}$. Another byte $Y_5[0,0]$ allows us to compute $\boldsymbol{\delta Z_5[0,0]} \in \mathbb{F}_{2^8}^d$ and $\boldsymbol{\delta W_5[0,0]} = \boldsymbol{\delta Z_5[0,0]}$.

Thus, we have $2^{64} \cdot 2^{64} \cdot 2^8 = 2^{136}$ values of $\boldsymbol{\delta W_5[0,0]}$, stored in the array

$\mathcal{L}_{\text{frw}}$. Each d-difference corresponds to the sequence of bytes

$$Y_3[0,0], Y_3[0,1], \ldots, Y_3[0,7],\ Y_4[0,0], Y_4[1,0], \ldots, Y_4[7,0],\ Y_5[0,0].$$

Consider the backward direction. We know d-difference $\boldsymbol{\delta}\widetilde{\boldsymbol{R}}$ and can compute $\boldsymbol{\delta Z_7}$ by backward propagation through $\mathsf{X}[K_8 \oplus H]$, L^{-1}, P^{-1}.

Guess one row of Z_7 (bytes $Z_7[1,0]$,...,$Z_7[1,7]$ on figure 2). We obtain 2^{64} values of corresponding column in $\boldsymbol{\delta Z}_6$. Guess one byte in Z_6 (byte $Z[0,1]$ on figure). Hence, we can compute 2^{72} possible values of $\boldsymbol{\delta Y_6[0,1]}$ and $\boldsymbol{\delta X_6[0,1] = \delta Y_6[0,1]}$.

Similar actions are performed in parallel for the other seven rows in $\boldsymbol{\delta Z_7}$. As a result, we computed values of $\boldsymbol{\delta X_6[0,0]}$, $\boldsymbol{\delta X_6[0,1]}$, ..., $\boldsymbol{\delta X_6[0,7]}$. Eight lists $\mathcal{L}_0$, $\mathcal{L}_1$, ...,$\mathcal{L}_7$ of 2^{72} values (d-difference) each were stored.

Hypothetically, all $(2^{72})^8 = 2^{576}$ values of $\boldsymbol{\delta X_6[0,\cdot]}$ can be computed, and therefore, $\boldsymbol{\delta W_5[0,\cdot]} = \mathsf{l}^{-1}(\boldsymbol{\delta X_6[0,\cdot]})$. Next, each variant of $\boldsymbol{\delta W_5[0,0]}$ can be checked by searching among previously computed ones in the forward direction. Obviously, this way is much expensive.

Let's rewrite the expression for the inverse linear transformation

$$W_5[0,\cdot] \times \mathbb{L} = X_6[0,\cdot],$$

$$W_5[0,\cdot] = X_6[0,\cdot] \times \mathbb{L}^{-1},$$

$$W_5[0,0] = c_0 \cdot X_6[0,0] \oplus c_1 \cdot X_6[0,1] \oplus \ldots \oplus c_7 \cdot X_6[0,7],$$

where: $\mathbb{L} \in \mathbb{F}_{2^8}^{8\times 8}$ (resp. $\mathbb{L}^{-1}$) is the MDS matrix of the linear transformation l (resp. the inverse transformation l^{-1}); $c_0, c_1, \ldots, c_7 \in \mathbb{F}_{2^8}$ are the coefficients from the column of $\mathbb{L}^{-1}$. The matrix representation from [22] is implicitly used here, but the expressions can be rewritten for the binary 64×64 matrix.

The same equality is also true for the correct pairs of the differences (d-differences)

$$\boldsymbol{\delta W_5[0,0]} = c_0 \cdot \boldsymbol{\delta X_6[0,0]} \oplus c_1 \cdot \boldsymbol{\delta X_6[0,1]} \oplus \ldots \oplus c_7 \cdot \boldsymbol{\delta X_6[0,7]},$$

where $c_i \cdot \boldsymbol{\delta X_6[0,i]} = (c_i \cdot \boldsymbol{\Delta x_1}, c_i \cdot \boldsymbol{\Delta x_2}, \ldots, c_i \cdot \boldsymbol{\Delta x_d})$, $\boldsymbol{\Delta x_j} \in \mathbb{F}_{2^8}$, $i = 0, ..., 7$, $j = 1, ..., d$, $\boldsymbol{\delta X_6[0,i]} \in \mathbb{F}_{2^8}^d$. Therefore, we can proceed to the simpler problem

$$\mathcal{L}_{\text{frw}}[p_{\text{frw}}] = c_0 \cdot \mathcal{L}_0[p_0] \oplus c_1 \cdot \mathcal{L}_1[p_1] \oplus \ldots \oplus c_7 \cdot \mathcal{L}_7[p_7], \tag{2}$$

we should find the indexes p_0, p_1, ..., p_7, p_{frw} so that the equation is correct, or prove that there are no such indexes. We obtain some example of a generalized birthday problem [23], but we have no task to find at least some «collision».

Our goal is to build only one unique correct solution. All others should be discarded. Because of this, we apply a naive approach.

Rearrange the components of the equation

$$\mathcal{L}_{\text{frw}}[p_{\text{frw}}] \oplus c_0 \cdot \mathcal{L}_0[p_0] \oplus c_1 \cdot \mathcal{L}_1[p_1] \oplus c_2 \cdot \mathcal{L}_2[p_2] = c_3 \cdot \mathcal{L}_3[p_3] \oplus \ldots \oplus c_7 \cdot \mathcal{L}_7[p_7]. \quad (3)$$

Combine all lists from the left side (3) into one. We obtain an array $\mathcal{L}_{\text{left}}$ containing $2^{136} \cdot \left(2^{72}\right)^3 = 2^{352}$ elements (d-differences) from $\mathbb{F}_{2^8}^d$. The hash table is used to store items. The d-difference is the «key», the guessed state bits are the «value». Hence, each item requires $(8 \cdot d + 352) < 3n$ bits of memory to be stored (in total, less than 2^{354} n-bit states).

It's not hard to see, that the right side (3) generates $(2^{72})^5 = 2^{360}$ items ($\mathcal{L}_{\text{right}}$) that can be constructed dynamically by iterating through 360 bits.

If the arbitrary element from $\mathcal{L}_{\text{right}}$ is found in $\mathcal{L}_{\text{left}}$, then we assume that the trail from $\boldsymbol{\delta W_2}$ to $\boldsymbol{\delta \widetilde{R}}$ exists and all the bits ($K_1[\cdot, 0]$, $Y_3[0, \cdot]$, $Y_4[\cdot, 0]$, $Y_5[0, 0]$, $Z_6[0, \cdot]$, Z_7) are guessed correctly. What is the average number of false assumptions? We have $2^{360+352} = 2^{712}$ pairs of d-differences ($8d$-bit values). Thus, under the hypothesis of a random and uniform distribution, we get $2^{64} \cdot 2^{712} \cdot 2^{-127 \cdot 8} = 2^{-240} \approx 0$ false solutions (the factor 2^{64} emerges due to the key guessing at step 2). The value of d can be reduced, but this does not significantly affect the estimation of the time complexity.

If no element from $\mathcal{L}_{\text{right}}$ is found in $\mathcal{L}_{\text{left}}$ then we guess the next value of $K_1[\cdot, 0]$. Steps 3-6 are repeated again.

The last round key $\widetilde{K}_8 = K_8 \oplus H$ is computed via the known state Z_7 and the corresponding output $\widetilde{R}$

$$\widetilde{K}_8 = \widetilde{R} \oplus \mathsf{LP}(Z_7).$$

In this way, the challenge is reduced to six rounds.

There is a different approach. The bytes of the other seven rows in Z_6 are determined by parallel guessing of $(Y_5[0, 1], Z_6[1, \cdot])$, $(Y_5[0, 2], Z_6[2, \cdot])$, ..., $(Y_5[0, 7], Z_6[7, \cdot])$. The correct values are obtained via similar check of the trail from $\boldsymbol{\delta X_6[i, \cdot]}$ to $\boldsymbol{\delta W_5[i, 0]}$ through inverse linear transformation l^{-1}, $i = 1, 2, \ldots 7$. Next, we use simple relation $Z_7 = \mathsf{S}(K_7 \oplus \mathsf{LP}(Z_6))$ and recover the round key

$$K_7 = \mathsf{S}^{-1}(Z_7) \oplus \mathsf{LP}(Z_6).$$

The secret H is computed due to the invertibility of the key schedule.

By the end, the time complexity of the key-recovery algorithm is

$$t = \underbrace{2^{64}}_{K_1[\cdot,0]} \cdot \left(\underbrace{16 \cdot 2^{64}}_{\text{step 2}} + \underbrace{d' \cdot 2^{136}}_{\text{step 4}} + \underbrace{d' \cdot 8 \cdot 2^{72}}_{\text{step 5}} + \underbrace{d' \cdot 2^{352}}_{\mathcal{L}_{\text{left}}} + \underbrace{d' \cdot 2^{360}}_{\mathcal{L}_{\text{right}}} + \underbrace{7 \cdot d' \cdot 2^{72}}_{Z_6 \text{ recovery}} \right),$$

where $d' = d+1 = 2^7$. In total, $t \approx 2^{431}$ Sbox computations. We estimate the computation complexity of the 7-round compression function as $2 \cdot 7 \cdot 64 \approx 2^{10}$ Sbox computations (memory access operations). As a result, we get time complexity $t = 2^{431} \cdot 2^{-10} = 2^{421}$. The proposed method requires less than 2^{354} (n-bit states) of memory. The data complexity is 2^{64} chosen pairs (M, R).

The described algorithm is deterministic – the probability of success is equal to one. Meanwhile, the most effective method [21] against 7-round AES-128 uses a rare event (truncated differential).

Note also that the ideas of the proposed method can be applied to 6 rounds of AES-128 (similar to steps 3-7 above). We were able to build an attack with time complexity about 2^{120} memory access operations and a small amount of the chosen plaintexts $q = d + 1 < 2^5$. Due to the relatively high time complexity, we were unable to extend the attack to 7 rounds (as in steps 1-2).

4 Message as a secret key

Let the message M be a secret

$$\mathsf{g}(H, M) = \mathsf{E}(H, M) \oplus H \oplus M = R.$$

An adversary has a full control over the master-key H and the round keys of the underlying block cipher

$$\mathsf{E}(H, M) \oplus M = R \oplus H = \widetilde{R}.$$

The function $\mathsf{E}(H, M) \oplus M$ with secret M is a secure PRF in the ideal cipher model (i.e. if E is a family of random permutations). The proof can be found, for example, in [4, Theorem 8.5]. In this case, there is no simple birthday-paradox distinguisher. Only brute-force key search is applicable attack.

Consider the algorithm against seven rounds, which consists of two stages.

«Offline» stage. Due to the rebound approach [25], about 2^{112} pairs (H, H') are formed ($q = 2^{113}$). Each pair generates a truncated differential trail $\boldsymbol{\Delta K_1} \to \boldsymbol{\Delta K_2} \to \ldots \to \boldsymbol{\Delta K_8}$ with the pattern

$$\text{«}8 - 1 - 8 - 64 - 16 - 16 - 64 - 64\text{»} \text{ of the active S-boxes.}$$

«Online» stage. For each H, we get the output $\widetilde{R}$ (resp. for H' and $\widetilde{R}'$). The truncated related-key differential trail $\boldsymbol{\Delta M} \to \ldots \to \boldsymbol{\Delta \widetilde{R}}$ is realized with a probability of at least 2^{-112}. The pattern of the active S-boxes is

«8 – 0 – 8 – 0 – 16 – 16 – 64 – 64». For each pair $(\widetilde{R}, \widetilde{R}')$ we construct about 2^{128} possible values of the unknown internal state. Each solution is checked directly. If the rare event actually occurred, then among the constructed solutions there will be a true one.

In more detail.

We should construct the suitable round keys for the block cipher. Choose arbitrary nonzero bytes in one column of the difference $\boldsymbol{\Delta HW_3}$ (highlighted with green on figure 3). Propagate forward to $\boldsymbol{\Delta HY_4} = \mathsf{L}(\boldsymbol{\Delta HW_3})$. Similarly in the backward direction $\boldsymbol{\Delta HZ_4} = \mathsf{P}^{-1}\mathsf{L}^{-1}(\boldsymbol{\Delta K_5})$. We choose two columns in $\boldsymbol{\Delta K_5}$ so that all bytes in $\boldsymbol{\Delta HZ_4}$ are active. Thus, we have $255^8 \cdot (2^{16} - 8 \cdot 255 - 1)^8 \approx 2^{191.6}$ pairs $(\boldsymbol{\Delta HY_4}, \boldsymbol{\Delta HZ_4})$. Solve equation $\mathsf{S}(HY_4 \oplus \boldsymbol{\Delta HY_4}) \oplus \mathsf{S}(HY_4) = \boldsymbol{\Delta HZ_4}$. We get a total of about $2^{190.4}$ solutions (see also Appendix A).

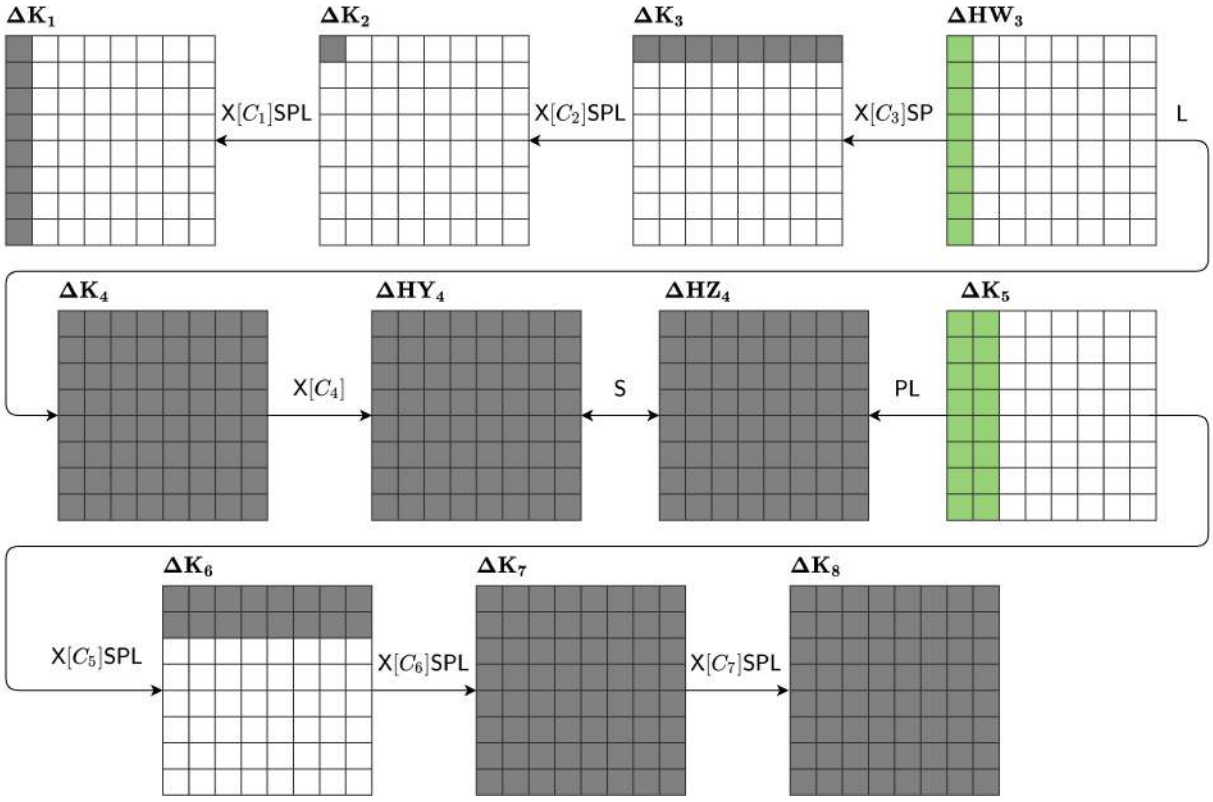

Figure 3: «Offline» stage. Truncated differential trail over round keys.

Next, in so-called «outbound phase» we compute

$$K_8 = \mathsf{LPSX}[C_7] \dots \mathsf{LPS}(HY_4) \text{ and } K_1 = \mathsf{X}[C_1]\mathsf{S}^{-1} \dots \mathsf{P}^{-1}\mathsf{L}^{-1}\mathsf{X}[C_4](HY_4).$$

We expect almost all trails $\boldsymbol{\Delta K_6} \to \boldsymbol{\Delta K_7} \to \boldsymbol{\Delta K_8}$ match the pattern «16 – 64 – 64». The trails with smaller number of active S-boxes are also appropriate. We assume that the part $\boldsymbol{\Delta K_1} \leftarrow \boldsymbol{\Delta K_2} \leftarrow \boldsymbol{\Delta K_3}$ of the constructed

trail match the pattern «8 – 1 – 8» with probability $255/255^8 \approx 2^{-56}$ due to the transition «1 ← 8».

As a result we obtain about $2^{134.4} = 2^{190.4-56}$ pairs (H, H').

We request $(\widetilde{R}, \widetilde{R}')$ for each (H, H') from the «oracle». Consider the propagation of the differences with secret M (figure 4). Obviously, $M = M'$ and $\boldsymbol{\Delta M} = 0$. Before the first non-linear layer $\boldsymbol{\Delta Y_1} = \boldsymbol{\Delta K_1} \oplus \boldsymbol{\Delta M} = \boldsymbol{\Delta K_1}$. We hope that $\boldsymbol{\Delta HZ_1} = \boldsymbol{\Delta Z_1}$. The transition $\boldsymbol{\Delta HY_1} \to \boldsymbol{\Delta HZ_1}$ is possible, hence, the probability $\boldsymbol{\Delta Y_1} \to \boldsymbol{\Delta Z_1}$ is not less than $(2/256)^8 = 2^{-56}$. If actually $\boldsymbol{\Delta HZ_1} = \boldsymbol{\Delta Z_1}$ then

$$\boldsymbol{\Delta Y_2} = \boldsymbol{\Delta K_2} \oplus \boldsymbol{\Delta X_2} = \mathsf{LP}(\boldsymbol{\Delta HZ_1}) \oplus \mathsf{LP}(\boldsymbol{\Delta Z_1}) = 0.$$

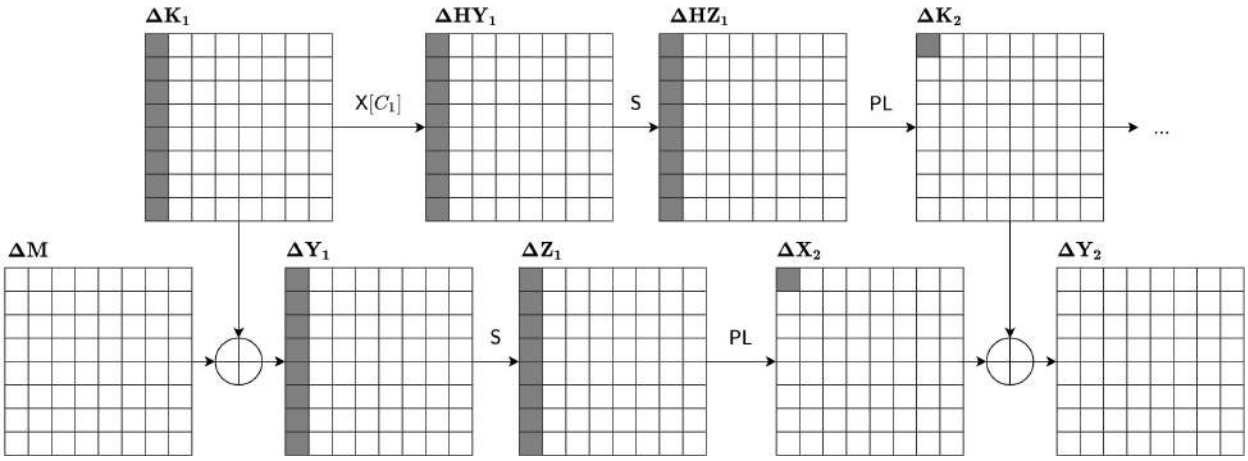

Figure 4: «Online» stage. Truncated related-key differential trail. The first round.

The same is true for $\boldsymbol{\Delta Y_3} = \boldsymbol{\Delta K_3}$ and «parallel» transitions $\boldsymbol{\Delta HY_3} \to \boldsymbol{\Delta HZ_3}$, $\boldsymbol{\Delta Y_3} \to \boldsymbol{\Delta Z_3}$ (figure 5). We also assume that $\Pr(\boldsymbol{\Delta Z_3} = \boldsymbol{\Delta HZ_3}) \geq 2^{-56}$.

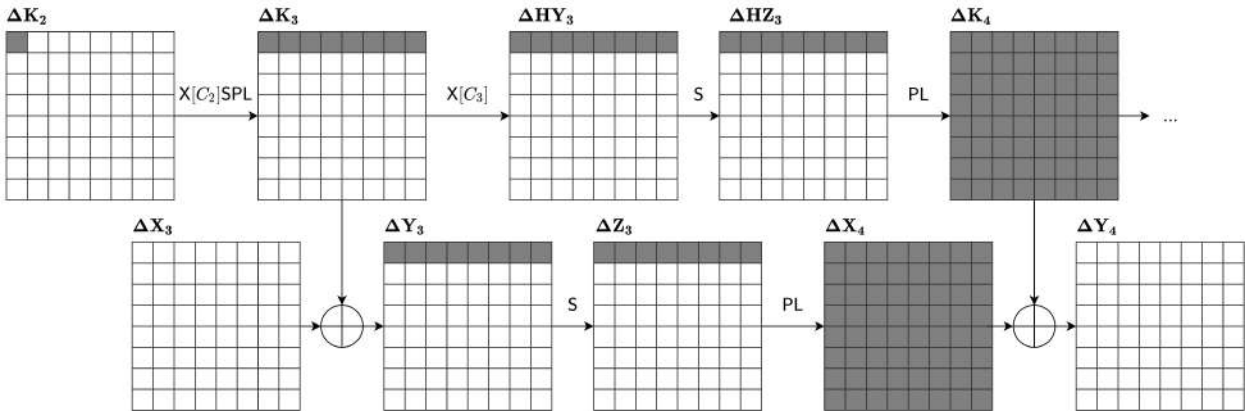

Figure 5: «Online» stage. The third round.

Thus, in the fifth round $\Pr(\boldsymbol{\Delta X_5 = 0}) = \Pr(\boldsymbol{\Delta K_5 = \Delta Y_5}) \geq 2^{-56 \cdot 2}$. So both differences $\boldsymbol{\Delta HW_6}$ and $\boldsymbol{\Delta W_6}$ have only two active columns each (figure 6). Each row in $\boldsymbol{\Delta Y_7}$ belongs to a set of 2^{16} differences (not 2^{64})

$$\boldsymbol{\Delta Y_7}[\boldsymbol{i}, \cdot] = \mathsf{l}(\boldsymbol{\Delta W_6}[\boldsymbol{i}, \boldsymbol{0}]) \oplus \mathsf{l}(\boldsymbol{\Delta HW_6}[\boldsymbol{i}, \boldsymbol{0}]) = \mathsf{l}(\boldsymbol{\Delta W_6}[\boldsymbol{i}, \boldsymbol{0}] \oplus \boldsymbol{\Delta HW_6}[\boldsymbol{i}, \boldsymbol{0}]),$$

where the difference $(\boldsymbol{\Delta W_6}[\boldsymbol{i}, \boldsymbol{0}] \oplus \boldsymbol{\Delta HW_6}[\boldsymbol{i}, \boldsymbol{0}])$ contains no more than two active bytes, $i = 0, 1, ..., 7$. For the sake of simplicity, it is assumed that all the rows in $\boldsymbol{\Delta Y_7}$ are active (this is not the case with a probability of only about $1 - (1 - 2^{-16})^8 \approx 2^{-13}$).

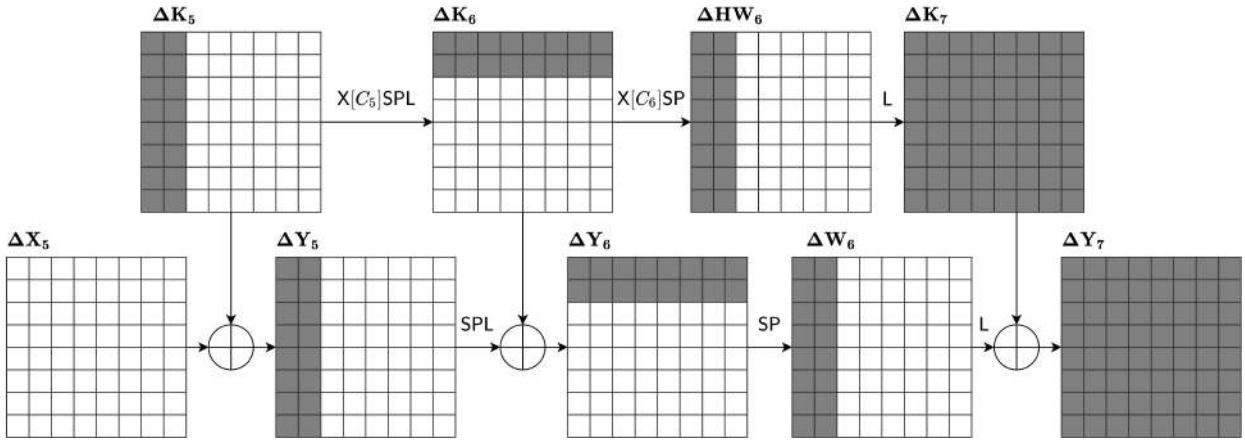

Figure 6: «Online» stage. Propagation to $\boldsymbol{\Delta Y_7}$.

Recall that $\boldsymbol{\Delta K_8}$ and the output difference $\boldsymbol{\Delta \widetilde{R}}$ are known, $\boldsymbol{\Delta M} = 0$. Therefore, the equation

$$\mathsf{S}(\boldsymbol{\Delta Y_7} \oplus Y_7) \oplus \mathsf{S}(Y_7) = \mathsf{P}^{-1}\mathsf{L}^{-1}(\boldsymbol{\Delta \widetilde{R}} \oplus \boldsymbol{\Delta K_8} \oplus \boldsymbol{\Delta M})$$

can be solved row-by-row. We expect *an average* (see also Appendix A) $2^{128} = 2^{16 \cdot 8}$ solutions Y_7. The possible secret value M is calculated by knowing Y_7 and the round keys. The truth of each value M is checked on an arbitrary input-output pair (H, R).

The time complexity of the proposed method is

$$t = \underbrace{2^{128} \cdot 2^{64}}_{\text{"offline"}} + \underbrace{2^{112} \cdot 2^{128}}_{\text{"online"}} \approx 2^{240} \text{ operations.}$$

«Offline» and «Online» stages can be performed simultaneously. Hence, the memory is only used to store the possible values of $\boldsymbol{\Delta Y_7}$ and similar tables (no more than 2^{20} states). The described algorithm is probabilistic. We estimate the lower bound of the success probability as $1 - (1 - 2^{-112})^{q/2} \approx 1 - e^{-1} \approx 0.63$ with $q = 2^{113}$ chosen pairs (H, R).

5 Conclusion

In this paper we examine Streebog compression function as preudo-random function (PRF). Each of the two inputs (the previous state and the message block) can be used as a secret parameter and these two cases were considered.

We present two key-recovery algorithms for 7 rounds (of 12).

Setting	Rounds	Time	Memory	Data	Method
secret state	7	2^{421}	2^{354}	2^{64}	impossible polytopic
secret message	7	2^{240}	2^{20}	2^{113}	truncated differentials

The security proofs of many keyed hash-based cryptoalgorithms rely on PRF-properties of the underlying compression function. Our results demonstrate a great security margin of the Streebog 12-round compression function as a PRF in the above-mentioned secret-key settings. Thus, we have another yet informal argument that Streebog-based keyed algorithms are secure.

References

[1] *GOST R 34.11-2012 – National standard of the Russian Federation – Information technology – Cryptographic data security – Hash function*, 2012.

[2] Damgård I., "A Design Principle for Hash Functions", *LNCS*, CRYPTO 1989, **435**, ed. Brassar G., Springer, Heidelberg, 1990, 416–427.

[3] Merkle R., "One way Hash Functions and DES", *LNCS*, CRYPTO 1989, **435**, ed. Brassard G., Springer, Heidelberg, 1990, 428–446.

[4] Boneh D., Shoup V., "A Graduate Course in Applied Cryptography", 2020.

[5] Tiessen T., "Polytopic Cryptanalysis", *LNCS*, Advances in Cryptology – EUROCRYPT 2016, **9665**, ed. Fischlin M., Coron JS., Springer, Berlin, Heidelberg, 2016.

[6] Bellare M., "New Proofs for NMAC and HMAC: Security without Collision-Resistance", *LNCS*, Advances in Cryptology – CRYPTO 2006, **4117**, ed. Dwork C., Springer, Berlin, Heidelberg, April 2014.

[7] Guo J., Jean J., Leurent G., Peyrin T., Wang L., "The Usage of Counter Revisited: Second-Preimage Attack on New Russian Standardized Hash Function", *LNCS*, Selected Areas in Cryptography – SAC 2014, **8781**, ed. Joux A., Youssef A., Springer, Cham, 2014.

[8] AlTawy R., Youssef A. M., "Integral distinguishers for reduced-round Stribog", *Information Processing Letters*, **114** (2014).

[9] AlTawy R., Youssef A. M., "Preimage Attacks on reduced-round Stribog", *LNCS*, Progress in Cryptology – AFRICACRYPT 2014, **8469**, ed. Pointcheval D., Vergnaud D., Springer, Cham, 2014.

[10] AlTawy R., Kircanski A., Youssef A. M., "Rebound attacks on Stribog", *LNCS*, Information Security and Cryptology – ICISC 2013, **8565**, ed. Lee HS., Han DG., Springer, Cham, 2014.

[11] Lin D., Xu S., Yung M., "Cryptanalysis of the Round-Reduced GOST Hash Function", *LNCS*, Information Security and Cryptology. Inscrypt 2013., **8567**, Springer, Cham, 2014.

[12] Ma B., Li B., Hao R., Li X., "Improved cryptanalysis on reduced-round GOST and Whirlpool hash function", *LNCS*, Applied Cryptography and Network Security. ACNS 2014., **8479**, ed. Boureanu I., Owesarski P., Vaudenay S., Springer, Cham, 2014.

[13] Wang Z., Yu H., Wang X., "Cryptanalysis of GOST R Hash Function", *Information Processing Letters*, **114** (2014), 655–662.

[14] Kölbl S., Rechberger C., "Practical Attacks on AES-like Cryptographic Hash Functions", *LNCS*, Progress in Cryptology – LATINCRYPT 2014, **8895**, ed. Aranha D., Menezes A., Springer, Cham, 2014.

[15] Abdelkhalek A., AlTawy R., Youssef A. M., "Impossible Differential Properties of Reduced Round Streebog", *LNCS*, Codes, Cryptology, and Information Security. C2SI 2015, **9084**, ed. El Hajji S., Nitaj A., Carlet C., Souidi E., Springer, Cham, 2015, 274–286.

[16] Ma B., Li B., Hao R., Li X., "Improved (Pseudo) Preimage Attacks on Reduced-Round GOST and Grøstl-256 and Studies on Several Truncation Patterns for AES-like Compression Functions", *LNCS*, Advances in Information and Computer Security. IWSEC 2015, **9241**, ed. Tanaka K., Suga Y., Springer, Cham, 2015, 79–96.

[17] Rongjia Li, Chenhui Jin, Ruya Fan, "Improved Integral Distinguishers on Compression Function of GOST R Hash Function", *Computer Journal*, **62** (2019), 535–544.

[18] Tingting Cui, Wei Wang, Meiqin Wang, "Distinguisher on full-round compression function of GOST R", *Information Processing Letters*, **156** (2019).

[19] Chang D., Nandi M., "A Short Proof of the PRP/PRF Switching Lemma", *Cryptology ePrint Archive, Report 2008/078*, 2008.

[20] Knudsen L., "Truncated and Higher Order Differentials", 2nd International Workshop on Fast Software Encryption (FSE 1994), 1994, 196–211.

[21] Derbez P., Fouque P.-A., Jean J., "Improved Key Recovery Attacks on Reduced-Round AES in the Single-Key Setting", *LNCS*, Advances in Cryptology – EUROCRYPT 2013, **7881**, Springer, Berlin, Heidelberg, 2013, 371–387.

[22] Kazymyrov O., Kazymyrova V., "Algebraic Aspects of the Russian Hash Standard GOST R 34.11-2012", *Cryptology ePrint Archive, Report 2013/556*, 2013.

[23] Wagner D., "A Generalized Birthday Problem", *LNCS*, Advances in Cryptology – CRYPTO 2002, **2442**, ed. Yung M., Springer, Berlin, Heidelberg, 2002.

[24] Dinur I., Leurent G., "Improved Generic Attacks Against Hash-based MACs and HAIFA", *LNCS*, Advances in Cryptology – CRYPTO 2014, **8616**, ed. Garay J.A., Gennaro R., Springer, Berlin, Heidelberg, 2014.

[25] Mendel F., Rechberger C., Schläffer M., Søren S. Thomsen, "The Rebound Attack: Cryptanalysis of Reduced Whirlpool and Grøstl", *LNCS*, Fast Software Encryption. FSE 2009, **5665**, ed. Dunkelman O., Springer, Berlin, Heidelberg, 2009.

A Differential properties of Streebog's S-box

The differential distribution table (DDT) is defined as follows

$$\mathrm{DDT}[\boldsymbol{\Delta x}][\boldsymbol{\Delta y}] = \left|\{x : \mathsf{s}(x) \oplus \mathsf{s}(x \oplus \boldsymbol{\Delta x}) = \boldsymbol{\Delta y}\}\right|,$$

where $\mathsf{s} : \mathbb{F}_{2^8} \to \mathbb{F}_{2^8}$, $x, \boldsymbol{\Delta x}, \boldsymbol{\Delta y} \in \mathbb{F}_{2^8}$.

The distribution of the number of solutions for Streebog's S-box is shown in the table below.

Solutions	0	2	4	6	8	256
Number	38235	22454	4377	444	25	1

For random non-zero $\boldsymbol{\Delta x}, \boldsymbol{\Delta y} \in \mathbb{F}_{2^8} \backslash 0$ the probability that at least some solution exists is

$$p = \Pr\left(\left|\{x : \boldsymbol{\Delta y} = \mathsf{s}(x) \oplus \mathsf{s}(x \oplus \boldsymbol{\Delta x})\}\right| > 0\right) = \frac{22454 + 4377 + 444 + 25}{255^2}.$$

Let $\boldsymbol{\Delta x} \neq 0$, $\boldsymbol{\Delta y} \neq 0$, and it is also known that the equation

$$\mathsf{s}(x) \oplus \mathsf{s}(x \oplus \boldsymbol{\Delta x}) = \boldsymbol{\Delta y}$$

has a solution x. Then we get a conditional distribution of the number of solutions

$$\begin{pmatrix} 2 & 4 & 6 & 8 \\ \frac{22454}{27300} & \frac{4377}{27300} & \frac{444}{27300} & \frac{25}{27300} \end{pmatrix}.$$

The expected value of such a distribution (i.e., the average number of solutions provided that at least one solution exists) is

$$\frac{1}{27300}\left(2 \cdot 22454 + 4 \cdot 4377 + 6 \cdot 444 + 8 \cdot 25\right) = \frac{2^{16} - 2^8}{27300} = 2.39\ldots = z.$$

The case «$\mathsf{S}(\boldsymbol{\Delta HY_4} \oplus HY_4) \oplus \mathsf{S}(HY_4) = \boldsymbol{\Delta HZ_4}$»

We assume, that $\boldsymbol{\Delta HZ_4}$ is a random difference. We also know that $\boldsymbol{\Delta HZ_4}$ consisting only of non-zero bytes.

Each row in $\boldsymbol{\Delta HY_4}$ is also completely non-zero and belongs to a set of 255 elements.

The probability that a single byte matches is $p \approx 0.419$. Hence a row matches with a probability of $p^8 \approx 2^{-10}$.

The probability that among the allowed $\boldsymbol{\Delta HY_4[0, \cdot]}$ there is a suitable one $1 - (1 - p^8)^{255} \approx 2^{-2.2}$.

Therefore the probability for a match of all 8 rows equals to $2^{-2.2 \cdot 8} = 2^{-17.6}$.

Each pair $(\boldsymbol{\Delta HY_4}, \boldsymbol{\Delta HZ_4})$ for which the equation is solvable gives an average of $z^{64} \approx 2^{80.4}$ solutions.

We have $(2^{16} - 8 \cdot 255 - 1)^8 \approx 2^{127.6}$ possible values $\boldsymbol{\Delta HZ_4}$. As a result we obtain about

$$2^{127.6+80.4-17.6} = 2^{190.4}$$

valid states HY_4.

The case «$\mathsf{S}(\boldsymbol{\Delta Y_7} \oplus Y_7) \oplus \mathsf{S}(Y_7) = \boldsymbol{\Delta Z_7}$»

The case is similar to the previous one. We also assume, that $\boldsymbol{\Delta Z_7}$ is a random fully active difference.

Each row in $\boldsymbol{\Delta Y_7}$ belongs to a set of $u = (2^{16} - 1)$ elements.

We expect that about $u \cdot p^8 \approx 2^6$ suitable $\boldsymbol{\Delta Y_7[i, \cdot]}$ for each $i = 0, \ldots, 7$.

In total, we have about $(2^6)^8 = 2^{48}$ possible variants of $\boldsymbol{\Delta Y_7}$.

Thus, the average number of solutions Y_7 is equal to $z^{64} \cdot 2^{48} \approx 2^{128}$.

The assumptions and estimates presented in the Appendix were also experimentally verified using software from [14].

Nonlinearity of Bent Functions over Finite Fields

Vladimir Ryabov

NP «GST», Russia
Public Council of the Federal Treasury of the Russian Federation, Russia
4vryabov@gmail.com

Abstract

A function of n variables over a finite field of q elements is called maximally nonlinear if it has the greatest nonlinearity among all q-valued functions of n variables. It is proved that for $q > 2$ and even values n, a necessary condition for the maximum nonlinearity of a function is the absence of a linear manifold of dimension greater than or equal to $n/2$, on which its restriction would coincide with the restriction of some affine function. In accordance with it, functions from Maiorana-McFarland's and Dillon's families of bent functions are not maximally nonlinear. A new family of maximally nonlinear bent functions of degree from 2 to $\max\{2, (q-1)(n/2-1)\}$ with nonlinearity equal $(q-1)q^{n-1} - q^{\frac{n}{2}-1}$ is constructed.

Keywords: finite field, nonlinearity, bent function, maximally nonlinear function

1 Introduction

Let $\mathbf{F}_q$ be a finite field of q elements, where $q = p^m$, p is a prime number, m is a positive integer, and $\mathbf{F}_q^n$ is an n-dimensional vector space over the field $\mathbf{F}_q$, where n is a positive integer. We denote by P_q^n the set of all mappings of the space $\mathbf{F}_q^n$ into the field $\mathbf{F}_q$ or q-valued logic functions of n variables, and by A_q^n its subset of affine mappings. Since any function $a(\mathbf{x}) \in A_q^n$ can be uniquely represented by a polynomial over the field $\mathbf{F}_q$ of the form

$$a(x_1, \ \ldots, \ x_n) = a_0 \oplus a_1 \otimes x_1 \oplus \cdots \oplus a_n \otimes x_n, \tag{1}$$

where $a_0, \ a_1, \ \ldots, \ a_n \in \mathbf{F}_q$, $\oplus$ and $\otimes$ are addition and multiplication operations in $\mathbf{F}_q$,[1] we can use the vector representation of affine mappings by associating the function $a(\mathbf{x})$ with the vector $\alpha = (a_0, \ a_1, \ \ldots, \ a_n) \in \mathbf{F}_q^{n+1}$, where the coordinates $a_0, \ a_1, \ \ldots, \ a_n$ are the coefficients in the polynomial representation (1).

[1]By analogy with works [7, 8], the symbols $\oplus$ and $\otimes$ are used to denote operations in $\mathbf{F}_q$ in order to avoid coincidence with the designation of operations with real numbers and, conversely, to obtain match in the special case q = 2.

Taking the Hamming distance between functions in the space $\mathbf{F}_q^{q^n}$ as the proximity metric and denoting the distance between functions $f(\mathbf{x}) \in P_q^n$ and $a(\mathbf{x}) \in A_q^n$ as ρ_f^α we define the nonlinearity of the function $f(\mathbf{x})$ by the formula

$$N_f = \min_{\alpha \in \mathbf{F}_q^{n+1}} \rho_f^\alpha. \tag{2}$$

This parameter plays an important role in cryptography and coding theory. It is known from practice that high nonlinearity is one of the necessary conditions for the resistance of cryptosystems built using such functions to decryption methods. We will call functions from P_q^n with the largest value of nonlinearity maximally nonlinear and denote the class of such functions by MN_q^n.

The study of this class of Boolean functions began in the 1960s. As noted in [2], Soviet cryptographers contributed greatly to the research in this field. At the same time the first public article on this issue appeared only in 1976. Drawing on character theory, O.S. Rothaus for even values of n described the class of Boolean functions, which he called bent functions, with the maximum possible nonlinearity equal to $2^{n-1}-2^{\frac{n}{2}-1}$, and showed that this class coincides with the class MN_2^n [6]. For odd values of n, there are no Boolean bent functions. Nonlinearity equal to $2^{n-1} - 2^{\frac{n-1}{2}}$ is provided by the so-called "1-plateaued" or "near-bent" functions. By the beginning of the 1980s it was proved that for $n \leq 7$ these functions are maximally nonlinear.However, later in the scientific literature Boolean functions of 9 variables with greater nonlinearity were presented. Subsequently, numerous studies of Boolean bent functions appeared, as well as generalizations of the results of [6] to other discrete mappings, including functions over finite fields for $q > 2$.

A generalization of the concept of a bent function to the case of a residue ring $\mathbf{Z}_k$, which is a field for prime k, was presented in [5]. For an arbitrary finite field, such a generalization was first presented in 1994 by A.S. Ambrosimov. In [1] the definition of a q-valued bent function is given, all quadratic bent functions are described and their number is calculated (hereinafter q-valued bent functions are considered by us as in article [1], and the class of such functions of n variables is denoted by B_q^n). In contrast to the case of Boolean functions, q-valued bent functions for odd values of the field characteristic p also exist for odd values of n. In [1], the generalized Rothaus criterion was also proved, which states that $f(\mathbf{x}) \in B_q^n$ if and only if the derivative of the function $f(\mathbf{x})$ in the direction $\mathbf{c}$ given by the difference $f(\mathbf{x} \oplus \mathbf{c}) \ominus f(\mathbf{x})$, where $\ominus$ is the subtraction operations in $\mathbf{F}_q$, is a balanced function for $\forall \mathbf{c} \in \mathbf{F}_q^n \backslash \{\mathbf{0}\}$ (functions for which all nontrivial derivatives are

balanced are also called perfect nonlinear). Almost simultaneously, a similar definition of a bent function and a proof of the coincidence of the class of bent functions with the class of perfect nonlinear functions in the case of an arbitrary finite field appeared in [4][2]. At the same time, the issues of nonlinearity of bent functions were not considered in these papers.

For $q > 2$, not all bent functions are maximally nonlinear. As noted in [3], while retaining perfect nonlinearity associated with resistance to differential cryptanalysis methods, q-valued bent functions are not optimal when using linear methods associated with the nonlinearity in the sense of (2). In this regard, the problem of describing the nonlinearity of q-valued bent functions and identifying subclasses of bent functions with the highest degree of nonlinearity is urgent.

In [7], for $f(\mathbf{x}) \in P_q^n$, the following upper bound for nonlinearity was obtained

$$N_f \leq (q-1)q^{n-1} - q^{\frac{n}{2}-1}, \tag{3}$$

which, in particular, is also valid for all q-valued bent functions. In [8], for $n = 1$, estimate (3) was refined as follows:

$$N_f \leq q - 2. \tag{4}$$

We denote the class of functions Extended-Affine equivalent or **EA**-equivalent to a mapping $f(\mathbf{x}) \in P_q^n$ by $F_{\mathbf{EA}}(f)$. It is known that membership in bent functions is preserved for the class $F_{\mathbf{EA}}(f)$. In [8], it was shown that all functions from the class $F_{\mathbf{EA}}(f)$ have the same nonlinearity and therefore the property of maximum nonlinearity is also preserved.

It follows from the results of [8] that for $q > 2$ and even values of n, the set of quadratic q-valued bent functions of n variables is split into two classes of **EA**-equivalent functions. For functions of one class, the equality

$$N_f = (q-1)q^{n-1} - q^{\frac{n}{2}-1} \tag{5}$$

holds, and this class, taking into account inequality (3), consists of maximally nonlinear functions, while for functions of another class the equality

$$N_f = (q-1)(q^{n-1} - q^{\frac{n}{2}-1})$$

holds, and this class does not contain maximally nonlinear functions. For fields of odd characteristic and odd values of n, all quadratic q-valued bent functions of n variables have the same nonlinearity equal to

$$N_f = (q-1)q^{n-1} - q^{\frac{n-1}{2}},$$

[2]The definition of a q-valued bent function [1, 4] falls under the general definition of a bent function from a finite abelian group into a finite abelian group [9].

and are maximally nonlinear in the case $n = 1$.

Thus, for even values of n in [8], a criterion was proved: $f(\mathbf{x}) \in MN_q^n$ if and only if equality (5) is valid for the nonlinearity of the function. This criterion allows for $q > 2$, to obtain the necessary condition for the maximum nonlinearity of a q-valued function and to draw conclusions regarding a number of famous classes of bent functions that consist not only of quadratic functions.

2 Necessary condition for maximum nonlinearity and results for bent functions

Let $f|_R$ be the restriction of a function $f(\mathbf{x}) \in P_q^n$ to a subset $R \subseteq \mathbf{F}_q^n$. For such restrictions we use the Hamming distance in the space $\mathbf{F}_q^{|R|}$ as the proximity metric and denote by $\rho^{\alpha}_{f|_R}$ the distance between the restrictions $f|_R$ and $a|_R$, where $a(\mathbf{x}) \in A_q^n$. Let's further use the auxiliary parameters $\delta^{\alpha}_{f|_R} = \frac{q-1}{q} - \frac{\rho^{\alpha}_{f|_R}}{|R|}$ introduced in [7]. In the case $R = \mathbf{F}_q^n$, we have $\delta^{\alpha}_f = \frac{q-1}{q} - \frac{\rho^{\alpha}_f}{q^n}$.

A special case of a subset is a linear manifold of the vector space $\mathbf{F}_q^n$. Let $\mathfrak{M}_q^n(r)$ denote the set of all r-dimensional linear manifolds in the space $\mathbf{F}_q^n$, where $0 \le r \le n$. In the case $M \in \mathfrak{M}_q^n(r)$, the parameter $\delta^{\alpha}_{f|_M}$ satisfies the formula $\delta^{\alpha}_{f|_M} = \frac{q-1}{q} - \frac{\rho^{\alpha}_{f|_M}}{q^r}$.

Theorem 1. *Let $f(\mathbf{x}) \in MN_q^n$, where $q > 2$ and n is even. Then in the space $\mathbf{F}_q^n$ there is no linear manifold of dimension greater than or equal to $n/2$ on which the restriction of the function $f(\mathbf{x})$ coincides with the restriction of some affine function.*

Proof. Let's prove the theorem by contradiction.

If there exists a manifold of dimension greater than $n/2$ on which the restriction of the function $f(\mathbf{x})$ coincides with the restriction of the affine function, then any of its $n/2$ - dimensional submanifolds also possess this property, which allows us to restrict ourselves to considering the latter.

Let $M \in \mathfrak{M}_q^n(r)$ be a manifold on which the restriction $f|_M$ coincides with the restriction $a|_M$, where $a(\mathbf{x}) \in A_q^n$. Then the relations $\rho^{\alpha}_{f|_M} = 0$ and $\delta^{\alpha}_{f|_M} = \frac{q-1}{q}$ are valid.

Like any linear manifold of dimension $n/2$, the manifold M is a collection of solutions of a system of $n/2$ linear equations over the field $\mathbf{F}_q$ of the form

$$\begin{cases} b_{1,0} \oplus b_{1,1} \otimes x_1 \oplus \cdots \oplus b_{1,n} \otimes x_n = 0, \\ \cdots \\ b_{n/2,0} \oplus b_{n/2,1} \otimes x_1 \oplus \cdots \oplus b_{n/2,n} \otimes x_n = 0 \end{cases} \tag{6}$$

with a matrix of coefficients of rank equal to $n/2$. The results of [7] for $\forall \alpha \in \mathbf{F}_q^{n+1}$ imply the relation

$$\delta^{\alpha}_{f|_M} = \sum_{c_1,\ldots,c_{n/2}} \delta_f^{\alpha \oplus c_1 \otimes \beta_1 \oplus \cdots \oplus c_{n/2} \otimes \beta_{n/2}}, \tag{7}$$

where $\beta_i = (b_{i,0}, b_{i,1}, \ldots, b_{i,n}) \in \mathbf{F}_q^{n+1}$ from (6) for $i = 1, \ldots, n/2$, and $c_1, \ldots, c_{n/2}$ are all possible different sets of $n/2$ elements of the field $\mathbf{F}_q$.

The right-hand side of (7) contains $q^{\frac{n}{2}}$ terms and therefore there is a term with coefficients $\hat{c_1}, \ \ldots, \ \hat{c_{n/2}}$, for which the following inequality holds:

$$\delta_f^{\alpha \oplus \hat{c_1} \otimes \beta_1 \oplus \cdots \oplus \hat{c_{n/2}} \otimes \beta_{n/2}} \geq (q-1)q^{-\frac{n}{2}-1}. \tag{8}$$

From (8) follows the inequality

$$\rho_f^{\alpha \oplus \hat{c_1} \otimes \beta_1 \oplus \cdots \oplus \hat{c_{n/2}} \otimes \beta_{n/2}} \leq (q-1)q^{n-1} - (q-1)q^{\frac{n}{2}-1},$$

which for $q > 2$ contradicts the hypothesis of the theorem, since equality (5) holds for the maximally nonlinear function.

Theorem is proved. □

Remark 1. *In the case of Boolean functions, Theorem 1 does not work. Indeed, Boolean bent functions based on the Mayorana-McFarland's construction* $f(\mathbf{x}) = \langle \mathbf{x}', \pi(\mathbf{x}'') \rangle \oplus g(\mathbf{x}'')$, *where* $\mathbf{x}' = (x_1, \ \ldots, \ x_{n/2})$, $\mathbf{x}'' = (x_{n/2+1}, \ \ldots, \ x_n) \in \mathbf{F}_2^{n/2}$, π *is an arbitrary substitution on the set* $\mathbf{F}_2^{n/2}$, $\langle *, * \rangle$ *is the scalar product of vectors in the space* $\mathbf{F}_2^{n/2}$, *and* g *is an arbitrary function from* $P_2^{n/2}$, *are maximally nonlinear functions with* $N_f = 2^{n-1} - 2^{\frac{n}{2}-1}$. *However, their restrictions for any fixation of the variables* $x_{n/2+1}, \ \ldots, \ x_n$ *constants coincide with the restrictions of affine functions. Boolean bent functions based on the Dillon's construction, obtained by summing the characteristic functions of* 2^{n-1} *or* $2^{n-1}+1$ *linear subspaces of the space* $\mathbf{F}_2^n$ *of dimension* $n/2$ *with pairwise intersection only along the zero vector*[3], *are also maximally nonlinear and at the same time certainly contain subspaces of dimension* $n/2$, *on which the function is equal to a unit constant.*

[3]These sets of bent functions are also called Partial Spreads and are denoted PS^- and PS^+ respectively.

In [3], for even values of n, generalizations Mayorana-McFarland's and Dillon's constructions to the case of functions of q-valued logic are presented.

The Mayorana-McFarland's construction for q-valued bent functions of n variables has a form similar to the Boolean case

$$f(\mathbf{x}) = \langle \mathbf{x}', \pi(\mathbf{x}'') \rangle \oplus g(\mathbf{x}''), \tag{9}$$

where $\mathbf{x}' = (x_1, \ \ldots, \ x_{n/2})$, $\mathbf{x}'' = (x_{n/2+1}, \ \ldots, \ x_n) \in \mathbf{F}_q^{n/2}$, π is an arbitrary substitution on the set $\mathbf{F}_q^{n/2}$, $\langle *, \ * \rangle$ is the scalar product of vectors in the space $\mathbf{F}_q^{n/2}$, and g is an arbitrary function from $P_q^{n/2}$.

Using the correspondence of the vector space $\mathbf{F}_q^{n/2}$ to the field $\mathbf{F}_{q^{n/2}}$, the Dillon's construction for q-valued bent functions of n variables are defined as follows

$$f(\mathbf{x}) = h(\mathbf{x}' \otimes (\mathbf{x}'')^{q^{\frac{n}{2}-2}}), \tag{10}$$

where $\mathbf{x}', \ \mathbf{x}'' \in \mathbf{F}_{q^{n/2}}$, $\otimes$ and $(*)^*$ are operations of multiplication and exponentiation in the field $\mathbf{F}_{q^{n/2}}$, respectively, and the mapping $h : \mathbf{F}_{q^{n/2}} \to \mathbf{F}_q$ is balanced function.

Corollary 1. *Let $q > 2$, n is even, $f(\mathbf{x}) \in B_q^n$ and $f(\mathbf{x})$ represented in the form* (9)*, that is, it belongs to the Mayorana-MacFarland's family. Then $f(\mathbf{x}) \notin MN_q^n$.*

To prove the corollary, it suffices to note that for any fixation of the variables $x_{n/2+1}, \ \ldots, \ x_n$ by constants, the restriction of the function $f(\mathbf{x})$ coincides with the restriction of the affine function and use Theorem 1.

Corollary 2. *Let $q > 2$, n is even, $f(\mathbf{x}) \in B_q^n$ and $f(\mathbf{x})$ represented in the form* (10)*, that is, it belongs to the Dillon's family. Then $f(\mathbf{x}) \notin MN_q^n$.*

Setting $x_1 = \ \ldots \ = x_{n/2} = 0$ or $x_{n/2+1} = \ \ldots \ = x_n = 0$, we obtain subspaces of the space $\mathbf{F}_q^n$ of dimension $n/2$, on which the restriction of the function $f(\mathbf{x})$ coincides with the constant function $h(\mathbf{0})$, and, hence, by Theorem 1, the function $f(\mathbf{x})$ is not maximally nonlinear.

3 Construction of a family of maximally nonlinear bent functions

The idea underlying the Mayorana-McFarland construction allows for $q > 2$ and even values of $n \geq 4$ to specify maximally nonlinear q-valued bent functions of degree greater than two[4].

[4]For $n = 4$ the condition $q > 3$ must be satisfied.

Theorem 2. *Let $q > 2$, n is even, $\mathbf{x}' = (x_1, x_2) \in \mathbf{F}_q^2$, $\mathbf{x}'' = (x_3, \ldots, x_{n/2+1})$, $\mathbf{x}''' = (x_{n/2+2}, \ldots, x_n) \in \mathbf{F}_q^{n/2-1}$, π is an arbitrary substitution on the set $\mathbf{F}_q^{n/2-1}$, $\langle *, * \rangle$ is the scalar product of vectors in the space $\mathbf{F}_q^{n/2-1}$, and g is an arbitrary function from $P_q^{n/2-1}$. Then*

a) for fields of even characteristic with respect to the function

$$f(\mathbf{x}) = x_1 \otimes x_2 \oplus x_1^2 \oplus c \otimes x_2^2 \oplus \langle \mathbf{x}'', \pi(\mathbf{x}''') \rangle \oplus g(\mathbf{x}'''), \tag{11}$$

where c is a free term of an irreducible polynomial $x^2 \oplus x \oplus c$,[5] the statements $f(\mathbf{x}) \in B_q^n$ and $f(\mathbf{x}) \in MN_q^n$ are true;

b) for fields of odd characteristic with respect to the function

$$f(\mathbf{x}) = x_1^2 \ominus d \otimes x_2^2 \oplus \langle \mathbf{x}'', \pi(\mathbf{x}''') \rangle \oplus g(\mathbf{x}'''), \tag{12}$$

where d is a quadratic nonresidue[6], the statements $f(\mathbf{x}) \in B_q^n$ and $f(\mathbf{x}) \in MN_q^n$ are true.

Proof. Let's consider the case of fields of even characteristic. For $n = 2$, the proof of the theorem is obvious, since expression (11) reduces to the quadratic form $x_1 \otimes x_2 \oplus x_1^2 \oplus c \otimes x_2^2$, and from [8] it follows that when c is a free term of a irreducible polynomial $x^2 \oplus x \oplus c$, this form is a maximally nonlinear bent function of two variables.

Now let $n \geq 4$. Let's prove that $f(\mathbf{x})$ is a bent function. As stated above, $f'(\mathbf{x}') = x_1 \otimes x_2 \oplus x_1^2 \oplus c \otimes x_2^2 \in B_q^2$. The function $f''(\mathbf{x}'', \mathbf{x}''') = \langle \mathbf{x}'', \pi(\mathbf{x}''') \rangle \oplus g(\mathbf{x}''')$ belongs to the Mayorana-McFarland family and therefore $f''(\mathbf{x}'', \mathbf{x}''') \in B_q^{n-2}$. It follows from [4] that the sum of bent functions of independent variables is a bent function defined on the totality of these variables. Therefore, $f(\mathbf{x}) = f'(\mathbf{x}') \oplus f''(\mathbf{x}'', \mathbf{x}''') \in B_q^n$.

Let's proceed to the proof of the maximum nonlinearity of $f(\mathbf{x})$. Fixing $n/2 - 1$ variables as constants

$$\begin{cases} x_{n/2+2} = b_{n/2+2}, \\ \ldots \\ x_n = b_n, \end{cases} \tag{13}$$

we obtain a subfunction $f_\beta(\mathbf{x}', \mathbf{x}'') \in P_q^{n/2+1}$ of the form

$$f_\beta(\mathbf{x}', \mathbf{x}'') = x_1 \otimes x_2 \oplus x_1^2 \oplus c \otimes x_2^2 \oplus \langle \mathbf{x}'', \pi(\beta) \rangle \oplus g(\beta),$$

[5]The equivalent condition in terms of the absolute trace is determined by the equality $Tr(c) = c \oplus c^2 \oplus \cdots \oplus c^{2^{m-1}} = 1$

[6]An element $d \in \mathbf{F}_q \backslash \{0\}$ is called a square nonresidue if $\nexists a \in \mathbf{F}_q$ such that the equality $a^2 = d$ holds.

where $\beta = (b_{n/2+2},\ \ldots,\ b_n) \in \mathbf{F}_q^{n/2-1}$.

For an affine function $\hat{a}(\mathbf{x}', \mathbf{x}'') = a_0 \oplus a_1 \otimes x_1 \oplus \cdots \oplus a_{n/2+1} \otimes x_{n/2+1} \in A_q^{n/2+1}$, which corresponds to the vector $\hat{\alpha} = (a_0, a_1, a_2, a_3, \ldots, a_{n/2+1}) = (a_0, a_1, a_2, \alpha'') \in \mathbf{F}_q^{n/2+2}$, in the case $\alpha'' \neq \pi(\beta)$ the sum $f_\beta(\mathbf{x}', \mathbf{x}'') \oplus \hat{a}(\mathbf{x}', \mathbf{x}'')$ has independent linear terms and therefore it is balanced. Then for $q^{n/2+2} - q^3$ corresponding parameters of the subfunction $f_\beta(\mathbf{x}', \mathbf{x}'')$ the equalities $\rho_{f_\beta}^{\hat{\alpha}} = (q-1)q^{\frac{n}{2}}$ hold. In the case $\alpha'' = \pi(\beta)$, this sum is a function that essentially depends only on two variables x_1 and x_2. From [8] it follows that in this case for q^3 remaining parameters the inequalities $\rho_{f_\beta}^{\hat{\alpha}} \geq (q-1)q^{\frac{n}{2}} - q^{\frac{n}{2}-1}$ hold, since $x_1 \otimes x_2 \oplus x_1^2 \oplus c \otimes x_2^2 \oplus a_1 \otimes x_1 \oplus a_2 \otimes x_2 \oplus g(\beta) \oplus a_0 \in MN_q^2$.

Let $M_\beta \in \mathfrak{M}_q^n(n/2+1)$ be the linear manifold defined by the same fixation of variables (13). It follows from [7] that $q^{n/2+2}$ parameters $\rho_{f_\beta}^{\hat{\alpha}}$ of the subfunction $f_\beta(\mathbf{x}', \mathbf{x}'')$, by multiplication, define q^{n+1} parameters $\rho_{f|_{M_\beta}}^{\alpha}$ of the restriction $f|_{M_\beta}$, namely, for all possible sets of $n/2-1$ elements of the field $c_{n/2+2}, \ldots, c_n$ the following relations hold

$$\rho_{f|_{M_\beta}}^{(\hat{\alpha},0,\ldots,0)\oplus c_{n/2+2}\otimes\beta_{n/2+2}\oplus c_n\otimes\beta_n} = \rho_{f_\beta}^{\hat{\alpha}},$$

where $(\hat{\alpha}, 0, \ldots, 0) \in \mathbf{F}_q^{n+1}$, and also for $i = n/2+2, \ldots, n$ the vector $\beta_i = (b_i, 0, \ldots, 0, 1, 0, \ldots, 0) \in \mathbf{F}_q^{n+1}$ and its $(i+1)$-th coordinate is 1. Thus, for $\forall \alpha \in \mathbf{F}_q^{n+1} \backslash \{(a_0, a_1, a_2, \pi(\beta), \alpha''')\}$, where $\alpha''' = (a_{n/2+2}, \ldots, a_n) \in \mathbf{F}_q^{n/2-1}$, the equalities $\rho_{f|_{M_\beta}}^{\alpha} = (q-1)q^{\frac{n}{2}}$ hold, then while for $\forall \alpha \in \{(a_0, a_1, a_2, \pi(\beta), \alpha''')\}$ the inequalities $\rho_{f|_{M_\beta}}^{\alpha} \geq (q-1)q^{\frac{n}{2}} - q^{\frac{n}{2}-1}$ hold.

Going through all possible $\beta \in \mathbf{F}_q^{n/2-1}$ we obtain $q^{n/2-1}$ pairwise disjoint linear manifolds for which the relation

$$\mathbf{F}_q^n = \bigcup_{\beta \in \mathbf{F}_q^{n/2-1}} M_\beta$$

holds. Since π is a substitution on $\mathbf{F}_q^{n/2-1}$, it is easy to see that $\forall \alpha = (a_0, a_1, a_2, \alpha'', \alpha''') \in \mathbf{F}_q^{n+1}$ the following relations chain holds

$$\rho_f^\alpha = \sum_{\beta \in \mathbf{F}_q^{n/2-1}} \rho_{f|_{M_\beta}}^{\alpha} = \sum_{\beta \in \mathbf{F}_q^{n/2-1} \backslash \pi^{-1}(\alpha'')} \rho_{f|_{M_\beta}}^{\alpha} + \rho_{f|_{M_{\pi^{-1}(\alpha'')}}}^{\alpha} \geq$$
$$\geq (q^{\frac{n}{2}-1} - 1)(q-1)q^{\frac{n}{2}} + (q-1)q^{\frac{n}{2}} - q^{\frac{n}{2}-1} = (q-1)q^{n-1} - q^{\frac{n}{2}-1}.$$

Taking into account inequality (3) and the criterion of maximum nonlinearity, based on equality (5), we obtain that $f(\mathbf{x}) \in MN_q^n$.

The proof of the second part of the theorem in the case of a field of odd characteristic and a function of the form (12) is carried out similarly, taking into account that it follows from [8] that, for a quadratic nonresidue d, the quadratic form $x_1^2 \ominus d \otimes x_2^2$ is a maximally nonlinear bent function of two variables.

Theorem is proved. □

Remark 2. *In a similar way to the proof of Theorem 2, we can show that for $q > 2$ and n is odd, for the function $f(\mathbf{x}) = x_1^2 \oplus \langle \mathbf{x}'', \pi(\mathbf{x}''') \rangle \oplus g(\mathbf{x}''')$, where $\mathbf{x}'' = (x_2, \ldots, x_{(n+1)/2})$, $\mathbf{x}''' = (x_{(n+3)/2}, \ldots, x_n) \in \mathbf{F}_q^{(n-1)/2}$, π is an arbitrary substitution on the set $\mathbf{F}_q^{(n-1)/2}$, $\langle *, * \rangle$ is the scalar product of vectors in the space $\mathbf{F}_q^{(n-1)/2}$, and g is an arbitrary function from $P_q^{(n-1)/2}$, the following inequality holds*

$$N_f \geq (q-1)q^{n-1} - q^{\frac{n-1}{2}},$$

and for $n = 1$, as follows from (4), $f(\mathbf{x})$ is a maximally nonlinear function. In the case of a field with an odd characteristic, $f(\mathbf{x})$ is a bent function, while for a field of an even characteristic, it is a balanced function and not a bent function.

Remark 3. *The families of functions constructed in Theorem 2 and as a result of Remark 2 can be extended by adding* **EA***-equivalent functions from P_q^n.*

Thus, Theorem 2, taking into account Remark 3, makes it possible for $q > 2$ to define a wide class of maximally nonlinear bent functions of q-valued logic from an even number of variables of arbitrary degree from 2 to $\max\{2, (q-1)(n/2-1)\}$. It does not overlap with the famous classes of bent functions listed above, since all of its functions are highly nonlinear.

4 Conclusion

The results obtained here confirm that for $q > 2$ and even values n, some famous families of q-valued bent functions do not possess the property of maximum nonlinearity. At the same time a new family of bent functions over finite fields is constructed, which are both perfect nonlinear and maximally nonlinear.

For odd values of n, a family of q-valued functions with a sufficiently high degree of nonlinearity is indicated. In the case of fields of odd characteristic, this family belongs to the class of bent functions, and for fields of even characteristic, it belongs to the class of balanced functions.

References

[1] Ambrosimov A. S., “Properties of bent functions of q-valued logic over finite fields”, *Discrete Mathematics and Applications*, **4**:4 (1994), 341–350.

[2] Glukhov M. M., “On the approximation of discrete functions by linear functions”, *Mathematical Aspects of Cryptography*, **7**:4 (2016), 29–50, In Russian.

[3] Carlet C., Ding C., “Highly nonlinear mappings”, *Journal of Complexity*, **20**:2-3 (2004), 205–244.

[4] Coulter, R. S., Matthews, R. W., “Bent polynomials over finite fields”, *Bulletin of The Australian Mathematical Society*, **56** (1997), 429–437.

[5] Kumar, P. V., Scholtz, R. A., Welch, L. R., “Generalized bent functions and their properties”, *Journal of Combinatorial Theory, Series A*, **40**:1 (1985), 90–107.

[6] Rothaus O. S., “On "bent" functions”, *Journal of Combinatorial Theory, Series A*, **20**:3 (1976), 300–305.

[7] Ryabov V. G., “On the approximation of restrictions of q-valued logic functions to linear manifolds by affine analogs”, *Discrete Mathematics*, **32**:4 (2020), 89–102, In Russian.

[8] Ryabov V. G., “Maximally nonlinear functions over finite fields”, *Discrete Mathematics*, **33**:1 (2021), 47–63, In Russian.

[9] Solodovnikov V. I., “Bent functions from a finite abelian group into a finite abelian group”, *Discrete Mathematics and Applications*, **12**:2 (2002), 111–126.

On Some Properties of the Curvature and Nondegeneracy of Boolean Functions

Reynier Antonio de la Cruz Jiménez

Institute of Cryptography, Havana University, Cuba
djr.antonio537@gmail.com

Abstract

The present article is concerned with the problem of obtaining exact formulas and bounds for the curvature (i.e., the sums of modules of Walsh coefficients) and nondegeneracy (parameter related to the resistance of a Boolean function against certain method of analysis) of some classes of cryptographic Boolean functions. Moreover, for these classes we determinate other relevant cryptographic parameters as nonlinearity, algebraic degree and distance to linear structures. Also, we extend the curvature parameter to the world of S-Boxes and for various nonlinear components of actual symmetric cryptographic algorithms we investigate the behavior of some parameters and matrices connected with the curvature of Boolean functions.

Keywords: Boolean functions, Walsh coefficients, filtering generators, combining generators, curvature, nondegeneracy, S-Box, block ciphers

Introduction

Boolean functions are an inseparable class of functions in modern Cryptography that play crucial roles in the combiner and filter models of stream ciphers systems and determine the most critical properties of the so-called Substitution Boxes (S-Boxes), which very often are included in the design of block ciphers and hash functions to provide nonlinear relationship between the input bits and the output bits, ensuring what Shannon called confusion.

In this work we continue the research line started in [6], considering the problem of obtaining exact formulas and bounds for the sums of modules of Walsh coefficients of some classes of cryptographic Boolean functions. The sums of modules of Walsh coefficients originally appeared in [14] during the study of the properties of binary filtering generator when estimating the frequencies of elements in the segments of output sequences, was also investigated in [6, 16] for various Boolean functions and used in [15] for obtaining

estimates of the number of solutions of nonlinear systems of equations in the case when the arguments of functions in these system, are obtained by using linear recurrent sequences over Galois rings. Recently, it has been appeared again in [19] when considering Boolean functions as points on the hypersphere in the Euclidean space, where the authors of this work suggested calling it — curvature of a Boolean function, due to its particular geometric meaning. In what follows, in order to unify the same concept, we shall support this term when referring to the sum of modules of Walsh coefficients.

The present article is mainly devoted to the study of the curvature and nondegeneracy (resistance to a method of analysis based on algebraically degenerate approximations [1]) of some cryptographic classes of Boolean functions and other significant cryptographic properties of these classes. Also, we extend the notion of the curvature to the world of S-Boxes and for several nonlinear components of actual symmetric cryptographic primitives we investigate some parameters connected with the curvature of Boolean functions.

Our work is structured as follows: We begin with preliminaries in Section 1, providing necessary notations and concepts about Boolean functions. In Section 2, we find the exact values and bounds of the curvature and nondegeneracy of some classes of Boolean functions and for these classes we also determine other relevant parameters such as non-linearity, algebraic degree and distance to linear structures. The variation of the curvature when changing randomly one or multiples values in the output of a Boolean function is analysed in Section 3. In Section 4, we extend the curvature parameter to n-bit S-Boxes and the behavior of this parameter when studing some concrete 8-bit S-Boxes is analysed. The article is concluded in Section 5.

1 Preliminaries

Let $\mathbb{F}_2$ be a finite field of two elements. For any $n \in \mathbb{N}$ we denote by $\mathbb{F}_2^n = \mathbb{F}_2 \times \cdots \times \mathbb{F}_2$, the vector space of dimension n with the components from the field $\mathbb{F}_2$, let $\mathbf{0} = (0, 0, \ldots, 0)$ be the null vector of $\mathbb{F}_2^n$ and by $\oplus$ we denote the addition operation of $\mathbb{F}_2^n$. A *Boolean function* of n variables is a mapping from $\mathbb{F}_2^n$ to $\mathbb{F}_2$. The set of all Boolean functions with n variables is denoted by $\mathcal{F}_n = \{f \colon \mathbb{F}_2^n \to \mathbb{F}_2\}$ and it is well-known that $\mathcal{F}_n$ is a linear vector space over $\mathbb{F}_2$. The *indicator function*, denoted by δ_a, is defined as follows, $\delta_a(x) = 1$, if $x = a$ and $\delta_a(x) = 0$, when $x \neq a$. *The Hamming weight* $w_H(x)$ of a binary vector $x \in \mathbb{F}_2^n$ being the number of its nonzero coordinates (i.e., the size of $\mathsf{supp}(x) = \{i \in \{1, 2, \ldots, n\} : x_i \neq 0\}$, the *support of vector* x), the *Hamming weight* $w_H(f)$ of a Boolean function $f \in \mathcal{F}_n$ is the size of $\mathsf{supp}(f) =$

$\{x \in \mathbb{F}_2^n : f(x) \neq 0\}$, the *support of function* f. The *distance* $\mathsf{dist}(f,g)$ between functions $f, g \in \mathcal{F}_n$ is the value of $w_H(f \oplus g)$. The *value vector* v_f of a Boolean function f is the vector $(f(0,\ldots,0),\ldots,f(1,\ldots,1))$ of length 2^n consisting of the values of f on all possible inputs in the lexicographic order. Every function $f \in \mathcal{F}_n$ has a unique polynomial representation over $\mathbb{F}_2$, of the form $f(x_1,\ldots,x_n) = \bigoplus_{I\subseteq\{1,\ldots,n\}} a_I\Big(\prod_{i\in I} x_i\Big)$, which very often is called the algebraic normal form (in brief the ANF) or Zhegalkin polynomial of f. The *algebraic degree* of a Boolean function f with n variables, denoted by $d_{alg}(f)$, is defined as $d_{alg}(f) = \max\{\#I \,|\, a_I \neq 0\}$, where $\#I$ denotes the size of I with the convention that the zero function has algebraic degree 0 (see, [5, p. 35]). For $x, y \in \mathbb{F}_2^n$ the *scalar product* of x and y is defined as

$$\langle x, y\rangle = x \cdot y^{\mathsf{T}} = \bigoplus_{i=1}^{n} x_i y_i \in \mathbb{F}_2.$$

The set $\left\{ \langle a, x\rangle \oplus b \;\middle|\; a \in \mathbb{F}_2^n, b \in \mathbb{F}_2 \right\}$ of *affine* Boolean functions in n variables is denoted as $\mathcal{A}_n$. For a Boolean function $f \in \mathcal{F}_n$ its *nonlinearity*, denoted by $nl(f)$, is the Hamming distance from the set of all affine functions $\mathcal{A}_n$:

$$nl(f) = \mathsf{dist}(f, \mathcal{A}_n) = \min_{l\in\mathcal{A}_n} \mathsf{dist}(f, \mathcal{A}_n).$$

When analysing the most relevant cryptographic properties of Boolean functions the Walsh–Hadamard transform is frequently used. The *Walsh–Hadamard transform* of a Boolean function $f \in \mathcal{F}_n$ is a function $\mathcal{W}_f \colon \mathbb{F}_2^n \to \mathbb{Z}$ such that

$$\mathcal{W}_f(u) = \sum_{x\in\mathbb{F}_2^n} (-1)^{f(x)\oplus\langle u,x\rangle}, u \in \mathbb{F}_2^n. \tag{1}$$

The value $\mathcal{W}_f(u)$ is called the Walsh–Hadamard coefficients (or Walsh coefficients). The nonlinearity of $f \in \mathcal{F}_n$ can be evaluated using its Walsh coefficients as follows:

$$nl(f) = 2^{n-1} - \frac{1}{2}\max_{u\in\mathbb{F}_2^n} |\mathcal{W}_f(u)|. \tag{2}$$

The Boolean function $f \in \mathcal{F}_n$ is called a *Bent function* if $|\mathcal{W}_f(u)| = 2^{\frac{n}{2}}$ for any $u \in \mathbb{F}_2^n$. The subset of $\mathcal{F}_n$ containing all Bent function is denoted by $\mathcal{B}_n$.

The *autocorrelation function* of a Boolean function $f \in \mathcal{F}_n$ is a function $\Delta_f \colon \mathbb{F}_2^n \to \mathbb{Z}$ such that

$$\Delta_f(u) = \sum_{x\in\mathbb{F}_2^n} (-1)^{f(x)\oplus f(x\oplus u)}, u \in \mathbb{F}_2^n. \tag{3}$$

It is well-known (see, for example, [18, p. 97]) that for any $f \in \mathcal{F}_n$ and $u \in \mathbb{F}_2^n$ the following equality holds

$$\sum_{v \in \mathbb{F}_2^n} \Delta_f(v)(-1)^{\langle u,v \rangle} = \mathcal{W}_f^2(u). \tag{4}$$

By applying in (4) the inverse Fourier transform formula (see, for example, [5, p. 59]), we have

$$\Delta_f(u) = \frac{1}{2^n} \sum_{v \in \mathbb{F}_2^n} \mathcal{W}_f^2(v)(-1)^{\langle u,v \rangle}. \tag{5}$$

The *space of linear structures* (see, [12]) for a Boolean function $f \in \mathcal{F}_n$ is the following set:

$$\mathsf{L}_f = \{u \in \mathbb{F}_2^n \,|\, \forall v \in \mathbb{F}_2^n \;\; f(v \oplus u) = f(v) \oplus \epsilon_u, \epsilon_u \in \mathbb{F}_2\}.$$

In this context, a vector $\mathbf{0} \neq u \in \mathbb{F}_2^n$ is a linear structure for $f \in \mathcal{F}_n$ if and only if the function $f'_u(v) = f(v \oplus u) \oplus f(v)$ is constant on $\mathbb{F}_2^n$. Nonlinear transformations used in block ciphers should have no nonzero linear structure [10] and the existence of nonzero linear structures, for the Boolean functions implemented in filtering and combining generators, is a potential weakness and is better avoided.

The *distance to linear structures* (see, [12]) of any n-variable Boolean function f, denoted by $\mathsf{ls}(f)$, can be calculated as follows

$$\mathsf{ls}(f) = 2^{n-2} - \frac{1}{4} \max_{\mathbf{0} \neq u \in \mathbb{F}_2^n} |\Delta_f(u)|. \tag{6}$$

The smaller parameter $\max_{\mathbf{0} \neq u \in \mathbb{F}_2^n} |\Delta_f(u)|$, the better the cryptographic quality of the Boolean function f.

Nondegeneracy of a function $f \in \mathcal{F}_n$, denoted by $\mathsf{nd}(f)$, is defined as the distance between f and the set of all algebraically degenerate[a] Boolean functions and can be evaluated (see, for example, [12]) using the autocorrelation of f as follows

$$\mathsf{nd}(f) = 2^{n-2} - \frac{1}{4} \max_{\mathbf{0} \neq u \in \mathbb{F}_2^n} \Delta_f(u). \tag{7}$$

As stated at beginning, the parameter $\mathsf{nd}(f)$ is closely related to the resistance offered by f against a method of analysis based on algebraically

[a] A function $f \in \mathcal{F}_n$ is called algebraically degenerate if there are $g \in \mathcal{F}_k$ and a binary $(k \times n)$-matrix D for some $k < n$ such that $f(u) = g\big(D \cdot u^{\mathsf{T}}\big)$ for all $u \in \mathbb{F}_2^n$.

degenerate approximations (see, [1]). The higher is $\mathsf{nd}(f)$, the better is the contribution of f to the resistance to this method.

The *curvature* of a Boolean function $f \in \mathcal{F}_n$, denoted by $\mathsf{curv}(f)$, is defined as:

$$\mathsf{curv}(f) = \sum_{u \in \mathbb{F}_2^n} |\mathcal{W}_f(u)|. \tag{8}$$

As showed in [6], the curvature of $f \in \mathcal{F}_n$ has the following bounds

$$2^n \leq \mathsf{curv}(f) \leq 2^{\frac{3n}{2}}, \tag{9}$$

where the lower bound becomes an equality if and only if $f \in \mathcal{A}_n$ and the upper bound is achieved only when $f \in \mathcal{B}_n$. The curvature parameter (which is affine invariant) is a very useful tool for characterizing "how close" is a Boolean function $f \in \mathcal{F}_n$ to being linear (or Bent) and in some sense this parameter can indicate some insight about the nonlinearity of f.

2 The curvature and nondegeneracy of some classes of Boolean functions

In this section, the main attention is paid to the calculation of the curvature and nondegeneracy of some classes of Boolean functions. For these classes, we also determine other parameters having a crucial significance in Cryptography such as non-linearity, algebraic degree and distance to linear structures.

2.1 Class of balanced functions with maximal algebraic degree

For an even natural number $n \geq 4$ and $\varphi \in \mathcal{B}_{n-2}$, a class of Balanced functions f_φ which its ANF contains a single term $x_1 x_2 \cdots x_{n-1}$ can be defined as follows

$$f_\varphi(x_1, \ldots, x_n) = x_1 x_2 \cdots x_{n-1} \oplus \varphi(x_1, \ldots, x_{n-2}) \oplus x_n, \tag{10}$$

this class was studied in [6] only in the case when n is an even natural number. For an odd natural number $n = 2k + 1 \geq 3$, choosing $\varphi \in \mathcal{B}_{n-1}$ we can construct the class f_φ as

$$f_\varphi(x_1, \ldots, x_n) = x_1 x_2 \cdots x_{n-1} \oplus \varphi(x_1, \ldots, x_{n-1}) \oplus x_n. \tag{11}$$

It should be pointed out that the use of these functions as filter functions in the filtering generators permit to guarantee non-trivial lower bounds of linear complexity of their output sequences (see, for example, [29]).

Proposition 1. *([6]) Let f_φ be the Boolean function defined by the equality (10). Then* $\mathsf{curv}(f_\varphi) = 2^{\frac{3n}{2}-1} + 2^n + (-1)^{f_\varphi(\mathbf{1})}2^{\frac{n}{2}+1}$, *where* $\mathbf{1}$=(1,1,...,1).

Next, we give the exact value of the curvature parameter for Boolean functions defined by (11) because it has never been published and some results contained in the proof of the following proposition will be used to determine other characteristic of this class.

Proposition 2. *Let f_φ be the Boolean function defined by the equality (11). Then* $\mathsf{curv}(f_\varphi) = 2^{\frac{3n-1}{2}} - 2^{\frac{n+3}{2}}$.

Proof. Walsh coefficients of f_φ are equals to

$$\mathcal{W}_{f_\varphi}(a) = \sum_{x_1,x_2,\ldots,x_n\in\mathbb{F}_2} (-1)^{x_1x_2\cdots x_{n-1}\oplus\varphi(x_1,\ldots,x_{n-1})\oplus x_n\oplus a_1x_1\oplus\cdots\oplus a_nx_n}.$$

If $a_n = 0$, then obviously $\mathcal{W}_{f_\varphi}(a) = 0$. If $a_n = 1$, then

$$\mathcal{W}_{f_\varphi}(a_1,\ldots,a_{n-1},1) = 2\mathcal{W}_\varphi(a_1,\ldots,a_{n-1}) - 4(-1)^{\varphi(\mathbf{1})\oplus a_1\oplus\cdots\oplus a_{n-1}},$$

where $\mathbf{1} = (1,1,\ldots,1)$. Thus

$$\mathcal{W}_{f_\varphi}(a_1,\ldots,a_n) = \begin{cases} 0, & \text{if } a_n = 0; \\ 2\mathcal{W}_\varphi(a_1,\ldots,a_{n-1}) - 4(-1)^{\varphi(\mathbf{1})\oplus a_1\oplus\cdots\oplus a_{n-1}}, & \text{if } a_n = 1. \end{cases}$$

Let us denote by $\tilde{\varphi}$ the dual of the Bent function φ, which satisfies the next relation

$$\mathcal{W}_\varphi(a) = (-1)^{\tilde{\varphi}(a)}2^{\frac{n-1}{2}}, \quad a\in\mathbb{F}_2^{n-1}. \tag{12}$$

It is well known (see, for example, [18, p. 253]), that $\tilde{\varphi}$ is a Bent-function. Hence the following function is also Bent

$$\psi(x_1,\ldots,x_{n-1}) = \tilde{\varphi}(x_1,\ldots,x_{n-1})\oplus x_1\oplus\cdots\oplus x_{n-1}\oplus\varphi(\mathbf{1}). \tag{13}$$

Taking into account, that $\tilde{\tilde{\varphi}} = \varphi$ the following relations holds $\mathcal{W}_\psi(\mathbf{0}) = (-1)^{\varphi(\mathbf{1})}\mathcal{W}_{\tilde{\varphi}}(1) = 2^{\frac{n-1}{2}} > 0$. In this way,

$$w_H(\psi) = 2^{n-2} - \frac{1}{2}\mathcal{W}_\psi(\mathbf{0}) = 2^{n-2} - 2^{\frac{n-1}{2}-1} = 2^{n-2} - 2^{\frac{n-3}{2}}.$$

By relation

$$(-1)^{a_1\oplus\cdots\oplus a_{n-1}\oplus\varphi(\mathbf{1})}\mathcal{W}_\varphi(a_1,\ldots,a_{n-1}) = 2^{\frac{n-1}{2}}(-1)^{\psi(a_1,\ldots,a_{n-1})}$$

it holds, that the set of numbers $(-1)^{a_1\oplus\cdots\oplus a_{n-1}\oplus\varphi(\mathbf{1})}\mathcal{W}_\varphi(a_1,\ldots,a_{n-1})$ contains $w_H(\psi)$ numbers equal to $-2^{\frac{n-1}{2}}$ and $2^{n-1} - w_H(\psi)$ numbers equal to $2^{\frac{n-1}{2}}$. Then,

$$\mathsf{curv}(f_\varphi) = (2^{\frac{n-1}{2}+1}+4)(2^{n-2}-2^{\frac{n-3}{2}})+(2^{\frac{n-1}{2}+1}-4)(2^{n-2}+2^{\frac{n-3}{2}}) = 2^{\frac{3n-1}{2}}-2^{\frac{n+3}{2}}.$$

□

Proposition 3. *Let f_φ be the Boolean function defined by the equality (11). Then* $\mathsf{nd}(f_\varphi) = 2^{n-2} - 2$.

Proof. We shall use the expressions of $\mathcal{W}_{f_\varphi}$, obtained in the proof of Proposition 2 and relation written in (5) for calculating the autocorelation of f_φ, which in this case has the following form

$$\begin{aligned}\Delta_{f_\varphi}(u_1,\ldots,u_n) &= \frac{1}{2^n}\sum_{v\in\mathbb{F}_2^n}\mathcal{W}^2_{f_\varphi}(v_1,\ldots,v_n)(-1)^{\langle(u_1,\ldots,u_n),(v_1,\ldots,v_n)\rangle} = \\ &= \frac{(-1)^{u_n}}{2^n}\sum_{v_1,\ldots,v_{n-1}\in\mathbb{F}_2}\mathcal{W}^2_{f_\varphi}(v_1,\ldots,v_{n-1},1)(-1)^{u_1v_1\oplus\ldots\oplus u_{n-1}v_{n-1}}.\end{aligned}$$

Let us denote $\hat{u} = (u_1,\ldots,u_{n-1}), \hat{v} = (v_1,\ldots,v_{n-1}) \in \mathbb{F}_2^{n-1}$. Then

$$\begin{aligned}\Delta_{f_\varphi}(\hat{u},u_n) &= \frac{(-1)^{u_n}}{2^n}\sum_{\hat{v}\in v_{n-1}}\mathcal{W}^2_{f_\varphi}(\hat{v},1)(-1)^{\langle\hat{u},\hat{v}\rangle} = \\ &= \frac{4(-1)^{u_n}}{2^n}\sum_{\hat{v}\in\mathbb{F}_2^{n-1}}\Big(\mathcal{W}_\varphi(\hat{v}) - 2(-1)^{v_1\oplus\ldots\oplus v_{n-1}\oplus\varphi(\mathbf{1})}\Big)^2(-1)^{\langle\hat{u},\hat{v}\rangle}.\end{aligned}$$

Now using relations (12), (13) we have

$$\begin{aligned}\Delta_{f_\varphi}(\hat{u},u_n) &= \frac{4(-1)^{u_n}}{2^n}\Bigg[\sum_{\hat{v}\in\mathbb{F}_2^{n-1}}2^{n-1}(-1)^{\langle\hat{u},\hat{v}\rangle} - 4\cdot 2^{\frac{n-1}{2}}\sum_{\hat{v}\in\mathbb{F}_2^{n-1}}(-1)^{\psi(\hat{v})\oplus\langle\hat{u},\hat{v}\rangle} + \\ &\qquad +4\sum_{\hat{v}\in\mathbb{F}_2^{n-1}}(-1)^{\langle\hat{u},\hat{v}\rangle}\Bigg] = \\ &= \frac{4(-1)^{u_n}}{2^n}\Big[2^{2n-2}\cdot\delta_{\mathbf{0}}(\hat{u}) - 4\cdot 2^{\frac{n-1}{2}}\mathcal{W}_\psi(\hat{u}) + 4\cdot 2^{n-1}\cdot\delta_{\mathbf{0}}(\hat{u})\Big],\end{aligned}$$

where $\delta_{\mathbf{0}}(\hat{u}) = \delta_0(u_1)\delta_0(u_2)\cdots\delta_0(u_{n-2})$.

Thus, if $\hat{u} = \mathbf{0}$ and $u_n = 1$ then $\Delta_{f_\varphi}(\mathbf{0},1) = -2^n$. If $u_n \in \{0,1\}$ and there exist some $i \in \{1,\ldots,n-1\}$ for which $u_i \neq 0$ we obtain $\Delta_{f_\varphi}(\hat{u},u_n) \in \{-8,8\}$. In this way we conclude that $\mathsf{nd}(f_\varphi) = 2^{n-2} - \frac{1}{4}\max\limits_{\mathbf{0}\neq u\in\mathbb{F}_2^n}\Delta_f(u) = 2^{n-2} - 2$. □

Proposition 4. *Let f_φ be the Boolean function defined by the equality (10). Then* $\mathsf{nd}(f_\varphi) = 2$.

Proof. The proof is quite similar to the proof of Proposition 3, using the expressions of $\mathcal{W}_{f_\varphi}$ obtained in [6]. □

n	$\mathsf{curv}(f_\varphi)$	$nl(f_\varphi)$	$d_{alg}(f_\varphi)$
even	$2^{\frac{3n}{2}-1}+2^n$ $+(-1)^{f_\varphi(\mathbf{1})}\cdot 2^{\frac{n}{2}+1}$	$2^{n-1}-2^{\frac{n}{2}}-2$	$n-1$
odd	$2^{\frac{3n-1}{2}}-2^{\frac{n+3}{2}}$	$2^{n-1}-2^{\frac{n-1}{2}}-2$	$n-1$

n	$\mathsf{ls}(f_\varphi)$	$\mathsf{nd}(f_\varphi)$
even	0	2
odd	0	$2^{n-2}-2$

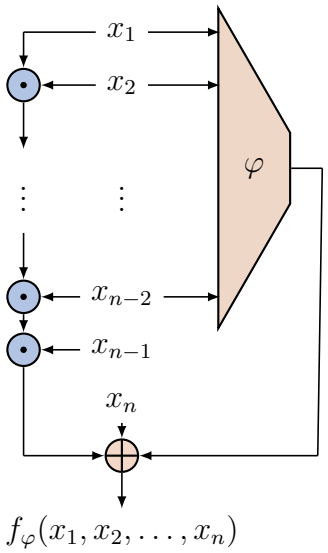

Figure 1: Cryptographic parameters of f_φ and its high level representation.

If denote $e_n = (0, \ldots, 0, 1)$, then from the proof of Proposition 3 we have $\Delta_{f_\varphi}(e_n) = \Delta_{f_\varphi}(\mathbf{0}, 1) = -2^n$ and for functions defined by (10) the following equality holds $f_\varphi(x) \oplus f_\varphi(x \oplus e_n) = 1$, which means that e_n is a linear structure for f_φ and in this case $\Delta_{f_\varphi}(e_n) = -2^n$. From relation $\max_{\mathbf{0}\neq u\in\mathbb{F}_2^n} |\Delta_{f_\varphi}(u)| \leq 2^n$, we conclude that both classes of functions defined by (10), (11) have $\mathsf{ls}(f_\varphi) = 2^{n-2} - \dfrac{1}{4}\max\limits_{\mathbf{0}\neq u\in\mathbb{F}_2^n} |\Delta_{f_\varphi}(u)| = 0$.

We compile in Figure 1 the high level representation when n is even (if n is odd, the scheme is almost identical) and some cryptographic characteristics of this class f_φ for any natural number. When $n \to \infty$, we have $\mathsf{curv}(f_\varphi) = O(2^{\frac{3n}{2}})$ and as can be observed from this figure, we can construct balanced Boolean function f_φ with maximal algebraic degree having high nonlinearity. However, if n is an even natural the nondegeneracy of f_φ is very low in contrast with the value $\mathsf{nd}(f_\varphi)$ when n is odd. In any case, the main weakness of these classes is the existence of nonzero linear structures.

2.2 Classes of functions obtained by using Maiorana–McFarland and Dobbertin constructions

Let $n = 2k$, where $k \geq 2$, $\Phi\colon \mathbb{F}_2^k \to \mathbb{F}_2^k$ - any mapping with coordinates functions $\Phi_i\colon \mathbb{F}_2^k \to \mathbb{F}_2, i = 1, 2, \ldots, k$, i.e., $\Phi(y) = (\Phi_1(y), \ldots, \Phi_k(y))$. For

all $x = (x_1, ..., x_k), y = (y_1, ..., y_k) \in \mathbb{F}_2^k$ we introduce the following notation.

$$\langle \Phi(y), x \rangle = \Phi_1(y)x_1 \oplus \Phi_2(y)x_2 \oplus \cdots \oplus \Phi_k(y)x_k. \tag{14}$$

Choosing an arbitrary Boolean function $h(y)$ in k-variables, we define the following n - variables Boolean function with the help of Maiorana–McFarland construction.

$$f_{\Phi,h}(x, y) = \langle x, \Phi(y) \rangle \oplus h(y), x, y \in \mathbb{F}_2^k. \tag{15}$$

Let $M = (m_{ij})_{k\times k}$ be a matrix over $\mathbb{F}_2$ with $\mathsf{rang}\, M = k - 1$. Consider the following map $\Phi\colon \mathbb{F}_2^k \to \mathbb{F}_2^k$ such that $\Phi(y) = y \cdot M$, for any $y \in \mathbb{F}_2^k$. By L we denote the linear subspace, generated by all rows of M. From condition $\mathsf{rang}\, M = k - 1$, we have $\#L = 2^{k-1}$ and in this context it is not difficult to see, that the linear transformation Φ satisfies the following relations.

$$\Phi^{-1}(\mathbf{0}) = \{\mathbf{0}, c\}, \quad \Phi^{-1}(v) = \begin{cases} \emptyset, & \text{if } v \notin L; \\ \{d, d \oplus c\}, & \text{if } v \in L \text{ and } dM = v. \end{cases} \tag{16}$$

Proposition 5. *([6]) Let $f_{\Phi,h}$ be the boolean function defined by relation (15), where $\Phi(y) = y \cdot M$. Then $\mathsf{curv}(f_{\Phi,h}) = 2^{\frac{3n}{2}-1}$.*

It should be pointed out that Boolean functions defined by (15) will be balanced if and only if $\mathcal{W}_{f_{\Phi,h}}(\mathbf{0}, \mathbf{0}) = 0$, which according to [6, p. 64] it happens when $h(\mathbf{0}) \neq h(c)$.

Proposition 6. *Let $f_{\Phi,h}$ be the Boolean function defined by the equality (15). Then $\mathsf{nd}(f_{\Phi,h}) = 0$.*

Proof. For any $u, \hat{u} \in \mathbb{F}_2^k$ the autocorrelation function of $f_{\Phi,h}$ is defined as $\Delta_{f_{\Phi,h}}(u, \hat{u}) = \sum\limits_{x,y\in\mathbb{F}_2^k} (-1)^{f_{\Phi,h}(x,y)\oplus f_{\Phi,h}(x\oplus u, y\oplus \hat{u})}$.

Let us consider the function $f_{\Phi,h}(x, y) \oplus f_{\Phi,h}(x \oplus u, y \oplus \hat{u})$. Using the properties of the scalar product and the fact that Φ is a linear transformation we obtain $f_{\Phi,h}(x, y) \oplus f_{\Phi,h}(x \oplus u, y \oplus \hat{u}) = \langle x, \Phi(\hat{u}) \rangle \oplus h(y) \oplus h(y \oplus \hat{u}) \oplus \langle u, \Phi(y \oplus \hat{u}) \rangle$. So $\Delta_{f_{\Phi,h}}(u, \hat{u})$ has the following form

$$\Delta_{f_{\Phi,h}}(u, \hat{u}) \;=\; \sum_{x\in\mathbb{F}_2^k} (-1)^{\langle x,\Phi(\hat{u})\rangle} \sum_{y\in\mathbb{F}_2^k} (-1)^{h(y)\oplus h(y\oplus\hat{u})\oplus\langle u,\Phi(y\oplus\hat{u})\rangle}.$$

If $\hat{u} = 0$ and $u \neq \mathbf{0}$ we have

$$\Delta_{f_{\Phi,h}}(u, \mathbf{0}) \;=\; 2^k \sum_{y\in\mathbb{F}_2^k} (-1)^{\langle u,\Phi(y)\rangle} = 2^{k+1} \sum_{v\in L} (-1)^{\langle u,v\rangle}.$$

Because $\mathsf{rang}\, M = k-1 \Rightarrow \exists\, \mathbf{0} \neq u \in \mathbb{F}_2^k \colon M \cdot u^{\mathsf{T}} = \mathbf{0}^{\mathsf{T}} \Rightarrow \forall\, d \in \Phi^{-1}(v) \neq \emptyset, (d \cdot M) \cdot u^{\mathsf{T}} = 0 \Rightarrow \forall\, v \in L, \langle u, v\rangle = 0 \Rightarrow \Delta_{f_{\Phi,h}}(u, \mathbf{0}) = 2^{k+1} \sum_{v \in L} (-1)^0 = 2^{k+1} \cdot \#L = 2^{2k}$.

Now, from relation $\max\limits_{(\mathbf{0},\mathbf{0}) \neq (u,\hat{u}) \in \mathbb{F}_2^k \times \mathbb{F}_2^k} \Delta_{f_{\Phi,h}}(u, \hat{u}) \leq 2^{2k}$ and equality $\Delta_{f_{\Phi,h}}(u, \mathbf{0}) = 2^{2k}$, which holds for some nonzero $u \in \mathbb{F}_2^k$, we conclude that $\mathsf{nd}(f_{\Phi,h}) = 2^{2k-2} - \dfrac{1}{4} \max\limits_{(\mathbf{0},\mathbf{0}) \neq (u,\hat{u}) \in \mathbb{F}_2^k \times \mathbb{F}_2^k} \Delta_{f_{\Phi,h}}(u, \hat{u}) = 2^{2k-2} - \dfrac{1}{4} \cdot 2^{2k} = 0$. □

From the proof of Proposition 6, we derive $\max\limits_{(\mathbf{0},\mathbf{0}) \neq (u,\hat{u}) \in \mathbb{F}_2^k \times \mathbb{F}_2^k} |\Delta_{f_{\Phi,h}}(u, \hat{u})| = 2^{2k}$, which means that $\mathsf{ls}(f_{\Phi,h}) = 0$. Also, from (15) we obtain that $d_{alg}(f_{\Phi,h}) = d_{alg}(h)$, when $d_{alg}(h) \geq 3$ and $d_{alg}(f_{\Phi,h}) = 2$, if $d_{alg}(h) \leq 2$.

n	$\mathsf{curv}(f_{\Phi,h})$	$nl(f_{\Phi,h})$	$d_{alg}(f_{\Phi,h})$
even	$2^{\frac{3n}{2}-1}$	$2^{n-1} - 2^{\frac{n}{2}}$	$d_{alg}(h)$, if $d_{alg}(h) \geq 3$; 2, if $d_{alg}(h) \leq 2$.

n	$\mathsf{ls}(f_{\Phi,h})$	$\mathsf{nd}(f_{\Phi,h})$
even	0	0

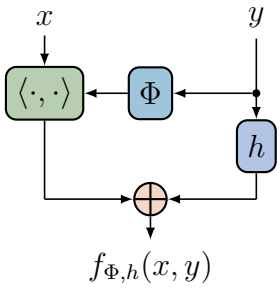

Figure 2: Cryptographic parameters of $f_{\Phi,h}$ and its high level representation.

We present in Figure 2 some cryptographic characteristics of the class $f_{\Phi,h}$. It can be seen that $\mathsf{curv}(f_{\Phi,h}) = O(2^{\frac{3n}{2}})$ when $n \to \infty$ and thus we can expect a high value of nonlinearity parameter as indeed was obtained in [6]. However, the algebraic degree of $f_{\Phi,h}$ can not exceed the degree of the function h in k variables. Moreover, this class has nonzero linear structures and the nondegeneracy of these functions is zero which mean that Boolean function $f_{\Phi,h}$ are susceptible to a method of analysis based on algebraically degenerate approximations [1].

Now we shall choose an arbitrary mapping $\Phi \colon \mathbb{F}_2^k \to \mathbb{F}_2^k$ which satisfies the following conditions

1. $\Phi(\beta) = \Phi(\hat{\beta}) = \gamma \neq \mathbf{0}$ for some $\beta, \hat{\beta}, \gamma \in \mathbb{F}_2^k$;
2. The mapping $\dot{\Phi} \colon \mathbb{F}_2^k \setminus \{\beta, \hat{\beta}\} \to \mathbb{F}_2^k \setminus \{\mathbf{0}, \gamma\}$ such that $\dot{\Phi}(x) = \Phi(x)$ for any $x \in \mathbb{F}_2^k \setminus \{\beta, \hat{\beta}\}$ is injective.

Mappings Φ which satisfies conditions $1, 2$ are particular cases of those non-bijective k-bit functions without preimage for $\mathbf{0}$ used in [7] for generating

highly nonlinear S-Boxes. Using these mappings we can construct a class of Boolean function in $n = 2k$ variables as follows

$$f_\Phi(x, y) = \langle \Phi(x), y \rangle, x, y \in \mathbb{F}_2^k. \tag{17}$$

Proposition 7. *Let f_Φ be the Boolean function defined by relation (17) where the mappings Φ satisfies conditions $1, 2$ listed above. Then $\mathsf{curv}(f_\Phi) = 2^{\frac{3n}{2}} - 2^n$.*

Proof. Walsh coefficients $\mathcal{W}_{f_\Phi}(v, w)$ for any $v, w \in \mathbb{F}_2^k$ are equals to

$$\begin{aligned}\mathcal{W}_{f_\Phi}(v, w) &= \sum_{(x,y)\in\mathbb{F}_2^k\times\mathbb{F}_2^k} (-1)^{f_\Phi(x,y)\oplus\langle(x,y),(v,w)\rangle} = \\ &= \sum_{x\in\mathbb{F}_2^k} (-1)^{\langle v,x\rangle} \sum_{y\in\mathbb{F}_2^k} (-1)^{\langle y,\Phi(x)\oplus w\rangle} = 2^k \cdot \sum_{x\in\Phi^{-1}(w)} (-1)^{\langle v,x\rangle}.\end{aligned}$$

Thus,

$$\mathcal{W}_{f_\Phi}(v, w) = \begin{cases} 2^k \cdot (-1)^{\left\langle v,\Phi^{-1}(w)\right\rangle}, & \text{if } w \notin \{\mathbf{0}, \gamma\}; \\ 0, & \text{if } w = \mathbf{0}; \\ 2^k \cdot \left((-1)^{\langle v,\beta\rangle} + (-1)^{\left\langle v,\hat{\beta}\right\rangle}\right), & \text{if } w = \gamma. \end{cases} \tag{18}$$

For the sake of clarity It should be pointed out that when $w = \mathbf{0}$ we have $\Phi^{-1}(\mathbf{0}) = \emptyset$ and $\mathcal{W}_{f_\Phi}(v, \mathbf{0}) = 0$ for any $v \in \mathbb{F}_2^k$. So, we conclude that $\mathsf{curv}(f_\Phi) = \sum_{(v,w)\in\mathbb{F}_2^k\times\mathbb{F}_2^k} |\mathcal{W}_{f_\Phi}(v, w)| = 2^k \cdot 2^k \cdot (2^k - 2) + 2^{k+1} \cdot 2^{k-1} = 2^{3k} - 2^{2k} = 2^{\frac{3n}{2}} - 2^n$. □

It follows, from the proof of Proposition 7, that $\mathcal{W}_{f_\Phi}(\mathbf{0}, \mathbf{0}) = 0$, i.e., all Boolean function of the form (17) are balanced. The next result shows that these functions are not susceptible to a method of analysis based on algebraically degenerate approximations presented in [1].

Proposition 8. *Let f_Φ be the Boolean function defined by the equality (17). Then $\mathsf{nd}(f_\Phi) = 2^{n-2} - 2^{\frac{n}{2}-1}$.*

Proof. For any $u, \hat{u} \in \mathbb{F}_2^k$ the autocorrelation function of f_Φ is defined as $\Delta_{f_\Phi}(u, \hat{u}) = \sum_{x,y\in\mathbb{F}_2^k} (-1)^{f_\Phi(x,y)\oplus f_\Phi(x\oplus u, y\oplus\hat{u})}$. Using the properties of the scalar product we obtain that $f_\Phi(x, y) \oplus f_\Phi(x \oplus u, y \oplus \hat{u}) = \langle \Phi(x) \oplus \Phi(x \oplus u), y\rangle \oplus \langle \Phi(x \oplus u), \hat{u}\rangle$. Then

$$\Delta_{f_\Phi}(u,\hat{u}) \;=\; \sum_{x\in\mathbb{F}_2^k}(-1)^{\langle\Phi(x\oplus u),\hat{u}\rangle}\sum_{y\in\mathbb{F}_2^k}(-1)^{\langle\Phi(x)\oplus\Phi(x\oplus u),y\rangle}.$$

If $u=\mathbf{0}$ and $\hat{u}\neq\mathbf{0}$ we have

$$\begin{aligned}\Delta_{f_\Phi}(\mathbf{0},\hat{u}) \;&=\; 2^k\cdot\sum_{x\in\mathbb{F}_2^k}(-1)^{\langle\Phi(x),\hat{u}\rangle}=2^k\cdot\Big((-1)^{\langle\Phi(\beta),\hat{u}\rangle}+(-1)^{\langle\Phi(\hat{\beta}),\hat{u}\rangle}\Big)+\\ &\quad+2^k\cdot\sum_{\beta,\hat{\beta}\neq x\in\mathbb{F}_2^k}(-1)^{\langle\Phi(x),\hat{u}\rangle}=\\ &=\; 2^k\cdot\Big(2\cdot(-1)^{\langle\gamma,\hat{u}\rangle}\Big)+2^k\cdot\sum_{\beta,\hat{\beta}\neq x\in\mathbb{F}_2^k}(-1)^{\langle\Phi(x),\hat{u}\rangle}.\end{aligned}$$

Let us find the value of $\sum_{\beta,\hat{\beta}\neq x\in\mathbb{F}_2^k}(-1)^{\langle\Phi(x),\hat{u}\rangle}$. By condition 2 (listed above), the mapping $\dot{\Phi}\colon\mathbb{F}_2^k\setminus\{\beta,\hat{\beta}\}\to\mathbb{F}_2^k\setminus\{\mathbf{0},\gamma\}$ such that $\dot{\Phi}(x)=\Phi(x)$ for any $x\in\mathbb{F}_2^k\setminus\{\beta,\hat{\beta}\}$ is injective. Let now $\Phi'\colon\mathbb{F}_2^k\to\mathbb{F}_2^k$ be a mapping defined as follows

$$\Phi'(x)=\begin{cases}\mathbf{0}, & \text{if } x=\beta;\\ \gamma, & \text{if } x=\hat{\beta};\\ \dot{\Phi}(x), & \text{if } x\notin\{\beta,\hat{\beta}\}.\end{cases}\tag{19}$$

Obviously, the mapping defined by (19) is a permutation on $\mathbb{F}_2^k\;\Rightarrow\;\sum_{x\in\mathbb{F}_2^k}(-1)^{\langle\Phi'(x),\hat{u}\rangle}=(-1)^{\langle\Phi'(\beta),\hat{u}\rangle}+(-1)^{\langle\Phi'(\hat{\beta}),\hat{u}\rangle}+\sum_{\beta,\hat{\beta}\neq x\in\mathbb{F}_2^k}(-1)^{\langle\Phi(x),\hat{u}\rangle}=0\;\Rightarrow\;\sum_{\beta,\hat{\beta}\neq x\in\mathbb{F}_2^k}(-1)^{\langle\Phi(x),\hat{u}\rangle}=-\Big(1+(-1)^{\langle\gamma,\hat{u}\rangle}\Big)\;\Rightarrow\;\Delta_{f_\Phi}(\mathbf{0},\hat{u})=2^k\big((-1)^{\langle\gamma,\hat{u}\rangle}-1\big)$ and in this way, we obtain $\Delta_{f_\Phi}(\mathbf{0},\hat{u})\in\{0,-2^{k+1}\}$.

When $u\neq\mathbf{0}$ and $\hat{u}\in\mathbb{F}_2^k$ we have

$$\begin{aligned}\Delta_{f_\Phi}(u,\hat{u}) \;&=\; (-1)^{\langle\Phi(\beta\oplus u),\hat{u}\rangle}\sum_{y\in\mathbb{F}_2^k}(-1)^{\langle\Phi(\beta)\oplus\Phi(\beta\oplus u),y\rangle}+\\ &\quad+(-1)^{\langle\Phi(\hat{\beta}\oplus u),\hat{u}\rangle}\sum_{y\in\mathbb{F}_2^k}(-1)^{\langle\Phi(\hat{\beta})\oplus\Phi(\hat{\beta}\oplus u),y\rangle}+\\ &\quad+\sum_{\beta,\hat{\beta}\neq x\in\mathbb{F}_2^k}(-1)^{\langle\Phi(x\oplus u),\hat{u}\rangle}\sum_{y\in\mathbb{F}_2^k}(-1)^{\langle\Phi(x)\oplus\Phi(x\oplus u),y\rangle}.\end{aligned}$$

Because Φ is an injective function on $\mathbb{F}_2^k\setminus\{\beta,\hat{\beta}\}\;\Rightarrow\;\forall\beta,\hat{\beta}\neq x\in\mathbb{F}_2^k,\Phi(x)\oplus\Phi(x\oplus u)\neq 0\Rightarrow\sum_{y\in\mathbb{F}_2^k}(-1)^{\langle\Phi(x)\oplus\Phi(x\oplus u),y\rangle}=0\Rightarrow\Delta_{f_\Phi}(u,\hat{u})=$

$$2^k \cdot \Big((-1)^{\langle \Phi(\beta)\oplus\Phi(\beta\oplus u),\hat{u}\rangle}\delta_{\Phi(\beta\oplus u)}(\Phi(\beta)) + (-1)^{\langle \Phi(\hat{\beta})\oplus\Phi(\hat{\beta}\oplus u),\hat{u}\rangle}\delta_{\Phi(\hat{\beta}\oplus u)}(\Phi(\hat{\beta}))\Big).$$

Thus, for a nonzero vector $u \in \mathbb{F}_2^k$ such that $\hat{\beta} = \beta \oplus u$, $\Delta_{f_\Phi}(u,\hat{u}) = 2^{k+1}$, otherwise $\Delta_{f_\Phi}(u,\hat{u}) = 0$. In this way, we have $\Delta_{f_\Phi}(u,\hat{u}) \in \{0, 2^{k+1}\}$.

From previous cases, we conclude that $\mathsf{nd}(f_\Phi) = 2^{2k-2} - \frac{1}{4}\max_{(\mathbf{0},\mathbf{0})\neq(u,\hat{u})\in\mathbb{F}_2^k\times\mathbb{F}_2^k}\Delta_{f_\Phi}(u,\hat{u}) = 2^{2k-2} - \frac{1}{4}\cdot 2^{k+1} = 2^{n-2} - 2^{\frac{n}{2}-1}$. □

From relation (14) we derive $d_{alg}(f_\Phi) = d_{alg}(\Phi) + 1$ and from the proof of Proposition 7, 8 we deduce $nl(f_\Phi) = 2^{n-1} - 2^{\frac{n}{2}}$ and $\max_{(\mathbf{0},\mathbf{0})\neq(u,\hat{u})\in\mathbb{F}_2^k\times\mathbb{F}_2^k}|\Delta_{f_\Phi}(u,\hat{u})| = 2^{k+1}$ respectively, the latter implies that $\mathsf{ls}(f_\Phi) = 2^{n-2} - 2^{\frac{n}{2}-1}$.

n	$\mathsf{curv}(f_\Phi)$	$nl(f_\Phi)$	$d_{alg}(f_\Phi)$	$\mathsf{ls}(f_\Phi)$
even	$2^{\frac{3n}{2}} - 2^n$	$2^{n-1} - 2^{\frac{n}{2}}$	$d_{alg}(\Phi)+1$	$2^{n-2} - 2^{\frac{n}{2}-1}$

n	$\mathsf{nd}(f_\Phi)$
even	$2^{n-2} - 2^{\frac{n}{2}-1}$

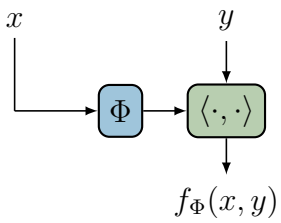

Figure 3: Cryptographic parameters of f_Φ and its high level representation.

When the simplicity of this class of functions it is more than clear, which can be observed in Figure 3, we can generate Boolean function with high nonlinearity having $\mathsf{curv}(f_\Phi) \sim 2^{\frac{3n}{2}}$ when $n \to \infty$, but the algebraic degree of this function is far from optimal. However, parameters $\mathsf{ls}(f_\Phi), \mathsf{nd}(f_\Phi)$ are not equal to zero and, thus these functions can resist some methods of analysis which exploit the low values of these parameters.

Let us consider the following construction studied in [16] which generalize the particular case investigated in [6]. Choosing an arbitrary Boolean function $g \in \mathcal{F}_k$ and a normal function[b] $\varphi \in \mathcal{B}_{2k}$, we construct $f_{\varphi,g}$ as follows

$$f_{\varphi,g}(x,y) = \begin{cases} g(y), & \text{if } x = \alpha; \\ \varphi(x,y), & \text{if } x \neq \alpha. \end{cases} \tag{20}$$

In the next proposition the exact value of $\mathsf{curv}(f_{\varphi,g})$ is given.

Proposition 9. *([16]) Let $f_{\varphi,g}$ be the Boolean function defined by relation (20). Then $\mathsf{curv}(f_{\psi,g}) = 2^{\frac{3n}{2}} - 2^n + 2^{\frac{n}{2}}|\mathcal{W}_g(\mathbf{0})|$.*

Let us point out, that all function of the form (20) will be balanced if and only if when $w_H(g) = 2^{k-1}$.

[b] An n-variable Boolean function is a normal function if it is constant (resp. affine) on at least one $\frac{n}{2}$ - dimensional subspace.

Proposition 10. *Let $f_{\varphi,g}$ be the Boolean function defined by relation (20) with $\alpha = \mathbf{0}$, any balanced Boolean function in k-variables g and $\varphi(x,y) = \langle \pi(x), y\rangle$, where π is any k-bit permutation such that $\pi(\mathbf{0}) = \mathbf{0}$. Then $\mathsf{nd}(f_{\varphi,g}) \geq 2^{n-2} - 2^{\frac{n}{2}-1}$.*

Proof. For any $u, \hat{u} \in \mathbb{F}_2^k$ the autocorrelation function of $f_{\varphi,g}$ is defined as $\Delta_{f_{\varphi,g}}(u,\hat{u}) = \sum_{x,y\in\mathbb{F}_2^k} (-1)^{f_{\varphi,g}(x,y)\oplus f_{\varphi,g}(x\oplus u, y\oplus\hat{u})}$.

If $u = \mathbf{0}$ and $\hat{u} \neq \mathbf{0}$. Then

$$\begin{aligned}
\Delta_{f_{\varphi,g}}(\mathbf{0},\hat{u}) &= \sum_{y\in\mathbb{F}_2^k} (-1)^{g(y)\oplus g(y\oplus\hat{u})} + \sum_{\substack{\mathbf{0}\neq x\in\mathbb{F}_2^k \\ y\in\mathbb{F}_2^k}} (-1)^{\langle\pi(x),y\rangle\oplus\langle\pi(x),y\oplus\hat{u}\rangle} = \\
&= \Delta_g(\hat{u}) + 2^k\cdot\left(\sum_{x\in\mathbb{F}_2^k} (-1)^{\langle\pi(x),\hat{u}\rangle} - (-1)^{\langle\pi(0),\hat{u}\rangle}\right) = \\
&= \Delta_g(\hat{u}) - 2^k,
\end{aligned}$$

where $\sum_{x\in\mathbb{F}_2^k} (-1)^{\langle\pi(x),\hat{u}\rangle} = 0$ because π is a permutation and $\hat{u} \neq \mathbf{0}$.

If $u \neq \mathbf{0}$ and $\hat{u} \in \mathbb{F}_2^k$ we can use the following partition of $\mathbb{F}_2^k \times \mathbb{F}_2^k$ for calculating the autocorrelation function, $\mathbb{F}_2^k \times \mathbb{F}_2^k = \{(\mathbf{0},y) \mid y \in \mathbb{F}_2^k\} \bigsqcup \{(u,y) \mid y \in \mathbb{F}_2^k\} \bigsqcup \{(x,y) \mid \mathbf{0}, u \neq x \in \mathbb{F}_2^k \; y \in \mathbb{F}_2^k\}$. Then we have

$$\begin{aligned}
\Delta_{f_{\varphi,g}}(u,\hat{u}) &= \sum_{\substack{x=\mathbf{0} \\ y\in\mathbb{F}_2^k}} (-1)^{f_{\varphi,g}(\mathbf{0},y)\oplus f_{\varphi,g}(u,y\oplus\hat{u})} + \sum_{\substack{x=u \\ y\in\mathbb{F}_2^k}} (-1)^{f_{\varphi,g}(u,y)\oplus f_{\varphi,g}(\mathbf{0},y\oplus\hat{u})} + \\
&\quad + \sum_{\substack{\mathbf{0},u\neq x\in\mathbb{F}_2^k \\ y\in\mathbb{F}_2^k}} (-1)^{f_{\varphi,g}(x,y)\oplus f_{\varphi,g}(x\oplus u,y\oplus\hat{u})} = \\
&= 2\cdot\sum_{y\in\mathbb{F}_2^k} (-1)^{g(y)\oplus\langle\pi(u),y\rangle\oplus\langle\pi(u),\hat{u}\rangle} + \sum_{\substack{\mathbf{0},u\neq x\in\mathbb{F}_2^k \\ y\in\mathbb{F}_2^k}} (-1)^{\langle\pi(x),y\rangle\oplus\langle\pi(x\oplus u),y\oplus\hat{u}\rangle} = \\
&= 2\cdot(-1)^{\langle\pi(u),\hat{u}\rangle}\mathcal{W}_g(\pi(u)) + \mathcal{T}(u,\hat{u}),
\end{aligned}$$

where $\mathcal{T}(u,\hat{u}) = \sum_{\substack{\mathbf{0},u\neq x\in\mathbb{F}_2^k \\ y\in\mathbb{F}_2^k}} (-1)^{\langle\pi(x),y\rangle\oplus\langle\pi(x\oplus u),y\oplus\hat{u}\rangle}$. Let us find the value of $\mathcal{T}(u,\hat{u})$. Using the properties of the scalar product we have

$$\mathcal{T}(u,\hat{u}) \;=\; \sum_{\mathbf{0},u\neq x\in\mathbb{F}_2^k} (-1)^{\langle \pi(x\oplus u),\hat{u}\rangle} \sum_{y\in\mathbb{F}_2^k} (-1)^{\langle \pi(x)\oplus\pi(x\oplus u),y\rangle}.$$

The function π is a permutation on $\mathbb{F}_2^k \Rightarrow \forall\, \mathbf{0}, u\neq x\in\mathbb{F}_2^k, \pi(x)\oplus\pi(x\oplus u)\neq 0 \Rightarrow \sum_{y\in\mathbb{F}_2^k}(-1)^{\langle \pi(x)\oplus\pi(x\oplus u),y\rangle} = 0 \Rightarrow \mathcal{T}(u,\hat{u}) = 0 \Rightarrow \Delta_{f_{\varphi,g}}(u,\hat{u}) = 2\cdot(-1)^{\langle \pi(u),\hat{u}\rangle}\mathcal{W}_g(\pi(u))$.

Now, taking into account that $\max\limits_{\hat{u}\in\mathbb{F}_2^k}\Delta_g(\hat{u}) \leq 2^k$ and $\max\limits_{u\in\mathbb{F}_2^k}\mathcal{W}_g(u)\leq 2^k$ we conclude that $\max\limits_{(\mathbf{0},\mathbf{0})\neq(u,\hat{u})\in\mathbb{F}_2^k\times\mathbb{F}_2^k}\Delta_{f_{\varphi,g}}(u,\hat{u}) \leq 2^{k+1}$, hence the nondegeneracy parameter of the class $f_{\varphi,g}$ can be lower bounded as follows $\mathsf{nd}(f_{\varphi,g}) = 2^{2k-2} - \frac{1}{4}\max\limits_{(\mathbf{0},\mathbf{0})\neq(u,\hat{u})\in\mathbb{F}_2^k\times\mathbb{F}_2^k}\Delta_{f_{\varphi,g}}(u,\hat{u}) \geq 2^{n-2} - 2^{\frac{n}{2}-1}$.

□

From the proof of Proposition 10 we can easy obtain that $\max\limits_{(\mathbf{0},\mathbf{0})\neq(u,\hat{u})\in\mathbb{F}_2^k\times\mathbb{F}_2^k}|\Delta_{f_{\varphi,g}}(u,\hat{u})| \leq 2^{k+1}$, hence $\mathsf{ls}(f_{\varphi,g}) \geq 2^{n-2} - 2^{\frac{n}{2}-1}$.

When $\varphi(x,y) = \langle \pi(x), y\rangle$ we present in Figure 4 some cryptographic characteristics of the class $f_{\varphi,g}$. As can be observed from this figure, $\mathsf{curv}(f_{\varphi,g}) \sim 2^{\frac{3n}{2}}$ when $n\to\infty$, which means that functions $f_{\varphi,g}$ live near the set of maximally nonlinear functions (as the non-linearity parameter of $f_{\varphi,g}$ confirms it) offering at the same time the possibility to construct some candidate with maximal algebraic degree.

n	$\mathsf{curv}(f_{\varphi,g})$	$nl(f_{\varphi,g})$	$d_{alg}(f_{\varphi,g})$
even	$2^{\frac{3n}{2}} - 2^n$	$2^{n-1} - 2^{\frac{n}{2}} + nl(g)$	$\frac{n}{2} + d_{alg}(g)$

n	$\mathsf{ls}(f_{\varphi,g})$	$\mathsf{nd}(f_{\varphi,g})$
even	$\geq (2^{n-2} - 2^{\frac{n}{2}-1})$	$\geq (2^{n-2} - 2^{\frac{n}{2}-1})$

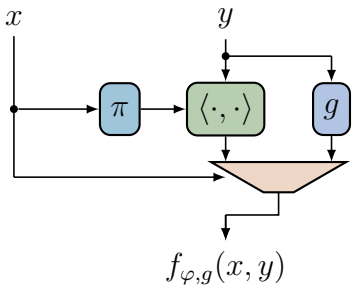

Figure 4: Cryptographic parameters of $f_{\varphi,g}$ and its high level representation.

2.3 A subclass of perfectly balanced Boolean functions without right barrier

In this section we are interesting in the analysis of the curvature parameter for one simple subclass of the so-called *perfectly balanced* Boolean

functions without right barrier. The property of being perfectly balanced was formalized by Sumarokov S.N. (see, for example, [24]) and this kind of functions were deeply studied in the following articles ([17, 24, 25, 26, 27, 28]). It is worth noticing that, if the function $f \in \mathcal{F}_n$ is a perfectly balanced Boolean function without left and right barriers, then it can be more resistant to the so-called inversion attack (see, [13]) and when using it as filter function in filtering generators, the statistical properties of sequences generated by them can be improved.

For a natural number n consider the following construction

$$f_g(x_1, \ldots, x_n) = x_1 \oplus x_2 x_3 g(x_4, \ldots, x_n) \oplus x_{n-1} x_n, \tag{21}$$

where the function g is a Boolean function in $n-3$ variables such that $g(1, \ldots, 1) = 1$. All functions of the form (21) conforms a particular subset of the class of perfectly balanced Boolean functions without right barrier considered in [24] and it is not difficult to see that $d_{alg}(f_g) = d_{alg}(g) + 2$. In the following proposition we determine the general expression of its Walsh coefficients when x_{n-1}, x_n are not essential arguments.

Proposition 11. *For any Boolean function $g(x_4, \ldots, x_{n-2}, x_{n-1}, x_n) \in \mathcal{F}_{n-3}$ which arguments x_{n-1}, x_n are not essential, the Walsh coefficients $\mathcal{W}_{f_g}$ of the function constructed by (21) are determined by the following relations*

$$\mathcal{W}_{f_g}(a_1, a_2, a_3, \hat{a}, a_{n-1}, a_n) = \begin{cases} 0, & \text{if } a_1 = 0; \\ 6 \cdot 2^{n-4} \delta_{\mathbf{0}}(\hat{a}) + 4\mathcal{W}_g(\hat{a}), & \text{if } a_1 = 1, (a_2, a_3, a_{n-1}, a_n) \in \mathcal{A}_1; \\ -6 \cdot 2^{n-4} \delta_{\mathbf{0}}(\hat{a}) - 4\mathcal{W}_g(\hat{a}), & \text{if } a_1 = 1, (a_2, a_3, a_{n-1}, a_n) \in \mathcal{A}_2; \\ 2 \cdot 2^{n-4} \delta_{\mathbf{0}}(\hat{a}) - 4\mathcal{W}_g(\hat{a}), & \text{if } a_1 = 1, (a_2, a_3, a_{n-1}, a_n) \in \mathcal{A}_3; \\ -2 \cdot 2^{n-4} \delta_{\mathbf{0}}(\hat{a}) + 4\mathcal{W}_g(\hat{a}), & \text{if } a_1 = 1, (a_2, a_3, a_{n-1}, a_n) \in \mathcal{A}_4. \end{cases},$$

where $\delta_{\mathbf{0}}(\hat{a}) = \delta_0(a_4) \cdots \delta_0(a_{n-2})$ for any $\hat{a} = (a_4, \ldots, a_{n-2}) \in \mathbb{F}_2^{n-5}$, $(a_2, a_3, a_{n-1}, a_n) \in \mathbb{F}_2^4$, and sets $\mathcal{A}_i, i = 1, 2, 3, 4$, are defined in the following way $\mathcal{A}_1 = \{(0,0,0,0), (0,0,0,1), (0,0,1,0)\}, \mathcal{A}_2 = \{(0,0,1,1)\}, \mathcal{A}_3 = \{(0,1,0,0), (0,1,0,1), (0,1,1,0), (1,0,0,0), (1,0,0,1), (1,0,1,0), (1,1,1,1)\}$ and $\mathcal{A}_4 = \{(0,1,1,1), (1,0,1,1), (1,1,0,0), (1,1,0,1), (1,1,1,0)\}$.

Proof. Walsh coefficients $\mathcal{W}_{f_g}$ are equals to

$$\mathcal{W}_{f_g}(a_1, \ldots, a_n) = \sum_{x_1, x_2, \ldots, x_n \in \mathbb{F}_2} (-1)^{x_1 \oplus x_2 x_3 g(x_4, \ldots, x_n) \oplus x_{n-1} x_n \oplus a_1 x_1 \oplus \cdots \oplus a_n x_n}.$$

If $a_1 = 0$, then obviously $\mathcal{W}_{f_g}(a_1, \ldots, a_n) = 0$. If $a_1 = 1$, then we have

$$\mathcal{W}_{f_g}(a_1, \ldots, a_n) = 2\mathcal{W}_{g'}(a_2, \ldots, a_n),$$

where $g' = x_2x_3g(x_4,\ldots,x_n)\oplus x_{n-1}x_n$. Let us find the value $\mathcal{W}_{g'}(a_2,\ldots,a_n)$.

$$\begin{aligned}
\mathcal{W}_{g'}(a_2,\ldots,a_n) &= \sum_{x_2,x_3,\ldots,x_n\in\mathbb{F}_2} (-1)^{x_2x_3g(x_4,\ldots,x_n)\oplus x_{n-1}x_n\oplus a_2x_2\oplus\cdots\oplus a_nx_n} = \\
&= \underbrace{\sum_{x_4,\ldots,x_n\in\mathbb{F}_2} (-1)^{x_{n-1}x_n\oplus a_4x_4\oplus\cdots\oplus a_nx_n}}_{x_2=0,x_3=0.} + \\
&\quad + \underbrace{\sum_{x_4,\ldots,x_n\in\mathbb{F}_2} (-1)^{x_{n-1}x_n\oplus a_3\oplus a_4x_4\oplus\cdots\oplus a_nx_n}}_{x_2=0,x_3=1.} + \\
&\quad + \underbrace{\sum_{x_4,\ldots,x_n\in\mathbb{F}_2} (-1)^{x_{n-1}x_n\oplus a_2\oplus a_4x_4\oplus\cdots\oplus a_nx_n}}_{x_2=1,x_3=0.} + \\
&\quad + \underbrace{\sum_{x_4,\ldots,x_n\in\mathbb{F}_2} (-1)^{g(x_4,\ldots,x_n)\oplus x_{n-1}x_n\oplus a_2\oplus a_3\oplus a_4x_4\oplus\cdots\oplus a_nx_n}}_{x_2=1,x_3=1.} = \\
&= \left(1+(-1)^{a_2}+(-1)^{a_3}\right)\cdot\mathcal{T}(a_4,\ldots,a_n) + \\
&\quad + (-1)^{a_2\oplus a_3}\mathcal{W}_{\dot{g}}(a_4,\ldots,a_n),
\end{aligned}$$

where $\mathcal{T}(a_4,\ldots,a_n) = \sum_{x_4,\ldots,x_n\in\mathbb{F}_2} (-1)^{x_{n-1}x_n\oplus a_4x_4\oplus\cdots\oplus a_nx_n}$ and $\dot{g}$ is a Boolean function in $n-3$ variables such that $\dot{g}(x_4,\ldots,x_n) = g(x_4,\ldots,x_n)\oplus x_{n-1}x_n$. By direct computations it is not hard to obtain the following relations for $\mathcal{T}(a_4,\ldots,a_n)$ and $\mathcal{W}_{\dot{g}}(a_4,\ldots,a_n)$ respectively.

$$\begin{aligned}
\mathcal{T}(a_4,\ldots,a_n) &= \left[1+(-1)^{a_{n-1}}+(-1)^{a_n}+(-1)^{a_{n-1}\oplus a_n\oplus 1}\right]2^{n-5}\delta_{\mathbf{0}}(\hat{a}). \\
\mathcal{W}_{\dot{g}}(a_4,\ldots,a_n) &= \mathcal{W}_{g(x_4,\ldots,x_{n-2},0,0)}(\hat{a}) + (-1)^{a_{n-1}}\mathcal{W}_{g(x_4,\ldots,x_{n-2},1,0)}(\hat{a}) + \\
&\quad + (-1)^{a_n}\mathcal{W}_{g(x_4,\ldots,x_{n-2},0,1)}(\hat{a}) + (-1)^{a_{n-1}\oplus a_n\oplus 1}\mathcal{W}_{g(x_4,\ldots,x_{n-2},1,1)}(\hat{a}),
\end{aligned}$$

where $\hat{a} = (a_4,\ldots,a_{n-2})\in\mathbb{F}_2^{n-5}$ and $\delta_{\mathbf{0}}(\hat{a}) = \delta_0(a_4)\cdots\delta_0(a_{n-2})$. Thus, we have

$$\begin{aligned}
&\mathcal{W}_{f_g}(a_1,\ldots,a_n) = \\
&= \left(1+(-1)^{a_2}+(-1)^{a_3}\right)\left[1+(-1)^{a_{n-1}}+(-1)^{a_n}+(-1)^{a_{n-1}\oplus a_n\oplus 1}\right]\times \\
&\times 2^{n-4}\delta_{\mathbf{0}}(\hat{a}) + 2(-1)^{a_2\oplus a_3}\Big[\mathcal{W}_{g(x_4,\ldots,x_{n-2},0,0)}(\hat{a}) + \\
&+(-1)^{a_{n-1}}\mathcal{W}_{g(x_4,\ldots,x_{n-2},1,0)}(\hat{a}) + (-1)^{a_n}\mathcal{W}_{g(x_4,\ldots,x_{n-2},0,1)}(\hat{a}) +
\end{aligned}$$

$$+(-1)^{a_{n-1}\oplus a_n\oplus 1}\mathcal{W}_{g(x_4,\ldots,x_{n-2},1,1)}(\hat{a})\Big].$$

Now taking into account that arguments x_{n-1}, x_n are not essential for $g(x_4,\ldots,x_{n-2},x_{n-1},x_n)\in\mathcal{F}_{n-3}$ we obtain

$$\begin{aligned}\mathcal{W}_{f_g}(a_1,\ldots,a_n) &= 2^{n-4}\big(1+(-1)^{a_2}+(-1)^{a_3}\big)\times\\ &\times\big[1+(-1)^{a_{n-1}}+(-1)^{a_n}+(-1)^{a_{n-1}\oplus a_n\oplus 1}\big]\delta_{\mathbf{0}}(\hat{a})+\\ &+2(-1)^{a_2\oplus a_3}\big[1+(-1)^{a_{n-1}}+(-1)^{a_n}+(-1)^{a_{n-1}\oplus a_n\oplus 1}\big]\times\\ &\times\mathcal{W}_g(\hat{a}).\end{aligned}$$

So for any $\hat{a}=(a_4,\ldots,a_{n-2})\in\mathbb{F}_2^{n-5}$ considering all possible values of $(a_2,a_3,a_{n-1},a_n)\in\mathbb{F}_2^4$ we can obtain the expression of $\mathcal{W}_{f_g}(a_1,a_2,a_3,\beta,a_{n-1},a_n)$ written in the proposition. □

Proposition 12. *For any Boolean function $g(x_4,\ldots,x_{n-2},x_{n-1},x_n)\in\mathcal{F}_{n-3}$ which arguments x_{n-1},x_n are not essential, the curvature of the function f_g constructed by (21) has the following upper bound*

$$\mathsf{curv}(f_g)\le 3\cdot 2^n+64\cdot\mathsf{curv}(g) \tag{22}$$

Proof. By definition we have $\mathsf{curv}(f_g)=\sum\limits_{a_1,\ldots,a_n\in\mathbb{F}_2}|\mathcal{W}_{f_g}(a_1,\ldots,a_n)|$. Now, using proposition 11 and the following property $|x+y|\le|x|+|y|$, which holds for any $x,y\in\mathbb{R}$ we obtain

$$\begin{aligned}\mathsf{curv}(f_g) &= \sum_{a_2,\ldots,a_n\in\mathbb{F}_2}|\mathcal{W}_{f_g}(1,a_2,\ldots,a_n)|\\ &\le \sum_{a_2,a_3,a_{n-1},a_n\in\mathbb{F}_2}|\mathcal{W}_{f_g}(1,a_2,a_3,0,\ldots,0,a_{n-1},a_n)|+\\ &+\sum_{\substack{a_2,a_3,\ldots,a_{n-2},a_{n-1},a_n\in\mathbb{F}_2\\ \exists\, a_i\neq 0,\, i\in\{2,\ldots,n-2\}}}|\mathcal{W}_{f_g}(1,a_2,\ldots,a_{n-1},a_n)|\le\\ &\le (6\cdot 2^{n-4}+4\cdot|\mathcal{W}_g(\mathbf{0})|)\cdot 4+(2\cdot 2^{n-4}+4\cdot|\mathcal{W}_g(\mathbf{0})|)\cdot 12+\\ &+\sum_{\substack{a_2,\ldots,a_{n-2}\in\mathbb{F}_2\\ \exists\, a_i\neq 0,\, i\in\{2,\ldots,n-2\}}}4\cdot 16\cdot|\mathcal{W}_g(a_2,\ldots,a_{n-2})|\le\\ &\le 3\cdot 2^n+64\cdot\mathsf{curv}(g).\end{aligned}$$

□

Also from the proof of proposition 11 we deduce that

$$\max_{a_1,a_2,\ldots,a_n\in\mathbb{F}_2} |\mathcal{W}_{f_g}(a_1,a_2,\ldots,a_n)| \geq 2\cdot 2^{n-4}+4\cdot \max_{a_4,\ldots,a_{n-2}\in\mathbb{F}_2} |\mathcal{W}_g(a_4,\ldots,a_{n-2})|,$$

hence the nonlinearity of $f_g \in \mathcal{F}_n$ can be upper bounded as follows $nl(f_g) \leq 2^{n-1} - 5\cdot 2^{n-4} + 4\cdot nl(g)$. If denote $e_1 = (1,0,\ldots,0)$ we obtain that $f_g(x)\oplus f_g(x\oplus e_1) = 1$, this means that e_1 is a linear structure for f_g and $\Delta_f(e_1) = \sum_{x\in\mathbb{F}_2^n}(-1)^{f_g(x)\oplus f_g(x\oplus e_1)} = -2^n$, then from relation $\max_{\mathbf{0}\neq u\in\mathbb{F}_2^n}|\Delta_{f_g}(u)| \leq 2^n$, we derive $\mathsf{ls}(f_g) = 2^{n-2} - \dfrac{1}{4}\max\limits_{\mathbf{0}\neq u\in\mathbb{F}_2^n}|\Delta_{f_g}(u)| = 0$.

n	$\mathsf{curv}(f_g)$	$nl(f_g)$	$d_{alg}(f_g)$
odd or even	$\leq (3\cdot 2^n$ $+64\cdot \mathsf{curv}(g))$	$\leq (2^{n-1} - 5\cdot 2^{n-4}$ $+4\cdot nl(g))$	$d_{alg}(g)+2$

n	$\mathsf{ls}(f_g)$
odd or even	0

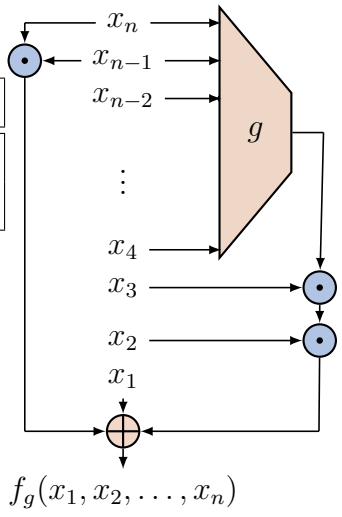

Figure 5: Cryptographic parameters of f_g and its high level representation.

The values of some cryptographic parameters of the class f_g and its high level representation are given Figure 5. As we can see the curvature of these functions is far from $2^{\frac{3n}{2}}$, moreover this class of function has nonzero linear structures and is not optimal in terms of such parameters as algebraic degree (in the case when arguments x_{n-1}, x_n are not essential for g) and nonlinearity but has the property of being perfectly balanced without right barrier. In addition, Boolean functions f_g can be used as building blocks for constructing perfectly balanced Boolean function without left and right barriers (see, [17, Theorem 4]). We leave the task of finding the exact value of nondegeneracy of functions f_g as a future work.

3 Variation of the curvature when changing randomly one (multiple) value(s) in the value vector of a Boolean function.

In this section for a given Boolean function $f \in \mathcal{F}_n$ we analyse the behavior of its curvature when changing randomly one (multiple) value(s) in the value vector v_f of f.

First of all, we shall fix an ordered family of all elements of the space $\mathbb{F}_2^n$, i.e, $\mathbb{F}_2^n = \{u_0, u_1, \ldots, u_{2^n-1}\} = \{(0,\ldots,0),(1,\ldots,0,0),\ldots,(1,\ldots,1)\}$. Changing randomly one value in v_f can be analytically expressed as adding to a given function f another Boolean function, which take the value 1 in those position of v_f where the bit was changed and 0 in the remaining values. The whole process can be synthesized using the well-known indicator function as $f \oplus \delta_{u_i}$, where $\delta_i(u) = \delta_{u_i}(u)$ has only one nonzero value at the point $u_i \in \mathbb{F}_2^n, i \in \{0,\ldots,2^n-1\}$.

The Walsh-Adamard transform for the Boolean function $f \oplus \delta_{u_i} \in \mathcal{F}_n$ has the following form

$$\begin{aligned}\mathcal{W}_{f\oplus\delta_{u_i}}(u) &= \sum_{\substack{x\in\mathbb{F}_2^n\\ x\neq u_i}} (-1)^{f(x)\oplus\langle u,x\rangle} - (-1)^{f(u_i)\oplus\langle u_i,u\rangle} = \\ &= \mathcal{W}_f(u) - 2\cdot(-1)^{f(u_i)\oplus\langle u_i,u\rangle}.\end{aligned}$$

Thus, $|\mathcal{W}_f(u)|-2 \leq |\mathcal{W}_{f\oplus\delta_{u_i}}(u)| \leq |\mathcal{W}_f(u)|+2$. If now selecting two vectors $u_i, u_j \in \mathbb{F}_2^n$ for which $f(u_i) \neq f(u_j)$, where $1 \leq i < j \leq 2^n$ we obtain by using recursively relation (23) the following equality

$$\mathcal{W}_{f\oplus\delta_{u_i}\oplus\delta_{u_j}}(u) = \mathcal{W}_f(u) - 2\cdot\left[(-1)^{f(u_i)\oplus\langle u_i,u\rangle} + (-1)^{f(u_j)\oplus\langle u_j,u\rangle}\right].$$

Hence, $|\mathcal{W}_f(u)|-4 \leq |\mathcal{W}_{f\oplus\delta_{u_i}\oplus\delta_{u_j}}(u)| \leq |\mathcal{W}_f(u)|+4$. In this way we have proved the following proposition.

Proposition 13. *Let f be a Boolean function of the set $\mathcal{F}_n$. Then*

1. *For any $u_i \in \mathbb{F}_2^n, i \in \{0,1,\ldots,2^n-1\}$,*

$$|\mathsf{curv}(f\oplus\delta_{u_i}) - \mathsf{curv}(f)| \leq 2\cdot 2^n. \tag{23}$$

2. *For some $u_i, u_j \in \mathbb{F}_2^n$ such that $f(u_i) \neq f(u_j)$, $1 \leq i < j \leq 2^n$,*

$$\left|\mathsf{curv}(f\oplus\delta_{u_i}\oplus\delta_{u_j}) - \mathsf{curv}(f)\right| \leq 4\cdot 2^n. \tag{24}$$

The second item of proposition 13 tell us that interchanging arbitrary values $0, 1$ in v_f the value of parameter $\mathsf{curv}(f \oplus \delta_{u_i} \oplus \delta_{u_j})$ may be increased which could be useful when searching highly nonlinear balanced Boolean functions $f \in \mathcal{F}_n$, maximizing the value of this parameter.

One other case in which the variation of $\mathsf{curv}(f)$, $f \in \mathcal{F}_n$ can be analysed when changing randomly one value in v_f was examined in [19] considering Boolean functions as points on the hypersphere in the Euclidean space. In this case we need introduce the following notation. For some function g, we denote $D(g) = \{f \in \mathcal{F}_n \mid \mathsf{sgn}(\mathcal{W}_f(u)) = (-1)^{g(u)} \text{ for all } u \in \mathbb{F}_2^n\}$, where the function $\mathsf{sgn}(\cdot)$ is the sign function of a number $r \in \mathbb{R}$, defined as follows

$$\mathsf{sgn}(r) = \begin{cases} 1, & \text{if } r > 0; \\ 0, & \text{if } r = 0; \\ -1, & \text{if } r < 0. \end{cases}$$

As the authors of [19, p. 53] stated , if $f \in D(g)$ for some Boolean function g, then the following relation holds

$$\mathsf{curv}(f \oplus \delta_{u_i}) = \mathsf{curv}(f) - 2 \cdot (-1)^{f(u_i)} \mathcal{W}_g(u_i). \tag{25}$$

In particular, the curvature remains unchanged if the function f varies at a point u_i at which $\mathcal{W}_g(u_i) = 0$, for some $i \in \{0, 1, \ldots, 2^n - 1\}$.

If $f = \tilde{\varphi}$ is a Bent function and $g = \tilde{\tilde{\varphi}} = \varphi$ — the dual of f, i.e., a Boolean function for which $\mathcal{W}_\varphi(u) = (-1)^{\tilde{\varphi}(u)} \cdot 2^{\frac{n}{2}}$ holds for all $u \in \mathbb{F}_2^n$, then from (25) we obtain $\mathsf{curv}(\tilde{\varphi} \oplus \delta_{u_i}) = 2^{\frac{3n}{2}} - 2 \cdot 2^{\frac{n}{2}}$ which means that for any change in the value vector $v_{\tilde{\varphi}}$, the curvature of the resulting Boolean function $\tilde{\varphi} \oplus \delta_{u_i}$ decrease by a factor of $2^{\frac{n}{2}}$.

4 Extending the curvature to S-Boxes

In this section we shall extend the curvature parameter to the case of n-bit S-Boxes, i.e., vectorial Boolean functions from $\mathbb{F}_2^n$ to $\mathbb{F}_2^n$, where typically $n = 3, 4, 8$. Also we investigate the behavior of some parameters and matrices connected with the curvature of Boolean functions when studying the nonlinear components of actual symmetric cryptographic primitives.

Recall that we can fix an ordered family of all elements of the space $\mathbb{F}_2^n$, i.e, $\mathbb{F}_2^n = \{u_0, u_1, \ldots, u_{2^n-1}\} = \{(0, \ldots, 0), (1, \ldots, 0, 0), \ldots, (1, \ldots, 1)\}$ and consider an n-bit S-Box $\mathcal{S}$ as a vector of Boolean functions:

$$\mathcal{S} = (s_1, \ldots, s_n), s_i \in \mathcal{F}_n, i = 1, \ldots, n. \tag{26}$$

Functions s_i are called coordinate Boolean functions of the S-Box $\mathcal{S}$ and it is well-known (see, [5, p. 76]) that most of the desirable cryptographic properties of $\mathcal{S}$ can be defined in terms of their non-trivial linear combinations (also-called S-Box component Boolean functions), denoted by $\mathcal{S}_b$, and defined as $\mathcal{S}_b = \langle b, \mathcal{S}\rangle = b^{(1)}s_1 \oplus \ldots \oplus b^{(n)}s_n$ where $\mathbf{0} \neq b = (b^{(1)}, \ldots, b^{(n)}) \in \mathbb{F}_2^n, b^{(i)} \in \mathbb{F}_2, i \in \{1, \ldots, n\}$.

For $a, b \in \mathbb{F}_2^n$ the Walsh transform $\mathcal{W}_{\mathcal{S}}(a, b)$ of an n-bit S-Box $\mathcal{S}$ is defined as $\mathcal{W}_{\mathcal{S}}(a, b) = \sum_{x\in\mathbb{F}_2^n}(-1)^{\langle b,\mathcal{S}(x)\rangle\oplus\langle a,x\rangle} = \sum_{x\in\mathbb{F}_2^n}(-1)^{\mathcal{S}_b(x)\oplus\langle a,x\rangle} = \mathcal{W}_{\mathcal{S}_b}(a)$ and the nonlinearity of $\mathcal{S}$, denoted by $\mathcal{NL}(\mathcal{S})$, is defined as $\mathcal{NL}(\mathcal{S}) = \min_{\mathbf{0}\neq b\in\mathbb{F}_2^n} nl(\mathcal{S}_b)$. Naturally, in the case of nonlinear transformation we are interested in the behavior of the curvature of its non-trivial linear combinations and also in analysing the following parameters

$$\mathsf{curv}_{min}(\mathcal{S}) = \min_{\mathbf{0}\neq b\in\mathbb{F}_2^n} \mathsf{curv}(\mathcal{S}_b), \tag{27}$$

$$\mathsf{curv}_{max}(\mathcal{S}) = \max_{\mathbf{0}\neq b\in\mathbb{F}_2^n} \mathsf{curv}(\mathcal{S}_b), \tag{28}$$

$$\rho_{\mathsf{curv}}(\mathcal{S}) = \mathsf{curv}_{max}(\mathcal{S}) - \mathsf{curv}_{min}(\mathcal{S}). \tag{29}$$

Parameters defined by (27),(28) are called here the *minimum (maximum) curvature* of the n-bit S-Box $\mathcal{S}$ and can be used to characterize "how close" is an S-Box component Boolean function to being linear or Bent. The parameter $\rho_{\mathsf{curv}}(\mathcal{S})$ characterize "how close" is the value $\mathsf{curv}_{min}(\mathcal{S})$ with respect to $\mathsf{curv}_{max}(\mathcal{S})$.

For a given n-bit S-Box $\mathcal{S}$ if $\mathsf{curv}_{min}(\mathcal{S}) = O(2^n)$, when $n \to \infty$ then we can expect a lower value of its nonlinearity, if $\mathsf{curv}_{min}(\mathcal{S}) = O(2^{\frac{3n}{2}})$ when $n \to \infty$ and at the same time $\rho_{\mathsf{curv}}(\mathcal{S})$ is very close to zero, then we can expect a higher value of the nonlinearity of $\mathcal{S}$.

For a natural number $n = 2^{2k}$, the *curvature matrix* of a nonlinear transformation $\mathcal{S}$, denoted by $\mathcal{M}_{\mathsf{curv}}(\mathcal{S})$, is defined as follows

$$\mathcal{M}_{\mathsf{curv}}(\mathcal{S}) = \Big\| m_{i,j} \Big\|_{i,j\in\mathbb{Z}_{2^k}}, \tag{30}$$

where $m_{i,j} = \mathsf{curv}\Big(\mathcal{S}_{u_{i\cdot 2^k+j}}\Big) = \mathsf{curv}\Big(\langle u_{i\cdot 2^k+j}, \mathcal{S}\rangle\Big)$. Obviously, $\mathcal{M}_{\mathsf{curv}}(\mathcal{S})$ defines a $2^k \times 2^k$ matrix over the set of positive integers — $\mathbb{Z}_+$ and parameters defined by (27),(28) can be expressed using this matrix as follows, $\mathsf{curv}_{min}(\mathcal{S}) = \min\limits_{(0,0)\neq(i,j)\in\mathbb{Z}_{2^k}\times\mathbb{Z}_{2^k}} m_{i,j}$ and $\mathsf{curv}_{max}(\mathcal{S}) = \max\limits_{(0,0)\neq(i,j)\in\mathbb{Z}_{2^k}\times\mathbb{Z}_{2^k}} m_{i,j}$.

For a Boolean function $f \in \mathcal{F}_n$ and any vector $u \in \mathbb{F}_2^n$, we call *derivative in the direction* u (or with the input difference u) of f, the Boolean function $D_u f(x) = f(x) \oplus f(x \oplus u)$.

It is well-known that there exist relations between the nonlinearity and the derivatives of Boolean functions [5, p. 82]. For a nonlinear transformation $\mathcal{S}\colon \mathbb{F}_2^n \to \mathbb{F}_2^n$, we call the *curvature matrix of derivatives* the following $2^n \times 2^n$ matrix defined over $\mathbb{Z}_+$

$$\mathcal{D}_{\mathsf{curv}}(\mathcal{S}) = \Big\| d_{i,j} \Big\|_{i,j \in \mathbb{Z}_{2^n}}, \tag{31}$$

where $d_{i,j} = \mathsf{curv}\Big(D_i\mathcal{S}_j\Big) = \sum\limits_{u \in \mathbb{F}_2^n} |\mathcal{W}_{D_i\mathcal{S}_j}(u)|$.

For any $u \in \mathbb{F}_2^n$, let us introduce the following sets $\mathcal{L}_0(u) = \{x \in \mathbb{F}_2^n \mid \langle u, x\rangle = 0\}$, $\mathcal{L}_1(u) = \{x \in \mathbb{F}_2^n \mid \langle u, x\rangle = 1\}$. Then, coefficients $d_{i,j}, i, j \in \{0, 1, \ldots, 2^n - 1\}$ can be redefined as follows

$$d_{i,j} = \sum_{u \in \mathbb{F}_2^n} \Big| \mathcal{W}_{(D_i\mathcal{S}_j)_{\mathcal{L}_0(u)}}(\mathbf{0}) - \mathcal{W}_{(D_i\mathcal{S}_j)_{\mathcal{L}_1(u)}}(\mathbf{0}) \Big|, \tag{32}$$

where $\mathcal{W}_{f_\mathcal{A}}(a) = \sum_{x \in \mathcal{A}} (-1)^{f_\mathcal{A}(x) \oplus \langle a, x\rangle}$ is called the incomplete Walsh-Adamard transform of partially defined Boolean function $f_\mathcal{A}, \mathcal{A} \subseteq \mathbb{F}_2^n$ (see, for example, [18, p. 273]).

From relation (32), we can rewrite coefficients $d_{i,j}$ as follows

$$d_{i,j} = \sum_{u \in \mathbb{F}_2^n} \left| \sum_{x \in \mathcal{L}_0(u)} (-1)^{\mathcal{S}_j(x) \oplus \mathcal{S}_j(x \oplus i)} - \sum_{x \in \mathcal{L}_1(u)} (-1)^{\mathcal{S}_j(x) \oplus \mathcal{S}_j(x \oplus i)} \right|. \tag{33}$$

If define the incomplete autoccorelation function of partially defined Boolean function $f_\mathcal{A}, \mathcal{A} \subseteq \mathbb{F}_2^n$ as $\Delta_{f_\mathcal{A}}(a) = \sum_{x \in \mathcal{A}} (-1)^{f_\mathcal{A}(x) \oplus f_\mathcal{A}(x \oplus a)}$, we can see that coefficients $d_{i,j}$ can be expressed in terms of absolute values of the differences between autocorrelation functions of non-trivial linear combinations $\mathcal{S}_j$ defined over sets $\mathcal{L}_i(u), i = 0, 1$, that is

$$d_{i,j} = \sum_{u \in \mathbb{F}_2^n} \Big| \Delta_{(\mathcal{S}_j)_{\mathcal{L}_0(u)}}(i) - \Delta_{(\mathcal{S}_j)_{\mathcal{L}_1(u)}}(i) \Big|. \tag{34}$$

We shall use ours matrices $\mathcal{M}_{\mathsf{curv}}(\mathcal{S}), \mathcal{D}_{\mathsf{curv}}(\mathcal{S})$ when analysing the behavior of the curvature for component Boolean functions of a given S-Box $\mathcal{S}$.

4.1 Case studies: Some known S-Boxes used in block ciphers

In this section we perform an analysis of those parameters related to the curvature of an n-bit S-Box, introduced in the previous section. As case

studies, we investigate the behavior of the curvature parameter of 8-bit nonlinear components used in actual block ciphers such as AES [21], ForkAE [11], Picaro [22], Kuznyechik [9] and Khazad [2].

In [6] were computed the middle values $\theta_n^f = \frac{1}{\#\mathcal{F}_n}\sum_{f\in\mathcal{F}_n}\mathsf{curv}(f), \theta_n^g = \frac{1}{\#\mathcal{G}_n}\sum_{g\in\mathcal{G}_n}\mathsf{curv}(g)$, where $\mathcal{G}_n$ is the sub-class of $\mathcal{F}_n$ containing all balanced Boolean functions. When $n = 8$, these magnitudes are equals to $\theta_8^f = 3264,945$ and $\theta_8^g = 3245,845$ respectively and will be used as useful quantities in the study of the curvature of some actual S-Boxes. To better illustrate the behavior of matrices $\mathcal{M}_{\mathsf{curv}}(\mathcal{S})$, $\mathcal{D}_{\mathsf{curv}}(\mathcal{S})$ when looking at their coefficients, we shall use, instead of looking at matrices itself, its visual representations[c] (similar to [3, 30]) with the library Matplotlib provided by SAGE [23], since it has been shown the usefulness of this approach to find some unexpected patterns of a given S-Box.

It should be noted that, in what follows when analysing the visual representation of matrix $\mathcal{M}_{\mathsf{curv}}(\mathcal{S})$ for any n-bit S-Box $\mathcal{S}$, the coefficient $m_{0,0} = \mathsf{curv}(\mathcal{S}_{u_0})$ will not be plotted because it always take the value 2^n.

4.1.1 AES S-Box

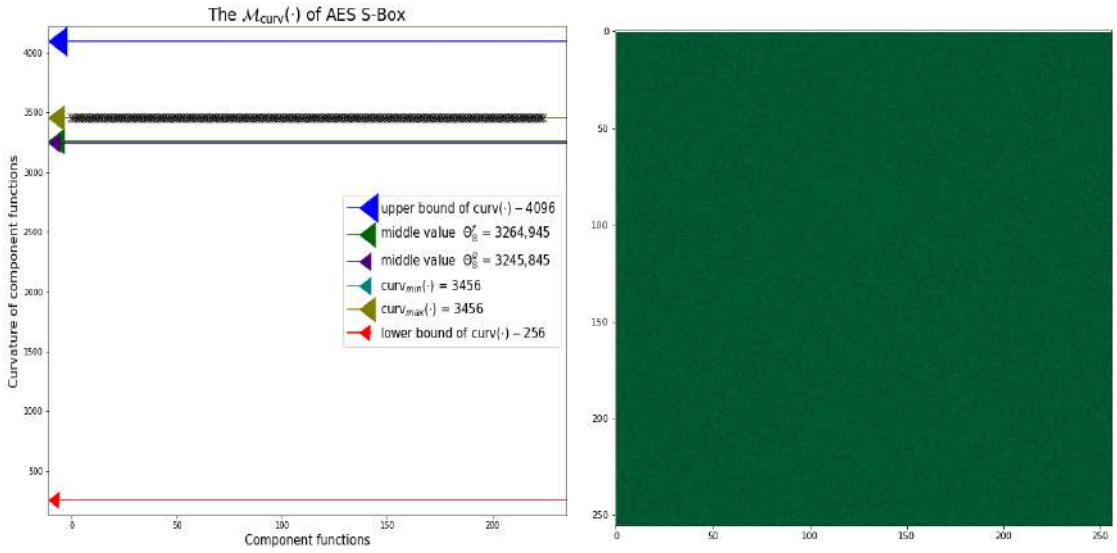

Figure 6: Visual representations of matrices $\mathcal{M}_{\mathsf{curv}}(\mathcal{S}_{AES})$ and $\mathcal{D}_{\mathsf{curv}}(\mathcal{S}_{AES})$.

The 8-bit permutation of the block cipher AES, denoted here by $\mathcal{S}_{AES}$, was designed by using the finite field inversion function having $\mathcal{NL}(\mathcal{S}_{AES}) = 112$ and coordinate functions with the following curvatures $(3456, 3456, 3456, 3456, 3456, 3456, 3456, 3456)$. Moreover, $\mathsf{curv}_{min}(\mathcal{S}_{AES}) =$

[c]The visualization of $\mathcal{M}_{\mathsf{curv}}(\mathcal{S})$ is performed by using the function matrix plot of SAGE. As matrix $\mathcal{D}_{\mathsf{curv}}(\mathcal{S})$ is too big to be displayed when $n = 8$, the visual representation of $\mathcal{D}_{\mathsf{curv}}(\mathcal{S})$ it is simply a picture where each curvature value has a color associated to it.

$\mathsf{curv}_{max}(\mathcal{S}_{AES}) = 3456, \rho_{\mathsf{curv}}(\mathcal{S}_{AES}) = 0$ which means that there is no variation in the curvature of its component functions and as we can see in the left side of Figure 6 these values are slightly far from the middles values θ_8^f and θ_8^f. When displaying the matrix $\mathcal{D}_{\mathsf{curv}}(\mathcal{S}_{AES})$ we don't find any patterns in the graphical visualization of this matrix.

4.1.2 ForkAE S-Box

The S-Box used in the authenticated cipher ForkAE[d] (see, for example, [11]) is a permutation on $\mathbb{F}_2^8$ and according to designers arguments (see, [4]) it was designed to provides a good tradeoff between security and area cost. However, the 8-bit coordinates functions of this S-Box have the following curvatures $(1152, 512, 512, 1152, 512, 512, 1536, 1920)$ and $\mathcal{NL}(\mathcal{S}_{ForkAE}) = 64$, $\mathsf{curv}_{min}(\mathcal{S}_{ForkAE}) = 512, \mathsf{curv}_{max}(\mathcal{S}_{ForkAE}) = 3072, \rho_{\mathsf{curv}}(\mathcal{S}_{ForkAE}) = 2560$.

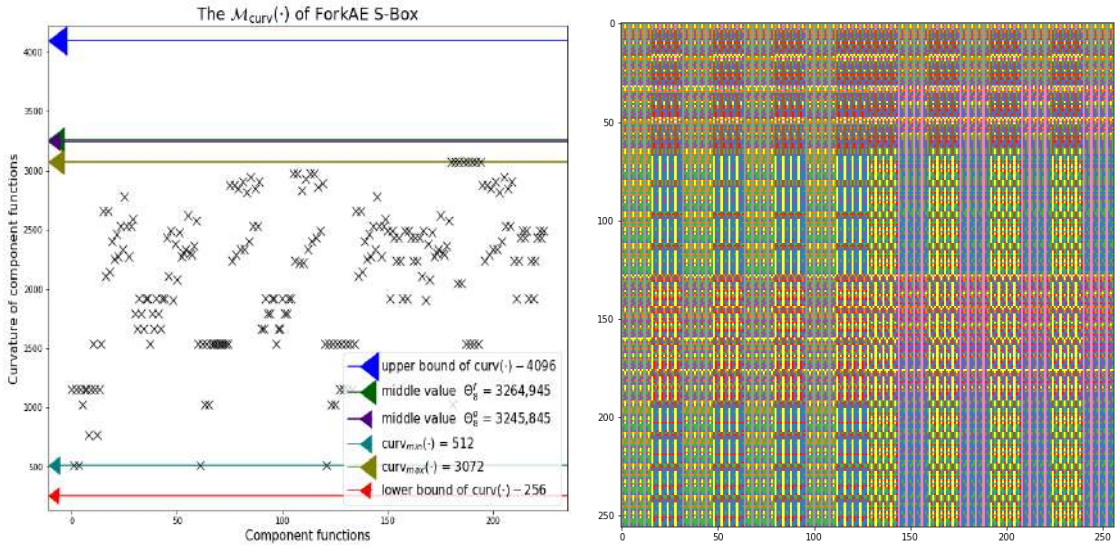

Figure 7: Visual representations of matrices $\mathcal{M}_{\mathsf{curv}}(\mathcal{S}_{Fork})$ and $\mathcal{D}_{\mathsf{curv}}(\mathcal{S}_{Fork})$.

It can be observed from the left side of Figure 7 that there are a few component functions having curvatures close to the expected middle values, but there are many non-trivial linear combinations which have curvatures much lower than θ_8^f and θ_8^g, in fact because of the value $\mathsf{curv}_{min}(\cdot) = 512$ is close to the lower bound we can not expect a high nonlinearity of this nonlinear bijective transformation. Also, when examining $\mathcal{D}_{\mathsf{curv}}(\mathcal{S}_{Fork})$ we can find a lot of coefficients with low values, which are responsible for the patterns detected in the visual representation of this matrix.

[d]Currently ForkAE is a 2nd round candidate in the NIST lightweight authenticated encryption standardization process.

4.1.3 Picaro S-Box

The 8-bit nonlinear transformation of Picaro block cipher, denoted here by $\mathcal{S}_{Picaro}$, is not bijective, however according to designers philosophy (see, [22]), because of its low implementation complexity, has an advantage when protecting the whole cipher against side-channel attacks. This S-Box have

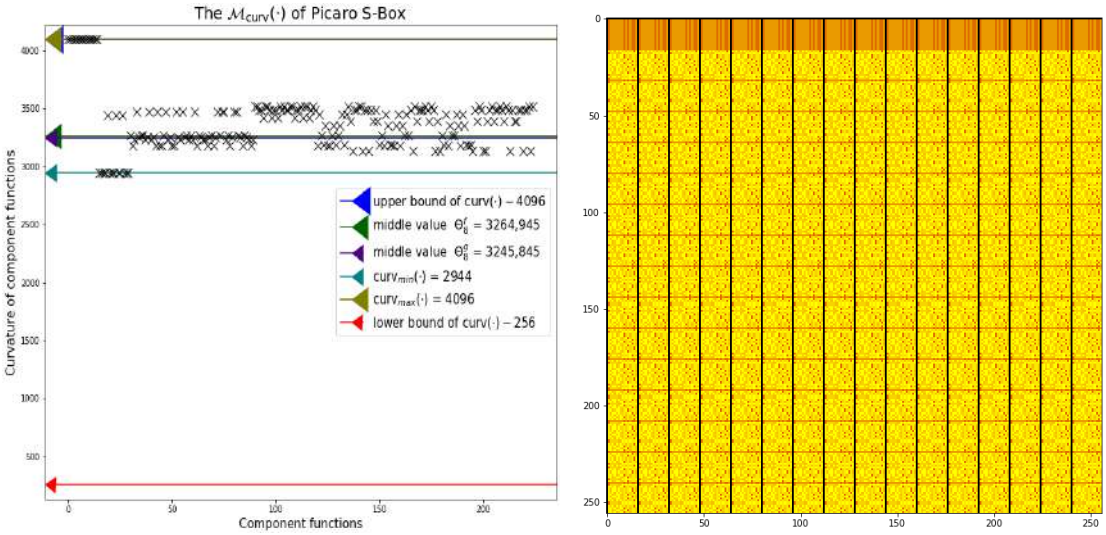

Figure 8: Visual representations of matrices $\mathcal{M}_{\mathsf{curv}}(\mathcal{S}_{Picaro})$ and $\mathcal{D}_{\mathsf{curv}}(\mathcal{S}_{Picaro})$.

$\mathcal{NL}(\mathcal{S}_{Picaro}) = 94$ and coordinate functions with the following curvatures $(4096, 4096, 4096, 4096, 3352, 3424, 3424, 3440)$, moreover $\mathsf{curv}_{min}(\mathcal{S}_{Picaro}) = 2944, \mathsf{curv}_{max}(\mathcal{S}_{Picaro}) = 4096, \rho_{\mathsf{curv}}(\mathcal{S}_{Picaro}) = 1152$ and as we can see in the left side Figure 8 there are several component function with maximal possible curvature which is normal because this nonlinear transformation was obtained by concatenating the outputs of a Bent function and of another function. Except 12 component functions with curvatures equal to 2944, there are some non-trivial linear combinations which are not so far from the expected middle values θ_8^f and θ_8^g. The visual representation of $\mathcal{D}_{\mathsf{curv}}(\mathcal{S}_{Picaro})$ displayed in the right side Figure 8 has strong patterns which existence is due to low values (close to the lower bound of $\mathsf{curv}(\cdot)$) of the curvatures of derivatives $D_i\mathcal{S}_j, i, j \in 0, \ldots, 255$, where $\mathcal{S} = \mathcal{S}_{Picaro}$.

4.1.4 Kuznyechik S-Box

The Kuznyechik S-Box, denoted here by $\mathcal{S}_{Kuz}$, has the so-called TU-decomposition (see, [3]) and $\mathcal{NL}(\mathcal{S}_{Kuz}) = 100$. The eight coordinate Boolean functions exhibits the following curvatures $(3248, 3320, 3840, 3200, 3232, 3344, 3224, 3200)$ and for this nonlinear bijective transformation we have $\mathsf{curv}_{min}(\mathcal{S}_{Kuz}) = 2992, \mathsf{curv}_{max}(\mathcal{S}_{Kuz}) = 3840$

and $\rho_{\mathsf{curv}}(\mathcal{S}_{Kuz}) = 848$.

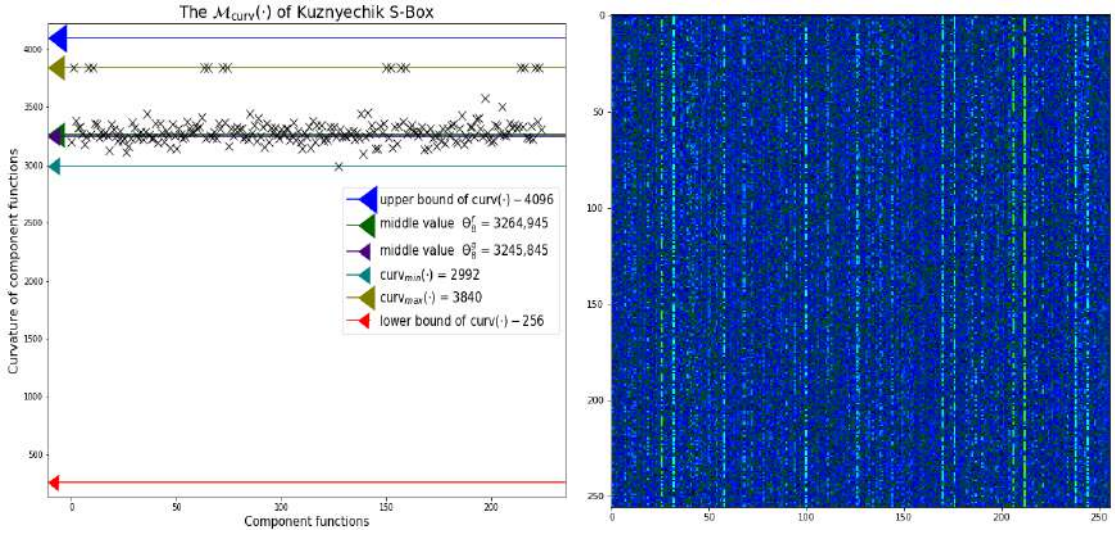

Figure 9: Visual representations of matrices $\mathcal{M}_{\mathsf{curv}}(\mathcal{S}_{Kuz})$ and $\mathcal{D}_{\mathsf{curv}}(\mathcal{S}_{Kuz})$.

It can be observed in left side of Figure 9 that there are several component functions (15 of them, to be exact) with curvatures equal to 3840[e] which despite the decomposition founded in [3], indicate the presence of component functions living near the set of Bent functions. Except one isolated function having a curvature value equal to 2992, the curvatures of the remaining component functions are very close to the expected middle values θ_8^f and θ_8^g. Interestingly when visualizing the matrix $\mathcal{D}_{\mathsf{curv}}(\mathcal{S}_{Kuz})$ the same patterns founded in [30] for the so-called column frequency matrix of this S-Box are presented in the representation, displayed in the right side of Figure 9.

4.1.5 Khazad S-Box

The S-Box of Khazad block cipher [2], denoted here by $\mathcal{S}_{Khazad}$, is an involution over $\mathbb{F}_2^8$ with $\mathcal{NL}(\mathcal{S}_{Khazad}) = 96$ and has an SPN structure, where the small nonlinear components selected in this design were chosen by using some form of hill climbing among the set of the differentially 4-uniform permutation with best non-linearity. Coordinate functions of this involution have the following curvatures $(3256, 3280, 3352, 3240, 3176, 3208, 3264, 3256)$ and $\mathsf{curv}_{min}(\mathcal{S}_{Khazad}) = 3088, \mathsf{curv}_{max}(\mathcal{S}_{Khazad}) = 3400$ and $\rho_{\mathsf{curv}}(\mathcal{S}_{Khazad}) = 312$.

It can be observed in left side of Figure 10 that almost all component functions have curvatures very close to the middle values θ_8^f and θ_8^g and we

[e]This value, when $n = 8$, can be expressed as $2^{\frac{3n}{2}} - 2^n$, and in this case coincide with the curvature parameter of some well-known classes studied in this work, which use Bent function as building blocks.

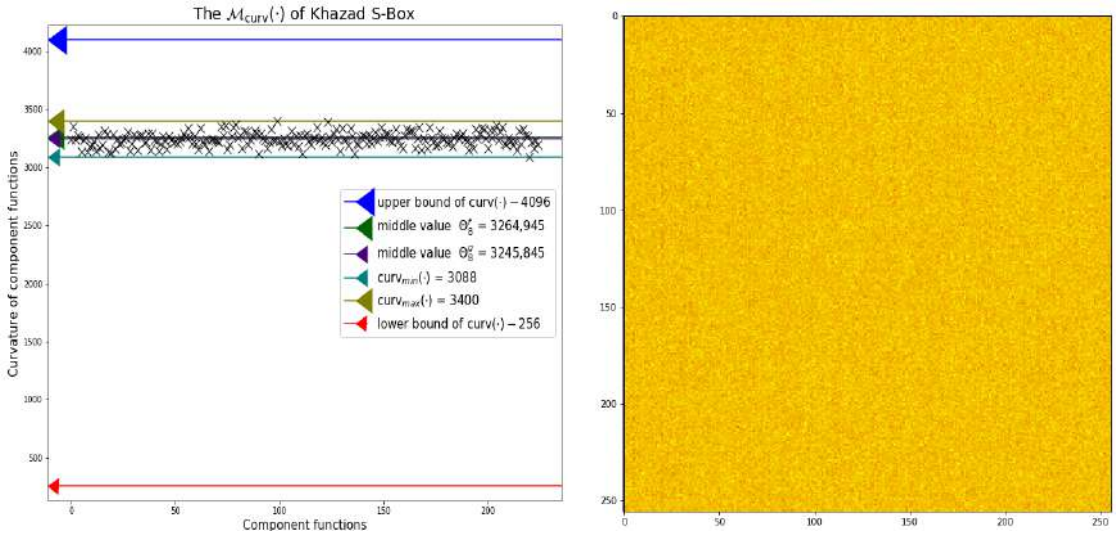

Figure 10: Visual representations of matrices $\mathcal{M}_{\mathsf{curv}}(\mathcal{S}_{Khazad})$ and $\mathcal{D}_{\mathsf{curv}}(\mathcal{S}_{Khazad})$.

can not find any pronounced patterns in the visual representation of matrix $\mathcal{D}_{\mathsf{curv}}(\mathcal{S}_{Khazad})$ displayed in the right side of Figure 10.

5 Conclusion

In this work we have obtained exact formulas and bounds for the curvature and nondegeneracy of some classes of Boolean functions and results shows that there exist several classes having a curvature values very close to $2^{\frac{3n}{2}}$, but some of them exhibits low values of nondegeneracy which implies that these functions are susceptible to a method of analysis based on algebraically degenerate approximations [1]. In addition, we have determined other relevant cryptographic parameters such as non-linearity, algebraic degree and distance to linear structures. Also, we have analysed the variation of the curvature when changing randomly one or multiples values in the output of a Boolean function.

Because of their representations as a vector of Boolean functions, we have extended the curvature parameter to S-Boxes, introducing the minimum (maximum) curvature and two matrices related with this parameter, which are useful when studying the behavior of the curvature of component functions presented in the nonlinear layers of actual symmetric algorithms.

There are some questions related to the curvature of Boolean functions which looks very interesting, among them:

– Can we connect parameters $\mathsf{curv}_{min}(\cdot), \mathsf{curv}_{max}(\cdot)$ and matrices $\mathcal{M}_{\mathsf{curv}}(\cdot), \mathcal{D}_{\mathsf{curv}}(\cdot)$ with some methods of analysis of block ciphers?

– Is it possible to use parameters $\mathsf{curv}_{min}(\cdot)$, $\mathsf{curv}_{max}(\cdot)$ and $\rho_{\mathsf{curv}}(\cdot)$ in some heuristic approach for designing highly nonlinear S-Boxes?

Acknowledgements. The author would like to thank Oleg V. Kamlovskii for useful comments and valuable observations, which helped to improve the final version of this article.

References

[1] Alekseev E. K., "Filtering generator attacks with function close to algebraically degenerate", *Collected Articles of Young Scientists of the Faculty of MVK at the Moscow State University*, № 8 (2011), 19-32, https://cs.msu.ru/science/smu/activity/proceedings.

[2] Barreto, P., Rijmen, V., *The Khazad Legacy-Level Block Cipher*, Submission to the NESSIE Project.

[3] Biryukov, A., Perrin L., and Udovenko A., "Reverse engineering the S-Box of streebog, kuznyechik and STRIBOBr1", *LNCS*, EUROCRYPT 2016., **9665**, ed. Marc Fischlin and Jean-Sébastien Coron, Springer, Berlin, Heidelberg, 2016, 372-402.

[4] Beierle C. et al., "The SKINNY Family of Block Ciphers and Its Low-Latency Variant MANTIS", *LNCS*, CRYPTO 2016., **9815**, ed. Robshaw M., Katz J., Springer, Berlin, Heidelberg, 2016, 123–153.

[5] Carlet C., *Boolean Functions for Cryptography and Coding Theory*, Cambridge, Cambridge University Press, 2021.

[6] De la Cruz Jiménez, R. A., Kamlovskiy, O. V, "The sum of modules of Walsh coefficients of Boolean functions", *Discretnaya Matematika*, **26**:5 (2016), 259–272.

[7] De la Cruz Jiménez, R. A., "Constructing permutations, involutions and orthomorphisms with almost optimal cryptographic parameters", In: Pre-proceedings of CTCrypt'20-Dorokhovo, Ruza District, Moscow Region, Russia, 2020, 115-148.

[8] Dobbertin H., "Construction of Bent Functions and Balanced Boolean Functions with High Nonlinearity", *FSE*, 1994, 61–74.

[9] Dygin D. M., Lavrikov I. V., Marshalko G. B. , Rudskoy V. I., Trifonov D. I., Shishkin V. A., "On a new Russian Encryption Standard", *Mat. Vopr. Kriptogr.*, **6**:2 (2015), 29–34.

[10] Evertse, J. H., "Linear structures in block ciphers", *LNCS*, EUROCRYPT 1987, **304**, 1988, 249–266.

[11] E. Andreeva, V. Lallemand, A. Purnal, R. Reyhanitabar, A. Roy and D. Vizár., *Forkcipher: a New Primtive for Authenticated Encryption of Very Short Messages*, Cryptology ePrint Archive, Report 2019/1004 https://eprint.iacr.org/2019/1004.

[12] Fedorov S. N., "On a new classification of Boolean functions", *Mat. Vopr. Kriptogr.*, **10**:2 (2019), 159–168.

[13] Golić J.D., "On the security of nonlinear filter generators.", *LNCS*, FSE 1996., **1039**, ed. Gollmann D., Springer, Berlin, Heidelberg, 1996.

[14] Kamlovskiy, O. V., "The number of occurrences of elements in the output sequences of filter generators", *Prikladnaya diskretnaya matematika*, **3**:21 (2013), 11–25.

[15] Kamlovskiy, O. V, "Estimating the number of solutions of systems of nonlinear equations with linear recurring arguments by the spectral method", *Discretnaya Matematika*, **28**:2 (2016), 27–43.

[16] Kamlovskiy, O. V, "The sum of modules of Walsh coefficients of some balanced Boolean functions", *Mat. Vopr. Kriptogr.*, **8**:4 (2017), 75–98.

[17] Logachev O. A., Smyshlyaev S. V., Yashenko V. V., "New methods of investigation of perfectly balanced Boolean functions", *Discretnaya Matematika*, **21**:2 (2009), 51–74.

[18] Logachev O. A., Sal'nikov A. A., Smyshlyaev S. V., Yashenko V. V., *Boolean functions in coding theory and cryptology*, URSS, Moscow, Russia, 2015, (in Russian) 576 p.

[19] Logachev O. A., Fedorov S. N., Yashenko V. V., "Boolean functions as points on the hypersphere in the Euclidean space", *Discretnaya Matematika*, **30**:1 (2018), 39–55.

[20] Logachev O. A., Fedorov S. N., Yashenko V. V., "On Δ-equivalence of Boolean functions", *Discretnaya Matematika*, **30**:4 (2018), 29–40.

[21] NIST., *Advanced Encryption Standard. Federal Information Processing Standard*, (FIPS) 197, November 2001.

[22] Piret G., Roche T., Carlet C., "PICARO – A Block Cipher Allowing Efficient Higher-Order Side-Channel Resistance", *LNCS*, ACNS 2012., **7341**, ed. Bao F., Samarati P., Zhou J., Springer, Berlin, Heidelberg, 2012, 311–328.

[23] *Sage Mathematics Software (Version 8.1)*, 2018, `http://www.sagemath.org`.

[24] Smyshlyaev S. V., "Constructing classes of perfectly balanced Boolean functions without barriers", *Prikladnaya Discretnaya Matematika*, 2010, 41–50.

[25] Smyshlyaev S. V., "Barriers of of perfectly balanced Boolean functions", *Discretnaya Matematika*, **22**:2 (2010), 66–79.

[26] Smyshlyaev S. V., "Boolean functions without prediction", *Discretnaya Matematika*, **23**:1 (2011), 102–118.

[27] Smyshlyaev S. V., "Locally Invertible Boolean Functions", *Prikladnaya Discretnaya Matematika*, **4(14)** (2011), 11–21.

[28] Smyshlyaev S. V., "Perfectly balanced k-valued functions and the Golić condition", *Discretnaya Matematika*, **25**:1 (2013), 63–75.

[29] Rueppel R.A., *Analysis and design of stream ciphers.*, Springer-Verlag, Berlin Heidelberg, 1986, 244 p..

[30] Udovenko, A., *Design and Cryptanalysis of symmetric-key algorithms in black and white-box models, phdthesis*, 2019, `http://hdl.handle.net/10993/39350`.

On the Properties of Some Sequences Generated by Shift Registers and Latin Squares

Ramses Rodriguez Aulet and Adrián Alfonso Peñate

Institute of Cryptography, University of Havana, Cuba
ramsesrusia@yahoo.com, yamilkate@infomed.sld.cu

Abstract

In this paper we will analyze the properties of some sequences generated by linear shift registers with primitive characteristic polynomial and Latin squares. For this particular kind of sequences it is theoretically computed its period and distribution as well as other parameters of interest, and practically we show the results on entropy and distribution for particular instances. In addition, a new method for the construction of Latin squares which can be used in the generation of such sequences is presented.

Keywords: Latin square, shift register, period, distribution, entropy.

1 Introduction

Generating good randomness is an important part of many cryptographic operations, because even the simplest cryptosystems use data that should be unpredictable to attackers. A widely used method to ensure randomness is the generation of pseudo-random sequences by deterministic mechanisms, which are used as seeds, keys, initialization vectors, nonces, etc. for cryptographic applications. Several statistical tests are applied to pseudo-random sequences to attempt to evaluate its quality, mostly based on the 3 classic metrics known as Golomb's Randomness Postulates [1].

It is extremely important that pseudo-random sequences look as much as possible to have a truly random behavior, this way the real distribution of its elements needs to be close to the uniform distribution. In this sense Linear Congruential Generators [1] with the appropriate parameters achieve highly pseudo-random sequences. An interesting problem is the generation of pseudo-random sequences with uniformly distributed elements that appear more times than in sequences generated by a linear congruential.

Another common way to generate good pseudo-random sequences is using a Linear Feedback Shift Register with primitive polynomial [2]; however, in

this case sequences where the value 1 appears once more than the value 0 are obtained, as for example in the family of generators WG [3]. In this paper we combine shift registers and Latin squares to construct a deterministic function that can be used in the generation of pseudo-random sequences with uniformly distributed elements. There are several ways to construct sequences such that its elements are uniformly distributed; a well known strategy is to use a linear congruential [1] and another alternative is presented in [4, 5] in which cases the sequences are constructed on a ring.

Our advantage over the aforementioned methods is that we generate the sequences over a finite field and the elements appear uniformly distributed more times than in the sequences constructed by linear generators.

The rest of this paper is organized as follows. In section 2 we present the deterministic mechanism used to generate the sequences of our attention and we establish some approaches to evaluate the quality of such sequences. In section 3 some theoretical results related to period, distribution and correlation are presented while section 4 shows practical results about entropy and distribution. In addition, a new method for constructing a class of Latin squares which can be used in the generation of the sequences of our attention is proposed in section 5. The paper finish in section 6 with the conclusions.

2 Sequences and evaluation criteria

Let $P = GF(q)$ be the finite field of q elements where $q = q_1^t$ for some prime q_1 and $t \in \mathbb{N}$. Let P^* and $F(x)$ be the multiplicative group of P and a primitive polynomial of degree k in $P[x]$.

Consider $\pi_0, \ldots, \pi_{q-1}$ different permutations in the symmetric group S_q that form a $q \times q$ Latin square [6], then using the primitive polynomial $F(x)$ we construct a filter generator [7] in such a way that the output sequence v is determined by the rule R_1 as following

$$v(i) = \pi_{i \bmod q}[c_0u(i) + \cdots + c_{k-1}u(i+k-1)] + c_ku(i) + \cdots + c_{2k-1}u(i+k-1)$$

where u is the linear recursive sequence associated to $F(x)$, $c_j \in P^*$ for all $0 \leq j \leq k-1$ and $c_j \in P$ for all $k \leq j \leq 2k-1$.

On the other hand let $\gamma \in P$ be a primitive element, then for $a \in P$ we can form a Latin square

$$\pi_i(x) = ax + \gamma^i(1 - \delta_{i,0})$$

where $\delta_{i,0} = 1$ if $i = 0$ else $\delta_{i,0} = 0$. Here v is determined by the rule

$$v(i) = (ac_0 + c_k)u(i) + \cdots + (ac_k + c_{k-1})u(i+k-1) + \gamma^{i \bmod q} \; \forall i \in \mathbb{N}$$

and there must be an index $j \in \{0, \ldots, k-1\}$ such that $(ac_j + c_{j+k}) \neq 0$. If this condition is not satisfied then always $v(i) = \gamma^{i \bmod q}$ and the two sequences v and u are independent. That sequences are not of our interest.

2.1 How to evaluate v ?

To evaluate the quality of the sequences of our interest, generated by the filter generator presented above, different approaches can be used. In this paper are analyzed the period, distribution, correlation and entropy of v, the sequence generated by means of the rule R_1, for which we will use the next measures where $T(v)$ denotes the period of v, $\vec{z} = (z_1, z_2)$ and $z, z_1, z_2 \in P$

$$N_v(z) = |\{0 \leqslant i \leqslant T(v) - 1 : v(i) = z\}| \quad (1)$$

$$N(u, v) = |\{0 \leqslant i \leqslant T(v) - 1 : v(i) = u(i)\}| \quad (2)$$

$$N(\vec{z}, u, v) = |\{0 \leqslant i \leqslant T(v) - 1 : v(i) = z_1,\ u(i) = z_2\}| \quad (3)$$

$$N_v(z, l) = |\{0 \leq i \leq l - 1 :\ v(i) = z\}| \quad (4)$$

Here (1) quantifies the number of appearances of z in v, (2) quantifies the number of coincidences between the two sequences u and v, (3) quantifies the dependency between the two sequences u and v and (4) quantifies the number of occurrences of z in a subsequence of v of length l.

The definition of period as well as some related properties can be found in [1, 2]. For the correlation and other approaches it is required first to define the additive character of P as

$$\chi : P \to \mathbb{C}^*$$

$$\chi(x) = e^{2 \cdot \pi \cdot i \cdot tr_{P_0}^{P}(x)/q_1}$$

where $tr_{P_0}^{P}(x)$ is the trace of $x \in P$ over the prime field $P_0 = GF(q_1)$. In [7] it can be seen that

$$\sum_{c \in P} \chi(cx) = \begin{cases} 0 & \text{if } x \in P^* \\ q & \text{if } x = 0 \end{cases}$$

Other measure for randomness used in our paper is entropy [8], also called Shannon's entropy. Let ξ be a discrete random variable and $(p_1, \ldots, p_i, \ldots)$ its probability distribution, then entropy of ξ is defined as

$$H_\xi = -\sum_{i \geq 1} p_i \log_2(p_i)$$

3 Theoretical results

3.1 Period

Proposition 1. *The period of the sequence v generated by the rule R_1 is*

$$T(v) = \begin{cases} q & \text{if } (u(0), \cdots, u(k-1)) = (0, \cdots, 0) \\ q(q^k - 1) & \text{if } (u(0), \cdots, u(k-1)) \neq (0, \cdots, 0) \end{cases}$$

Proof. Assuming $(u(0), \cdots, u(k-1)) = (0, \cdots, 0)$ for all $i \in \mathbb{N}$ we have that $v(i) = \pi_{i \bmod q}[0] + 0 = \pi_{i \bmod q}[0]$. Since the permutations $\pi_0, \ldots, \pi_{q-1}$ form a Latin square the first q elements of v are different and later elements begin to repeat, therefore $T(v) = q$.

Suppose now that $(u(0), \cdots, u(k-1)) \neq (0, \cdots, 0)$. For the sequence v it is expected that $v(i + q(q^k - 1)) = v(i)$, then $T(v) | q(q^k - 1)$. Let us now show that for any $r < q$ the value $r(q^k - 1)$ cannot be the period. Analyzing the elements in positions $0, q^k - 1, \cdots, (q-1)(q^k - 1)$ we can see that

$$\begin{aligned} 0 &\equiv 0 \bmod q \\ q^k - 1 &\equiv q - 1 \bmod q \\ 2(q^k - 1) &\equiv q - 2 \bmod q \\ &\vdots \\ (q-1)(q^k - 1) &\equiv 1 \bmod q \end{aligned}$$

this way the different permutations used by the rule R_1 to generate the elements $v(0), v(q^k - 1), \cdots, v((q-1)(q^k - 1))$ are $\pi_0, \pi_{q-1}, \cdots, \pi_1$. Thus, taking into account that $q^k - 1$ is the period of u and $q(q^k - 1) \equiv 0 \bmod q$ we have that $T(v) = q(q^k - 1)$. ▲

3.2 Distribution

Proposition 2. *The sequence v generated by the rule R_1 satisfies*

$$N_v(z) = \begin{cases} 1 & \text{if } (u(0), \cdots, u(k-1)) = (0, \cdots, 0) \\ q^k - 1 & \text{if } (u(0), \cdots, u(k-1)) \neq (0, \cdots, 0) \end{cases}$$

for all $z \in P$.

Proof. Assuming $(u(0), \cdots, u(k-1)) = (0, \cdots, 0)$ from the proof of the proposition 1 all elements $v(0), \cdots, v(q-1)$ are different and $T(v) = q$, this way $N_v(z) = 1$.

Suppose now that $(u(0), \cdots, u(k-1)) \neq (0, \cdots, 0)$, then we can form a matrix $\mathcal{M}$ with q rows and $q^k - 1$ columns in the following way

$$\mathcal{M} = \begin{pmatrix} v(0) & v(1) & \cdots & v(q^k-2) \\ v(q^k-1) & v(q^k) & \cdots & v(2q^k-3) \\ \cdots & \cdots & \cdots & \cdots \\ v((q-1)(q^k-1)) & v((q-1)(q^k-1)+1) & \cdots & v(q(q^k-1)-1) \end{pmatrix}$$

this way from the proof of the proposition 1 it is easy to see that each possible element appears only once in every column of $\mathcal{M}$, then it follows that the value of $N_v(z)$ coincides with the number of columns. ▲

From this point we can find the probability distribution in a piece of v of length $l \leqslant q^k - 1$ using the measure $N_v(z, l)$ defined in the previous section. let's assume that each element in the subsequence v' of v of length l is independently from each other with probability $p = 1/q$ and let ξ be the random variable that characterizes the event of how many times an element can appear in v', then ξ is a binomially distributed random variable

$$P(\xi = s) = \binom{l}{s} p^s (1-p)^{l-s}$$

with expected value $E(\xi) = lp$. This way we can approximate $N_v(z, l)$, the number of occurrences of z in the subsequence v', to the expected value $E(\xi)$. For instance $N_v(z, l) \approx (q^k - 1)/q$ if $l = q^k - 1$ and $N_v(z, l) \approx 1$ if $l = q$.

3.3 Correlation

Proposition 3. *The sequence v generated by the rule R_1 satisfies*

$$N(u, v) = \begin{cases} 1 & \text{if } (u(0), \cdots, u(k-1)) = (0, \cdots, 0) \\ q^k - 1 & \text{if } (u(0), \cdots, u(k-1)) \neq (0, \cdots, 0) \end{cases}$$

Proof. Assuming $(u(0), \cdots, u(k-1)) = (0, \cdots, 0)$ then $u(i) = 0$ for all $0 \leq i \leq T(v) - 1$. On the other hand, from proposition 1 we have $T(v) = q$ and from proposition 2 we have $N_v(0) = 1$, thus $N(u, v) = 1$.

Suppose now that $(u(0), \cdots, u(k-1)) \neq (0, \cdots, 0)$ and consider the matrix $\mathcal{M}$ defined in the proof of the proposition 2, then it is easy to note that each column determines a permutation in S_q, so as $T(u) = q^k - 1$ each element of u match a single element of v for each column of $\mathcal{M}$. Therefore it is clear that $N(u, v) = q^k - 1$. ▲

Corollary 1. *For any linear recursive sequence u with primitive characteristic polynomial of degree k, not necessarily $F(x)$, the sequence v generated by the rule R_1 satisfied $N(u,v) = q^k - 1$.*

Proposition 4. *If $(u(0), \cdots, u(k-1)) \neq (0, \cdots, 0)$ then for all $\vec{z} = (z_1, z_2)$ where $z_1, z_2 \in P$ we have $N(\vec{z}, u, v) = N_u(z_1)$.*

Proof. For the sequences u and v it is satisfied

$$\begin{aligned}
N(\vec{z},u,v) &= \sum_{i=0}^{T(v)-1} \left(\frac{1}{q}\sum_{t_1\in P}\chi(t_1(u(i)-z_1))\right)\left(\frac{1}{q}\sum_{t_2\in P}\chi(t_2(v(i)-z_2))\right) \\
&= \frac{1}{q^2}\sum_{\vec{t}\in P^2}\chi(-\vec{t}\vec{z})\sum_{i=0}^{T(v)-1}\chi(t_1u(i)+t_2v(i)) \\
&= \frac{T(v)}{q^2} + \frac{1}{q^2}\sum_{\vec{t}\in P^2\setminus\{\vec{0}\}}\chi(-\vec{t}\vec{z})\sum_{i=0}^{T(v)-1}\chi(t_1u(i)+t_2v(i)) \\
&= \frac{T(v)}{q^2} + \frac{1}{q^2}\sum_{t_1\in P^*}\chi(-t_1z_1)\sum_{i=0}^{T(v)-1}\chi(t_1u(i)) + \\
&\qquad + \frac{1}{q^2}\sum_{t_2\in P^*}\chi(-t_2z_2)\sum_{i=0}^{T(v)-1}\chi(t_2v_1(i)) + \\
&\qquad + \frac{1}{q^2}\sum_{\vec{t}\in P^{*2}}\chi(-\vec{t}\vec{z})\sum_{i=0}^{T(v)-1}\chi(t_1u(i)+t_2v(i))
\end{aligned}$$

Let us denote by A, B and C each of these addends and introduce

$$u_1(i) = c_0u(i) + \cdots + c_{k-1}u(i+k-1)$$
$$u_2(i) = (t_2c_k + t_1)u(i) + t_2c_{k+1}u(i+1) + \cdots + t_2c_{2k-1}u(i+k-1)$$

then

$$\begin{aligned}
C &= \frac{1}{q^2}\sum_{\vec{t}\in P^{*2}}\chi(-\vec{t}\vec{z})\sum_{i=0}^{T(v)-1}\chi(t_1u(i)+t_2v(i)) \\
&= \frac{1}{q^2}\sum_{\vec{t}\in P^{*2}}\chi(-\vec{t}\vec{z})\sum_{i=0}^{T(v)-1}\chi(u_2(i)+\pi_{i \bmod q}[u_1(i)]
\end{aligned}$$

from proposition 2 and the properties of the additive character we have

$$C = \frac{(q^k-1)}{q^2}\sum_{\vec{t}\in P^{*2}}\chi(-\vec{t}\vec{z})\sum_{x\in P}\chi(x) = 0$$

Let's see now A

$$\begin{aligned}
A &= \frac{1}{q^2}\sum_{t_1\in P^*}\chi(-t_1z_1)\sum_{i=0}^{T(v)-1}\chi(t_1u(i)) \\
&= \frac{1}{q^2}\sum_{t_1\in P^*}\chi(-t_1z_1)q\sum_{i=0}^{T(u)-1}\chi(c_1u(i)) \\
&= -\frac{T(u)}{q}+\frac{T(u)}{q}+\frac{1}{q}\sum_{t_1\in P^*}\chi(-t_1z_1)\sum_{i=0}^{T(u)-1}\chi(t_1u(i)) \\
&= -\frac{T(u)}{q}+\frac{1}{q}\sum_{t_1\in P}\chi(-t_1z_1)\sum_{i=0}^{T(u)-1}\chi(t_1u(i)) \\
&= -\frac{T(u)}{q}+N_u(z_1)
\end{aligned}$$

Now taking into account that $N_v(z) = T(u)$ for all $z \in P$

$$\begin{aligned}
B &= \frac{T(v)}{q^2}+\frac{1}{q^2}\sum_{t_2}\chi(-t_2z_2)\sum_{i=0}^{T(v)-1}\chi(t_2v(i)) \\
&= \frac{1}{q}\left(\frac{1}{q}\sum_{t_2\in P}\chi(-t_2z_2)\sum_{i=0}^{T(v)-1}\chi(t_2v(i))\right) = \frac{N_v(z_2)}{q} = \frac{T(u)}{q}
\end{aligned}$$

This way $N(\vec{z},u,v) = A+B+C = N_u(z_1)$. ▲

Corollary 2. *If* $(u(0),\cdots,u(k-1)) \neq (0,\cdots,0)$ *then for all* $\vec{z} = (z_1,z_1)$ *where* $z_1 \in P$ *we have* $N(\vec{z},u,v) = q^k - 1$.

Proof. Since $z_1 \in P$ the number of possible $\vec{z} = (z_1,z_1)$ is q, then from proposition 4 we have

$$N(\vec{z},u,v) = (q-1)N_u(z) + N_u(0) = (q-1)q^{k-1} + q^{k-1} - 1 = q^k - 1. \blacktriangle$$

Because proposition 4 establishes some kind of dependency between the two sequences u and v it is natural to ask about the independence hypothesis [10]. To carry out this analysis we will take into account the following table with the values of $N(\vec{z},u,v)$ where each element in P is expressed as power of a primitive element γ.

	$u \setminus v$	0	γ	γ^2	$\cdots$	γ^{q-1}	$\sum$
1	0	$q^{k-1}-1$	$q^{k-1}-1$	$q^{k-1}-1$	$\cdots$	$q^{k-1}-1$	$q(q^{k-1}-1)$
2	γ	q^{k-1}	q^{k-1}	q^{k-1}	$\cdots$	q^{k-1}	q^k
3	γ^2	q^{k-1}	q^{k-1}	q^{k-1}	$\cdots$	q^{k-1}	q^k
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
q	γ^{q-1}	q^{k-1}	q^{k-1}	q^{k-1}	$\cdots$	q^{k-1}	q^k
$\sum$		q^k-1	q^k-1	q^k-1	$\cdots$	q^k-1	$T(v)$

Table 1: values of $N(\vec{z},u,v)$ for the sequence v generated by the rule R_1. The last row contains the sum of all the values in each column and the last column contains the sum of all the values in each row.

For the different data in table 1 let us find $\hat{\chi}^2_n$, where $n = q(q^k-1)$, determined by the expression

$$\hat{\chi}^2_n = n\left(\sum_{i=1}^{q}\sum_{j=1}^{q}\frac{\nu_{ij}^2}{\nu_{i.}\nu_{j.}} - 1\right)$$

where $\nu_{i.}$ values are the q^k-1 elements. Then it can be seen that

$$\begin{aligned}
\sum_{i=1}^{q}\sum_{j=1}^{q}\frac{\nu_{ij}^2}{\nu_{i.}\nu_{j.}} &= \frac{1}{q^k-1}\sum_{i=1}^{q}\sum_{j=1}^{q}\frac{\nu_{ij}^2}{\nu_{i.}} \\
&= \frac{1}{q^k-1}\left(\sum_{j=1}^{q}\frac{\nu_{1j}^2}{\nu_{1.}} + \sum_{i=2}^{q}\sum_{j=1}^{q}\frac{\nu_{ij}^2}{\nu_{i.}}\right) \\
&= \frac{1}{q^k-1}\left(\sum_{j=1}^{q}\frac{(q^{k-1}-1)^2}{q(q^{k-1}-1)} + \sum_{i=2}^{q}\sum_{j=1}^{q}\frac{(q^{k-1})^2}{q^k}\right) \\
&= \frac{1}{q^k-1}\left(q^{k-1}-1+(q-1)qq^{k-2}\right) = \frac{q^k-1}{q^k-1} = 1
\end{aligned}$$

Thus $\hat{\chi}^2_n = q(q^k-1)(1-1) = 0$.

From the previous statement it follows that $\hat{\chi}^2_n < \chi^2_{1-\alpha;(q-1)^2}$ whatever $0 \le \alpha \le 1$, then the null hypothesis H_0 is satisfied, which means that both, u and v are statistically independent, even when v is generated by u. In other words, if we observe the random variable $\xi = (\xi_1, \xi_2)$ with distribution function $F_\xi(x,y)$, since H_0 is always satisfied then $F_\xi(x,y) = F_{\xi_1}(x)F_{\xi_2}(y)$. This result ensures that v is uniformly distributed.

4 Practical results

In the previous section a theoretical analysis was carried out on the distribution in the subsequence $(v(0), \ldots, v(l-1))$ where $l \leqslant q^k - 1$. This section shows practical experiments about the distribution, validating the theoretical results previously shown, and we also use entropy to quantify the randomness of the generated sequences.

We carry out $2^{16} - 1$ practical experiments corresponding to the possible non-zero internal states of the filter generator constructed from the rule R_1 according to the following parameters:

- $P = GF(2^8) \simeq GF(2)[y]/y^8 + y^7 + y^5 + y + 1$
- $F(x) = x^2 + (y+1)x + (y^7 + y^4 + y^2 + y)$
- $\pi_0 = (0, 1, \cdots, 255)$, $\pi_1 = (1, 2, \cdots, 0)$, $\cdots$, $\pi_{255} = (255, 0, \cdots, 254)$

where for each of these experiments the sequence v was completely generated and 256 subsequences of length $l = 2^{16} - 1$ were analyzed.

First we present the entropy analysis based on the criterion that pseudo-random sequences must have entropy close enough to 8 for these parameters. To quantify entropy the RStudio software [9] and the Chao-Shen (CS), Dirichlet (D), Empirical (E), Miller-Madow (MM), Nemenman-Shafee-Bialek (NSB), Plugin (Pl) and Shrink (S) estimators were used. Table 2 shows the mean entropy values for the $2^{16} - 1$ experiments carried out, and we can see in all cases how these values are always above 7.997.

CS	D	E	MM	NSB	Pl	S
7.99763	7.99764	7.99771	7.99733	7.99743	7.99746	7.99741

Table 2: mean entropy values for all sequences generated by the rule R_1.

Now we present the practical results related to distribution for the $2^{16} - 1$ experiments carried out based on the criterion that pseudo-random sequences must have distribution close enough to the uniform distribution. To check this in practice the Pearson's chi-square test [10] was used.

In all experiments the null hypothesis H_0 was accepted and the entropy was greater than 7.99, sufficiently close to the expected entropy for uniformly distributed sequences. We also check the success rate for an element in the subsequence $(v(il), \ldots, v((i+1)l-1))$ where $0 \leq i \leq 255$, obtaining 255.992 for an expected value of 255.996 according to the theoretical analysis.

We conclude that, in practice, all analyzed sequences satisfy the entropy and distribution acceptance criteria for pseudo-randomness.

5 Construction of Latin squares

To carry out the experiments presented in the previous section a basic Latin square was selected. Although the properties of the output sequences of the proposed filter generator are independent of the Latin square chosen, with the exception of those that were discarded in section 2, we present here a new method to construct Latin squares which can be used, for instance, to extend the lifetime of the initial state of the linear feedback shift register.

We stress we are not solving the problem of generating random Latin squares; we only construct a particular class of Latin squares for their use in the generation of sequences by means of the rule R_1.

From now on we will consider the function mod i in $\mathbb{Z}_i = \{1, 2, \cdots, i\}$ where $i \geq 1$ defined as follows

$$(x+y) \bmod i = \begin{cases} x+y & , \ x+y \leq i \\ x+y-i & , \ x+y > i \end{cases}$$

Then using the fact that $x \equiv 1 \bmod 1$ for all $x \in \mathbb{Z}$ we are able to present the following proposition.

Proposition 5. *Let $(x_1, x_2 \cdots, x_{q-1})$ be such that $x_j \in \mathbb{Z}_j$ and let y be a permutation of $S(\mathbb{Z}_q)$, then the different elements*

$$z_{ij} = ((\cdots(x_j + x_{j+1}) \ mod \ (j+1) + \cdots + x_{q-1}) \ mod \ (q-1) + y[i]) \ mod \ q$$

determine a unique $q \times q$ Latin square for all $1 \leq i, j \leq q$.

Proof. To prove the previous proposition it is enough to show that the elements z_{ij} form unique permutations by rows and by columns.

Let's fix the index i of the rows, then we can see that

$$z_{iq} = y[i] \quad \text{and} \quad z_{i(q-1)} = (x_{q-1} + y[i]) \bmod q$$

so as $x_{q-1} \in \mathbb{Z}_{q-1}$ then $z_{i(q-1)} \neq z_{iq}$.

$$\begin{aligned} z_{i(q-2)} &= ((x_{q-2} + x_{q-1}) \bmod (q-1) + y[i]) \bmod q \\ &= (x'_{q-1} + y[i]) \bmod q \end{aligned}$$

where $x'_{q-1} = (x_{q-2} + x_{q-1}) \bmod (q-1)$, so as $x_{q-2} \in \mathbb{Z}_{q-2}$ then $x'_{q-1} \neq x_{q-1}$ and $z_{i(q-2)}, z_{i(q-1)}, z_{iq}$ are different elements. Following the previous idea we can see that $z_{i1}, z_{i2}, \cdots, z_{iq}$ are all different elements, therefore the i-th row $(z_{i1}, z_{i2}, \cdots, z_{iq})$ determines a permutation whose uniqueness is evident.

Let us now show that the columns also form permutations. Let's see that the column q-th is given by the elements

$$z_{iq} = y[i]$$

that form a permutation. The column $(q-1)$-th is given by the elements

$$z_{i(q-1)} = (x_{q-1} + y[i]) \bmod q$$

that form a permutation. The column $(q-2)$-th is given by the elements

$$z_{i(q-1)} = ((x_{q-2} + x_{q-1}) \bmod (q-1) + y[i]) \bmod q$$

that form a permutation. Following this idea, for all fixed j the elements z_{ij} form a permutation. ▲

6 Conclusion

In this paper a method was presented to construct uniformly distributed sequences on any finite field using the properties of the linear feedback shift registers with primitive characteristic polynomial and Latin squares. The theoretical results show that the elements generated by this method appear with the same frequency in the sequence exactly $q^k - 1$ times each of these, where k is the degree of the primitive polynomial.

Other theoretical analysis were performed for the distribution of elements in subsequences of length $l \leq q^k - 1$ and the correlation between the output sequence v and the recursive sequence u, showing that both are statistically independent even when v is determined by means of u.

On the other hand practical experiments show that the output sequences have a mean entropy value greater than 7.99, very close to the expected entropy for uniformly distributed sequences. Other practical analyzes show that the generated sequences satisfy the Golomb's randomness postulates, however, further correlation and distribution analysis must still be done and a bound for the range of these sequences must be found.

Acknowledgments

The authors thank Reynier Antonio de la Cruz Jimenez for his valuable contributions to this paper and his recommendations to submit it to the 10th Workshop on Current Trends in Cryptology (CTCrypt 2021).

References

[1] Fomichev V. M., "Methods of Discrete Mathematics in Cryptology", *Dialog-MEPhI Publ*, 2010, In Russian.

[2] Lidl R. and Niederreiter H., *Finite fields*, Cambridge university press, 1997.

[3] Fan X., Mandal K. and Gong G., "Wg-8: A lightweight stream cipher for resource-constrained smart devices", *International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*, 2013, 617-632.

[4] Kamlovskii, O.V., "Distribution of r-tuples in one class of uniformly distributed sequences over residue rings", *Problems of Information Transmission*, **50**:1 (2014), 98-115.

[5] Kamlovskii, O.V., "Equidistributed sequences over finite fields produced by one class of linear recurring sequences over residue rings", *Problems of Information Transmission*, **50**:1 (2014), 171–185.

[6] Sachkov V., "Introduction to combinatorial methods of discrete mathematics", 1982, In Russian.

[7] Kamlovsky O. V., "The number of occurrences of elements in the output sequences of filter generators", *Applied discrete mathematics*, **21**:3 (2013), 11-25, In Russian.

[8] Klein A., *Stream ciphers*, Springer, 2013.

[9] Kronthaler F. and Zöllner S., *Data Analysis with RStudio*, Springer Spektrum, 2021.

[10] Ivchenko G. I. and Medvedev Y. I. Introduction to Mathematical statistics, 2010, In Russian.

On Derivatives of Boolean Bent Functions

Alexander Shaporenko

Sobolev Institute of Mathematics, Novosibirsk, Russia
Novosibirsk State University, Novosibirsk, Russia
JetBrains Research, Novosibirsk, Russia
shaporenko.alexandr@gmail.com

Abstract

In this paper, the property of affine functions to be derivatives of bent functions is studied. The importance of Boolean bent functions in symmetric cryptography stems from linear cryptanalysis of stream ciphers. In that context bent functions are the ones which are the worst approximated by affine functions. There are also connections between bent functions and distinct objects of coding theory such as Reed-Muller and Kerdock codes. Recall, that a bent function is a Boolean function f in n variables (n is even) such that for any nonzero vector y its derivative $D_y f(x) = f(x) \oplus f(x \oplus y)$ is balanced, i.e., it takes values 0 and 1 equally often. Whether every balanced function is a derivative of a some bent function or not is an open problem. In this paper, special case of this problem was studied. It was proven that every nonconstant affine function in n (even) variables is a derivative of $(2^{n-1}-1)|\mathcal{B}_{n-2}|^2$ bent functions, where $\mathcal{B}_n$ is a set of all bent functions in n variables. Iterative lower bound for the number of bent functions is presented.

Keywords: Boolean functions, bent functions, derivatives of a bent function, lower bound for the number of bent functions.

1 Introduction

A Boolean function in even number of variables is called *bent* if it has maximal nonlinearity [1]. Nonlinearity is an important property in cryptography. Ciphers using functions with high nonlinearity as components are more resistant to linear cryptanalysis [2] because bent functions are badly approximated by affine functions. Bent functions were used in design of the block cipher CAST as coordinate functions of S-blocks [3]. The nonlinear feedback polynomial of the NFSR (nonlinear feedback shift register) of the stream cipher Grain is constructed as the sum of a linear function and a bent function [4]. There are also connections between bent functions and distinct objects of coding theory such as *Reed-Muller and Kerdock codes* [5]. In coding theory, there is a well-known task of determining the covering radius for the *Reed-Muller code* $RM(l, n)$. This task is related (if the code has order 1) to the

task of finding the most nonlinear Boolean functions [6, 7]. Special sets of quadratic bent functions allow one to construct Kerdock codes [8] that are optimal and have large code distances that grow with the code lengths [9, 10]. This very optimality of Kerdock codes is caused by extremal properties of bent functions.

Another definition of a bent function is the following. It is a Boolean function f in n variables (n is even) such that for any nonzero vector y its derivative $D_y f(x) = f(x) \oplus f(x \oplus y)$ is balanced, i.e., it takes values 0 and 1 equally often [5]. In [11] it was shown that every balanced function is a derivative of a some bent function for $n \leqslant 6$ (n even). Whether it is true for every even n is an open problem. We will study this problem for the case of affine functions.

2 Necessary definitions and statements

Let $\mathbb{Z}_2 = \{0, 1\}$. Denote by $\mathbb{Z}_2^n$ the n-dimensional vector space over $\mathbb{Z}_2$. Let us denote by $\oplus$ the addition modulo 2. We will also use the following inner product:

$$\langle x, y \rangle = x_1 y_1 \oplus \ldots \oplus x_n y_n.$$

A function $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$ is called *a Boolean function* in n variables. A Boolean function f is called *affine* if it can be represented as $l_{a,b}(x) = \langle a, x \rangle \oplus b$, where $a \in \mathbb{Z}_2^n$ and $b \in \mathbb{Z}_2$. A Boolean function is called *balanced* if it takes values 0 and 1 equally often.

Let us recall a well known fact.

Lemma 1. *An affine function $l_{a,b}(x) = \langle a, x \rangle \oplus b$, where $a \in \mathbb{Z}_2^n$ (nonzero) and $b \in \mathbb{Z}_2$, is balanced.*

The Hamming weight $wt(f)$ of a Boolean function f is the number of vectors $x \in Z_2^n$ such that $f(x) = 1$. For nonconstant affine functions it is equal to 2^{n-1}. We denote by $dist(f, g)$ *the Hamming distance* between two Boolean functions f and g; it is the number of positions in which their vectors of values differ:

$$dist(f, g) = |\{x \in \mathbb{Z}_2^n : f(x) \neq g(x)\}|.$$

Every Boolean function f in n variables can be associated with its *support*:

$$\mathrm{supp}(f) = \{x \in \mathbb{Z}_2^n : f(x) = 1\}.$$

A Boolean function $D_y f(x) = f(x) \oplus f(x \oplus y)$ is called a *derivative* of a Boolean function f in n variables in *the direction* y, where $y \in \mathbb{Z}_2^n$.

Lemma 2. *A Boolean function f in n variables is a derivative of a some Boolean function g in n variables in nonzero direction y if and only if $f(x) \oplus f(x \oplus y) = 0$ for all $x \in \mathbb{Z}_2^n$.*

Proof. ($\Rightarrow$) One can see that $D_y g(x) = g(x) \oplus g(x \oplus y) = D_y g(x \oplus y)$ for all $x \in \mathbb{Z}_2^n$. Therefore, $f(x) = f(x \oplus y)$ for all $x \in \mathbb{Z}_2^n$.

($\Leftarrow$) Let i be the first nonzero coordinate of y. Define a Boolean function g in the following way $g(x) = x_i f(x)$ for all $x \in \mathbb{Z}_2^n$. Then

$$D_y g(x) = x_i f(x) \oplus (x_i \oplus 1) f(x \oplus y) = f(x) \text{ for all } x \in \mathbb{Z}_2^n.$$

Therefore, f is a derivative of g in the direction y. □

The *nonlinearity* of a Boolean function f in n variables is the Hamming distance N_f from this function to the set of all affine functions, i.e., $N_f = \min\limits_{a \in \mathbb{Z}_2^n, b \in \mathbb{Z}_2} dist(f, l_{a,b})$.

A *bent function* is a Boolean function in an even number of variables that has the maximal nonlinearity, i.e., $N_f = 2^{n-1} - 2^{n/2-1}$. Denote by $\mathcal{B}_n$ a set of all bent functions in n variables.

The Walsh-Hadamard transform of a Boolean function f in n variables is the integer-valued function on $\mathbb{Z}_2^n$ defined as

$$W_f(y) = \sum_{x \in \mathbb{Z}_2^n} (-1)^{f(x) \oplus \langle x, y \rangle}, \text{ for every } y \in \mathbb{Z}_2^n.$$

For a bent function f, the *dual function* $\tilde{f}$ in n variables is defined by the equality $W_f(y) = 2^{n/2}(-1)^{\tilde{f}(y)}$. This definition is correct since $W_f(y) = \pm 2^{n/2}$ for any vector y if f is a bent function [5].

Lemma 3. *(see, for instance, [5]) A Boolean function f in n variables is bent if and only if for any nonzero vector y its derivative $D_y f(x) = f(x) \oplus f(x \oplus y)$ is balanced or equivalently it holds*

$$\sum_{x \in \mathbb{Z}_2^n} (-1)^{f(x) \oplus f(x \oplus y)} = 0, \text{ for any nonzero } y.$$

Lemma 4. *Let $l_{a,b}(x) = \langle a, x \rangle \oplus b$, where $a \in \mathbb{Z}_2^n$, a is nonzero, and $b \in \mathbb{Z}_2$. There are $2^{n-1} - 1$ nonzero directions for which $l_{a,b}$ is a derivative of a some Boolean function. Namely, these directions are exactly those nonzero vectors y such that $\langle a, y \rangle = 0$.*

Proof. If $\langle a, y \rangle = 0$ then

$$l_{a,b}(x) \oplus l_{a,b}(x \oplus y) = \langle a, x \rangle \oplus b \oplus \langle a, x \oplus y \rangle \oplus b = \langle a, y \rangle = 0.$$

Therefore, it follows from Lemma 2 that the function $l_{a,b}$ is a derivative of a some Boolean function in the direction y. It is known that there exist 2^{n-1} different nonzero vectors y such that $\langle a, y\rangle = 0$ if a is nonzero. Since $\langle a, 0\rangle = 0$ as well, the statement is proved. □

Lemma 5. *Let l be a nonconstant affine function that is a derivative of bent functions g and g' in distinct nonzero directions y and y', respectively. Then $g \neq g'$.*

Proof. Suppose that $g = g'$ is a bent function such that $D_y g(x) = D_{y'} g(x) = l(x)$ for $y \neq y'$. Then for every $x \in \mathbb{Z}_2^n$ it holds

$$\begin{aligned} &D_y g(x) \oplus D_{y'} g(x) = \\ &= g(x) \oplus g(x \oplus y) \oplus g(x) \oplus g(x \oplus y') = \\ &= g(x \oplus y) \oplus g(x \oplus y') = \\ &= g(x \oplus y) \oplus g(x \oplus y \oplus (y' \oplus y)) = \\ &= D_{y' \oplus y} g(x \oplus y) = 0, \end{aligned}$$

which contradicts Lemma 3. □

3 Affine functions as derivatives of bent functions

In what follows we suppose that n is even.

Theorem 1. *Any nonconstant affine function $l_{a,b}$ in n variables is a derivative of $(2^{n-1} - 1)|\mathcal{B}_{n-2}|^2$ bent functions in $n \geq 4$ variables.*

Proof. Let $l_{a,b}(x) = \langle a, x\rangle \oplus b$ be an affine function in $n \geq 4$ variables, where $a \in \mathbb{Z}_2^n$ is nonzero and $b \in \mathbb{Z}_2$. Suppose that $l_{a,b}$ is a derivative of some Boolean function in the direction y'.

Let i be the number of the first nonzero coordinate of y' and j be the number such that $j \neq i$ and x_j is an essential variable for $l_{a,b}$. Let us show that such j always exists. Suppose the opposite. Then $l_{a,b}(x) = x_i \oplus b$ and $D_{y'} l_{a,b}(x) = l_{a,b}(x) \oplus l_{a,b}(x \oplus y') = 1$, for every $x \in \mathbb{Z}_2^n$, which by Lemma 2 contradicts the fact that $l_{a,b}$ is a derivative of g in the direction y'.

Without loss of generality, let $i = 1$ and $j = 2$. It follows from Lemma 2 that $l_{a,b}(x) = l_{a,b}(x \oplus y')$ and hence

$$x \in \operatorname{supp}(l_{a,b}) \iff x \oplus y' \in \operatorname{supp}(l_{a,b}). \tag{1}$$

Note that for any Boolean function g in n variables that has $\ell_{a,b}$ as its derivative in the direction y' it holds that

$$g(x) \oplus g(x \oplus y') = l_{a,b}(x). \tag{2}$$

It follows from (1) and (2) that any Boolean function g, such that $D_{y'}g(x) = \ell_{a,b}(x)$, has the following representation

$$\begin{aligned}
&g(0, x_2, \bar{x}) = f_0(\bar{x}), && (0, x_2, \bar{x}) \in \mathrm{supp}(l_{a,b}),\\
&g(1, x_2 \oplus y'_2, \bar{x} \oplus \bar{y'}) = f_0(\bar{x}) \oplus 1, && (1, x_2 \oplus y'_2, \bar{x} \oplus \bar{y'}) \in \mathrm{supp}(l_{a,b}),\\
&g(0, x_2, \bar{x}) = f_1(\bar{x}), && (0, x_2, \bar{x}) \notin \mathrm{supp}(l_{a,b}),\\
&g(1, x_2 \oplus y'_2, \bar{x} \oplus \bar{y'}) = f_1(\bar{x}), && (1, x_2 \oplus y'_2, \bar{x} \oplus \bar{y'}) \notin \mathrm{supp}(l_{a,b}),
\end{aligned}$$

where

$$\bar{z} = (z_3, \ldots, z_n), \text{for } z_k \in \mathbb{Z}_2$$

and f_0, f_1 are some Boolean functions in $n-2$ variables. Therefore, by considering different Boolean functions in $n-2$ variables as f_0 and f_1, we can get all Boolean functions in n variables that have $\ell_{a,b}$ as its derivative in the direction y'.

Let f_0 and f_1 be bent functions and g be defined as above. Denote by $M = \{x \in \mathbb{Z}_2^n : x_1 = 0\}$. Thus, $x \in M \iff x \oplus y' \in \mathbb{Z}_2^n \backslash M$.

Let us show that g is bent by checking that $D_y g(x)$ is balanced for every nonzero $y \neq y'$.

Suppose that $b = 0$. Then $l_{a,b}(x \oplus y) = l_{a,b}(x) \oplus l_{a,b}(y)$ and from (1) and (2) we have

$$\begin{aligned}
&\sum_{x \in \mathbb{Z}_2^n} (-1)^{g(x) \oplus g(x \oplus y)} =\\
&= \sum_{\substack{x \in M\\ x \in \mathrm{supp}(l_{a,b})}} (-1)^{g(x) \oplus g(x \oplus y)} + (-1)^{g(x \oplus y') \oplus g(x \oplus y \oplus y')} +\\
&+ \sum_{\substack{x \in M\\ x \notin \mathrm{supp}(l_{a,b})}} (-1)^{g(x) \oplus g(x \oplus y)} + (-1)^{g(x \oplus y') \oplus g(x \oplus y \oplus y')} =
\end{aligned}$$

$$\begin{aligned}
&= \sum_{\substack{x \in M\\ x \in \mathrm{supp}(l_{a,b})}} (-1)^{g(x) \oplus g(x \oplus y)} + (-1)^{g(x) \oplus g(x \oplus y) \oplus l_{a,b}(x) \oplus l_{a,b}(x \oplus y)} +\\
&+ \sum_{\substack{x \in M\\ x \notin \mathrm{supp}(l_{a,b})}} (-1)^{g(x) \oplus g(x \oplus y)} + (-1)^{g(x) \oplus g(x \oplus y) \oplus l_{a,b}(x) \oplus l_{a,b}(x \oplus y)} =
\end{aligned}$$

$$= \sum_{\substack{x\in M\\ x\in \mathrm{supp}(l_{a,b})}} (-1)^{g(x)\oplus g(x\oplus y)} + (-1)^{g(x)\oplus g(x\oplus y)\oplus l_{a,b}(y)} +$$
$$+ \sum_{\substack{x\in M\\ x\notin \mathrm{supp}(l_{a,b})}} (-1)^{g(x)\oplus g(x\oplus y)} + (-1)^{g(x)\oplus g(x\oplus y)\oplus l_{a,b}(y)}.$$

There are two cases:

Case 1. If $l_{a,b}(y) = 1$. Then

$$\sum_{x\in\mathbb{Z}_2^n} (-1)^{g(x)\oplus g(x\oplus y)} =$$
$$= \sum_{\substack{x\in M\\ x\in \mathrm{supp}(l_{a,b})}} (-1)^{g(x)\oplus g(x\oplus y)} + (-1)^{g(x)\oplus g(x\oplus y)\oplus 1} +$$
$$+ \sum_{\substack{x\in M\\ x\notin \mathrm{supp}(l_{a,b})}} (-1)^{g(x)\oplus g(x\oplus y)} + (-1)^{g(x)\oplus g(x\oplus y)\oplus 1} = 0.$$

Case 2. If $l_{a,b}(y) = 0$. Then $l_{a,b}(x \oplus y) = 1 \iff l_{a,b}(x) = 1$.
Suppose that $y_1 = 0$. Then

$$g(0, x_2 \oplus y_2, \bar{x} \oplus \bar{y}) = f_0(\bar{x} \oplus \bar{y}), \qquad (0, x_2, \bar{x}) \in \mathrm{supp}(l_{a,b}),$$
$$g(0, x_2 \oplus y_2, \bar{x} \oplus \bar{y}) = f_1(\bar{x} \oplus \bar{y}), \qquad (0, x_2, \bar{x}) \notin \mathrm{supp}(l_{a,b}),$$

and

$$\sum_{x\in\mathbb{Z}_2^n} (-1)^{g(x)\oplus g(x\oplus y)} =$$
$$= \sum_{\substack{x\in M\\ x\in \mathrm{supp}(l_{a,b})}} (-1)^{g(x)\oplus g(x\oplus y)} + (-1)^{g(x)\oplus g(x\oplus y)} +$$
$$+ \sum_{\substack{x\in M\\ x\notin \mathrm{supp}(l_{a,b})}} (-1)^{g(x)\oplus g(x\oplus y)} + (-1)^{g(x)\oplus g(x\oplus y)} =$$
$$= 2\Big(\sum_{\substack{x\in M\\ x\in \mathrm{supp}(l_{a,b})}} (-1)^{g(x)\oplus g(x\oplus y)} + \sum_{\substack{x\in M\\ x\notin \mathrm{supp}(l_{a,b})}} (-1)^{g(x)\oplus g(x\oplus y)} \Big) =$$
$$= 2\Big(\sum_{\substack{x\in M\\ x\in \mathrm{supp}(l_{a,b})}} (-1)^{f_0(\bar{x})\oplus f_0(\bar{x}\oplus\bar{y})} + \sum_{\substack{x\in M\\ x\notin \mathrm{supp}(l_{a,b})}} (-1)^{f_1(\bar{x})\oplus f_1(\bar{x}\oplus\bar{y})} \Big). \qquad (3)$$

Let us show that $\bar{y} \neq 0$. Since y is nonzero, then $\bar{y} = 0$ only if $y_2 = 1$. But in that case $l_{a,b}(y) = 1$ since x_2 is an essential variable for $l_{a,b}$ and $b = 0$.

Note that if $(a_1, a_2, \bar{x}) \in \mathrm{supp}(l_{a,b})$ then $(a_1, a_2 \oplus 1, \bar{x}) \notin \mathrm{supp}(l_{a,b})$, where $a_1, a_2 \in \mathbb{Z}_2$, since x_2 is essential for $l_{a,b}$. Therefore,

$$\{\bar{x} : (0, x_2, \bar{x}) \in \mathrm{supp}(l_{a,b})\} = \{\bar{x} : (0, x_2, \bar{x}) \notin \mathrm{supp}(l_{a,b})\} = \mathbb{Z}_2^{n-2}, \qquad (4)$$

and since f_0 and f_1 are bent it follows from Lemma 3 that

$$\begin{aligned}
&\sum_{x \in \mathbb{Z}_2^n} (-1)^{g(x) \oplus g(x \oplus y)} = \\
&= 2\Big(\sum_{\substack{x \in M \\ x \in \mathrm{supp}(l_{a,b})}} (-1)^{f_0(\bar{x}) \oplus f_0(\bar{x} \oplus \bar{y})} + \sum_{\substack{x \in M \\ x \notin \mathrm{supp}(l_{a,b})}} (-1)^{f_1(\bar{x}) \oplus f_1(\bar{x} \oplus \bar{y})} \Big) = \\
&= 2\Big(\sum_{\bar{x} \in \mathbb{Z}_2^{n-2}} (-1)^{f_0(\bar{x}) \oplus f_0(\bar{x} \oplus \bar{y})} + \sum_{\bar{x} \in \mathbb{Z}_2^{n-2}} (-1)^{f_1(\bar{x}) \oplus f_1(\bar{x} \oplus \bar{y})} \Big) = 0.
\end{aligned}$$

Suppose that $y_1 = 1$. Then

$$\begin{aligned}
g(1, x_2 \oplus y_2, \bar{x} \oplus \bar{y}) &= f_0(\bar{x} \oplus \bar{y} \oplus \bar{y'}) \oplus 1, \qquad && (0, x_2, \bar{x}) \in \mathrm{supp}(l_{a,b}), \\
g(1, x_2 \oplus y_2, \bar{x} \oplus \bar{y}) &= f_1(\bar{x} \oplus \bar{y} \oplus \bar{y'}), && (0, x_2, \bar{x}) \notin \mathrm{supp}(l_{a,b}),
\end{aligned}$$

and

$$\begin{aligned}
&\sum_{x \in \mathbb{Z}_2^n} (-1)^{g(x) \oplus g(x \oplus y)} = \\
&= \sum_{\substack{x \in M \\ x \in \mathrm{supp}(l_{a,b})}} (-1)^{g(x) \oplus g(x \oplus y)} + (-1)^{g(x) \oplus g(x \oplus y)} + \\
&+ \sum_{\substack{x \in M \\ x \notin \mathrm{supp}(l_{a,b})}} (-1)^{g(x) \oplus g(x \oplus y)} + (-1)^{g(x) \oplus g(x \oplus y)} = \\
&= 2\Big(\sum_{\substack{x \in M \\ x \in \mathrm{supp}(l_{a,b})}} (-1)^{g(x) \oplus g(x \oplus y)} + \sum_{\substack{x \in M \\ x \notin \mathrm{supp}(l_{a,b})}} (-1)^{g(x) \oplus g(x \oplus y)} \Big) = \\
&= 2\Big(\sum_{\substack{x \in M \\ x \in \mathrm{supp}(l_{a,b})}} (-1)^{f_0(\bar{x}) \oplus f_0(\bar{x} \oplus \bar{y} \oplus \bar{y'}) \oplus 1} + \sum_{\substack{x \in M \\ x \notin \mathrm{supp}(l_{a,b})}} (-1)^{f_1(\bar{x}) \oplus f_1(\bar{x} \oplus \bar{y} \oplus \bar{y'})} \Big). \qquad (5)
\end{aligned}$$

Therefore, if $\bar{y'} \neq \bar{y}$, then from (4) and since f_0 and f_1 are bent it follows

from Lemma 3 that

$$\sum_{x\in\mathbb{Z}_2^n}(-1)^{g(x)\oplus g(x\oplus y)} =$$
$$= 2\Big(\sum_{\substack{x\in M\\ x\in\mathrm{supp}(l_{a,b})}}(-1)^{f_0(\bar{x})\oplus f_0(\bar{x}\oplus\bar{y}\oplus\bar{y'})\oplus 1} + \sum_{\substack{x\in M\\ x\notin\mathrm{supp}(l_{a,b})}}(-1)^{f_1(\bar{x})\oplus f_1(\bar{x}\oplus\bar{y}\oplus\bar{y'})}\Big) =$$
$$= 2\Big(\sum_{\bar{x}\in\mathbb{Z}_2^{n-2}}(-1)^{f_0(\bar{x})\oplus f_0(\bar{x}\oplus\bar{y}\oplus\bar{y'})\oplus 1} + \sum_{\bar{x}\in\mathbb{Z}_2^{n-2}}(-1)^{f_1(\bar{x})\oplus f_1(\bar{x}\oplus\bar{y}\oplus\bar{y'})}\Big) = 0.$$

If $\bar{y'} = \bar{y}$, then from (4) we have that

$$\sum_{x\in\mathbb{Z}_2^n}(-1)^{g(x)\oplus g(x\oplus y)} =$$
$$= 2\Big(\sum_{\bar{x}\in\mathbb{Z}_2^{n-2}}(-1)^{f_0(\bar{x})\oplus f_0(\bar{x})\oplus 1} + \sum_{\bar{x}\in\mathbb{Z}_2^{n-2}}(-1)^{f_1(\bar{x})\oplus f_1(\bar{x})}\Big) =$$
$$= 2\big(-2^{n-2}+2^{n-2}\big) = 0.$$

It follows from Lemma 3 that g is bent.

If $b = 1$ then

$$\sum_{x\in\mathbb{Z}_2^n}(-1)^{g(x)\oplus g(x\oplus y)} =$$
$$= \sum_{\substack{x\in M\\ x\in\mathrm{supp}(l_{a,b})}}(-1)^{g(x)\oplus g(x\oplus y)} + (-1)^{g(x)\oplus g(x\oplus y)\oplus l_{a,b}(y)\oplus 1} +$$
$$+ \sum_{\substack{x\in M\\ x\notin\mathrm{supp}(l_{a,b})}}(-1)^{g(x)\oplus g(x\oplus y)} + (-1)^{g(x)\oplus g(x\oplus y)\oplus l_{a,b}(y)\oplus 1},$$

and to show that g is bent it is sufficient to switch Cases 1 and 2. It is worth to elaborate on the case when $b = 1$, $\ell_{a,b}(y) = 1$ and $y_1 = 0$. If $\bar{y} = 0$ then from (3) we get

$$\sum_{x\in\mathbb{Z}_2^n}(-1)^{g(x)\oplus g(x\oplus y)} =$$
$$= 2\Big(\sum_{\substack{x\in M\\ x\in\mathrm{supp}(l_{a,b})}}(-1)^{f_0(\bar{x})\oplus f_0(\bar{x}\oplus\bar{y})} + \sum_{\substack{x\in M\\ x\notin\mathrm{supp}(l_{a,b})}}(-1)^{f_1(\bar{x})\oplus f_1(\bar{x}\oplus\bar{y})}\Big) =$$
$$= 2\Big(\sum_{\substack{x\in M\\ x\in\mathrm{supp}(l_{a,b})}}(-1)^{f_0(\bar{x})\oplus f_0(\bar{x})} + \sum_{\substack{x\in M\\ x\notin\mathrm{supp}(l_{a,b})}}(-1)^{f_1(\bar{x})\oplus f_1(\bar{x})}\Big) = 2^n$$

and hence $D_y g(x)$ is not balanced. Let us show that this case is not possible. Since y is nonzero, then $\bar{y} = 0$ only if $y_2 = 1$. But in that case $l_{a,b}(y) = 1 \oplus b = 0$ since x_2 is an essential variable for $l_{a,b}$ and $b = 1$.

Now let us show that if g is bent then f_0 and f_1 are bent. Suppose the opposite. Let f_0 is not bent. Then it follows from Lemma 3 that there is exist a nonzero vector $\bar{y}$ such that $D_{\bar{y}} f_0(\bar{x})$ is not balanced.

Note that there is always a nonzero vector $y = (0, y_2, \bar{y}) \notin \mathrm{supp}(l_{a,b})$, since x_2 is essential for $l_{a,b}$ and hence either $(0, a, \bar{y}) \notin \mathrm{supp}(l_{a,b})$ or $(0, a \oplus 1, \bar{y}) \notin \mathrm{supp}(l_{a,b})$, where $y_2, a \in \mathbb{Z}_2$.

Suppose that $b = 0$. Then from (3), (4) and since g is bent

$$\begin{aligned}
&\sum_{x \in \mathbb{Z}_2^n} (-1)^{g(x) \oplus g(x \oplus y)} = \\
&= 2\Big(\sum_{\substack{x \in M \\ x \in \mathrm{supp}(l_{a,b})}} (-1)^{f_0(\bar{x}) \oplus f_0(\bar{x} \oplus \bar{y})} + \sum_{\substack{x \in M \\ x \notin \mathrm{supp}(l_{a,b})}} (-1)^{f_1(\bar{x}) \oplus f_1(\bar{x} \oplus \bar{y})} \Big) = \\
&= 2\Big(\sum_{\bar{x} \in \mathbb{Z}_2^{n-2}} (-1)^{f_0(\bar{x}) \oplus f_0(\bar{x} \oplus \bar{y})} + \sum_{\bar{x} \in \mathbb{Z}_2^{n-2}} (-1)^{f_1(\bar{x}) \oplus f_1(\bar{x} \oplus \bar{y})} \Big) = 0,
\end{aligned}$$

and hence

$$\sum_{\bar{x} \in \mathbb{Z}_2^{n-2}} (-1)^{f_0(\bar{x}) \oplus f_0(\bar{x} \oplus \bar{y})} = - \sum_{\bar{x} \in \mathbb{Z}_2^{n-2}} (-1)^{f_1(\bar{x}) \oplus f_1(\bar{x} \oplus \bar{y})}. \tag{6}$$

From (1) we know that $(1, y_2 \oplus y_2', \bar{y} \oplus \bar{y'}) \notin \mathrm{supp}(l_{a,b})$. Therefore, from (5), (4) and since g is bent

$$\begin{aligned}
&\sum_{x \in \mathbb{Z}_2^n} (-1)^{g(x) \oplus g(x \oplus y \oplus y')} = \\
&= 2\Big(\sum_{\substack{x \in M \\ x \in \mathrm{supp}(l_{a,b})}} (-1)^{f_0(\bar{x}) \oplus f_0(\bar{x} \oplus \bar{y}) \oplus 1} + \sum_{\substack{x \in M \\ x \notin \mathrm{supp}(l_{a,b})}} (-1)^{f_1(\bar{x}) \oplus f_1(\bar{x} \oplus \bar{y})} \Big) = \\
&= 2\Big(\sum_{\bar{x} \in \mathbb{Z}_2^{n-2}} (-1)^{f_0(\bar{x}) \oplus f_0(\bar{x} \oplus \bar{y}) \oplus 1} + \sum_{\bar{x} \in \mathbb{Z}_2^{n-2}} (-1)^{f_1(\bar{x}) \oplus f_1(\bar{x} \oplus \bar{y})} \Big) = 0,
\end{aligned}$$

and hence

$$\sum_{\bar{x} \in \mathbb{Z}_2^{n-2}} (-1)^{f_0(\bar{x}) \oplus f_0(\bar{x} \oplus \bar{y})} = \sum_{\bar{x} \in \mathbb{Z}_2^{n-2}} (-1)^{f_1(\bar{x}) \oplus f_1(\bar{x} \oplus \bar{y})}. \tag{7}$$

Consequently, (6) and (7) contradict each other, since $D_{\bar{y} f_0(\bar{x})}$ is not balanced.

If $b = 1$ it is sufficient to consider a nonzero vector $(0, y_2 \oplus 1, \bar{y}) \in \mathrm{supp}(l_{a,b})$.

Note that for different $\{f_0, f_1\}$ and $\{f'_0, f'_1\}$ we get different g. Since f_0 and f_1 are arbitrary bent functions in $n-2$ variables there are $|\mathcal{B}_{n-2}|^2$ bent functions g for which $l_{a,b}$ is a derivative in the direction y'.

It follows from Lemma 5 that for different directions y' we get different bent funcions that have $l_{a,b}$ as its derivative. Therefore, it follows from Lemma 4 that there are $(2^{n-1}-1)|\mathcal{B}_{n-2}|^2$ bent functions that have $l_{a,b}$ as the derivative. □

4 Iterative lower bound

Theorem 1 gives us an iterative lower bound.

Theorem 2. *For even $n \geq 2$ it holds $|\mathcal{B}_{n+2}| \geqslant (2^{n+2}-2)|\mathcal{B}_n|^2$.*

Proof. Let l a be nonconstant affine function in $n+2$ variables. It follows from Theorem 1 that there are $(2^{n+1}-1)|\mathcal{B}_n|^2$ bent functions in $n+2$ variables that have l as its derivative. Therefore, $|\mathcal{B}_{n+2}| \geqslant (2^{n+1}-1)|\mathcal{B}_n|^2$.

Let us show that it is not possible for some bent function to have both l and $l \oplus 1$ as its derivatives. Suppose that $g(x)$ is a bent and $D_y = l(x)$ and $D_{y'} = l(x) \oplus 1$ for $y \neq y'$. Then for every $x \in \mathbb{Z}_2^n$

$$\begin{aligned}
&D_y g(x) \oplus D_{y'} g(x) = \\
&= g(x) \oplus g(x \oplus y) \oplus g(x) \oplus g(x \oplus y') = \\
&= g(x \oplus y) \oplus g(x \oplus y') = \\
&= g(x \oplus y) \oplus g(x \oplus y \oplus (y' \oplus y)) = \\
&= D_{y' \oplus y} g(x \oplus y) = l(x) \oplus l(x) \oplus 1 = 1,
\end{aligned}$$

which contradicts Lemma 3. Thus, we can multiply our bound by 2. □

Let us compare this iterative lower bound with other known. We have this iterative lower bound from [12]

$$|\mathcal{B}_{n+2}| \geqslant 6|\mathcal{B}_n|^2 - 8|\mathcal{B}_n|$$

but it is not better than the following one.

Proposition 1. *(Climent et al, [13]) For even $n \geq 2$ it holds*

$$|\mathcal{B}_{n+2}| \geqslant 6|\mathcal{B}_n|^2 + 2^{n+2}(2^n - 3)|\mathcal{B}_n|.$$

The Iterative lower bound from Proposition 1 is worse than one from Theorem 2 for every even $n \leq 8$. See Table 1.

Variables	4	6	8	10
Bent	896	5 425 430 528	$2^9\times$ 193 887 869 660 028 067 003 488 010 240 $\approx 2^{106.29}$	?
Proposition 1	512	5 562 368	176 611 863 208 449 277 952 $\approx 2^{68}$	$\approx 2^{215}$
Theorem 2	896	49 774 592	7 476 565 289 195 207 131 136 $\approx 2^{72.6}$	$\approx 2^{222.5}$
Proposition 3	512	322 961 408	$\approx 2^{87.35}$	$\approx 2^{262.16}$

Table 1: Number of bent functions constructed with different methods

Proposition 2. *(Canteaut et al, [14], Tokareva [15]) Let functions f_0, f_1, and f_2 be bent functions in n variables. Then function g defined as*

$$g(0,0,x) = f_0(x), \qquad g(0,1,x) = f_1(x),$$
$$g(1,0,x) = f_2(x), \qquad g(1,1,x) = f_3(x),$$

is a bent function in $n+2$ variables if and only if f_3 is a bent function in n variables and $\tilde{f}_0 \oplus \tilde{f}_1 \oplus \tilde{f}_2 \oplus \tilde{f}_3 = 1$.

Bent functions that can be obtained by Proposition 2 are called *bent iterative* functions. Let $\mathcal{BI}_{n+2}$ denote the class of all such functions in $n+2$ variables.

The following iterative lower bound is based on Proposition 2. It was proven by the author [15] in 2011. For now it is the best iterative lower bound for the number of bent functions.

Proposition 3. *(Tokareva, [15]) For even $n \geq 2$ it holds*

$$|\mathcal{B}_{n+2}| \geqslant |\mathcal{BI}_{n+2}| \geq |\mathcal{B}_n|^4/|X_n|,$$

where X_n is the set of all Boolean functions in n variables that can be represented as the sum of two distinct bent functions.

The Iterative lower bound from Theorem 2 is not better than one from Proposition 3 when $n \geq 6$. See Table 1.

5 Conclusion and open problems

In [11] it was shown that every balanced function f in n variables is a derivative of a some bent function for $n \leqslant 6$ (n even). Whether it is true for nonaffine balanced functions for every even n is an open problem.

Iterative lower bound from Theorem 2 theoretically can be improved if we consider more then two affine functions l and $l \oplus 1$. Unfortunately, it is hard to keep track of bent functions that were already counted because it is possible that $D_y g(x) = l(x)$ and $D_{y'} g(x) = h(x)$, where $h \neq l$, $h \neq l \oplus 1$ and $y \neq y'$.

We also can consider bent functions that do not have affine derivatives. Such functions of degree 3 were studied for example in [14]. Although, the number of such functions was not presented.

6 Acknowledgement

The work was carried out within the framework of the state contract of the Sobolev Institute of Mathematics (project no. 0314-2019-0017) and supported by Laboratory of Cryptography JetBrains Research.

References

[1] Rothaus O. S., "On 'bent' functions", *J. Combinat. Theory A*, **20**:3 (1976), 300–305.

[2] Matsui M., "Linear Cryptanalysis Method for DES cipher", *Advances in Cryptology*, EUROCRYPT 1993., Springer, Berlin, Heidelberg, 386–397.

[3] Adams C., "Constructing symmetric ciphers using the CAST design procedure", *Proc. Design, Codes, and Cryptography*, **12**:3 (1997), 283–316.

[4] Hell M., Johansson T., Maximov A., and Meier W., "A stream cipher proposal: Grain-128", *IEEE International Symposium on Information Theory*, 2006, 1614–1618.

[5] Tokareva N., *Bent functions: results and applications to cryptography*, Acad. Press. Elsevier, 2015.

[6] Kavut S., Maitra S., Yucel MD., "Search for Boolean functions with excellent profiles in the rotation symmetric class", *IEEE Trans Inform Theory*, **53**:5 (2007), 1743–1751.

[7] Maitra S., Sarkar P., "Maximum nonlinearity of symmetric Boolean functions on odd number of variables", *IEEE Trans Inform Theory*, **48**:9 (2002), 2626-2630.

[8] Kerdock AM., "A class of low-rate non-linear binary codes", *Inform Control*, **20**:2 (1972), 182-187.

[9] Delsarte P., "An algebraic approach to the association schemes of coding theory", *Ph.d. thesis*, 1973.

[10] Sidelnikov V. M., "On extremal polynomials used in code size estimation", *Probl Inf Transm*, **16**:3 (1980), 174-186.

[11] Tokareva N. N., "On the set of derivatives of a boolean bent function", *Applied Discrete Mathematics. Supplement*, **9** (2016), 327–350.

[12] Climent JJ, Garcia FJ, Requena V., "On the construction of bent functions of n + 2 variables from bent functions of n variables", *Adv Math Commun*, **2**:4 (2008), 421–431.

[13] Climent JJ, Garcia FJ, Requena V, "A construction of bent functions of n + 2 variables from a bent function of n variables and its cyclic shifts", *Algebra*, 2014.

[14] Canteaut A, Charpin P., "Decomposing bent functions", *IEEE Trans Inform Theory*, **49**:8 (2003), 2004–2019.

[15] Tokareva N. N., "On the number of bent functions from iterative constructions: lower bounds and hypotheses", *Adv Math Commun*, **5**:4 (2011), 609–621.

The Duality Mapping and Unitary Operators Acting on the Set of All Generalized Boolean Functions

Aleksandr Kutsenko[1,2] and Anastasiya Gorodilova[1]

[1]Sobolev Institute of Mathematics, Novosibirsk, Russia
[2]Novosibirsk State University, Novosibirsk, Russia
alexandrkutsenko@bk.ru, gorodilova@math.nsc.ru

Abstract

Functions of the form $\mathbb{F}_2^n \to \mathbb{Z}_q$, where $q \geqslant 2$ is a positive integer, are known as generalized Boolean functions. Bent functions within this generalization are called generalized bent (gbent). A gbent function is said to be regular if it is possible to define its dual gbent function. A duality mapping is the mapping that transforms every regular gbent function to its dual gbent. A regular gbent function is said to be self-dual if it coincides with its dual. In this paper we define the action of linear operator $\mathbb{C}^{2^n} \to \mathbb{C}^{2^n}$ on the set of all generalized Boolean functions in n variables via their sign functions. The characterization of unitary operators that transform the set of all generalized Boolean functions in n variables into itself is provided. Further all such operators that commute with the duality mapping and preserve self-duality of a gbent function are found. Based on this result the known classification of quaternary self-dual bent functions is clarified. We also study the properties if sign functions of self-dual gbent functions.

Keywords: Duality mapping, Generalized bent function, Self-dual bent

1 Introduction

The study of Boolean functions having strong cryptographic properties is the domain of current interest, see monographies [2, 4] for detail. Boolean bent functions were introduced by O. Rothaus [26] in 1976. Due to maximal nonlinearity they have a number of applications in cryptography and coding theory. They were used as building blocks of stream (Grain, 2004) and block (CAST, 1997) ciphers and, for instance, in 2000 T. Wada [34] established a connection between bent functions and binary constant-amplitude codewords. But despite the long history of research in this area there are still many open problems. Among them the exact number of bent functions as well as their complete classification that seem elusive to be solved for now. One can find more details on bent functions in books [33, 22].

Bent functions were initially generalized by P. V. Kumar et al. in 1985 by considering functions of the form $\mathbb{Z}_q^n \to \mathbb{Z}_q$ with corresponding definition of bentness, see [12]. Bent functions from a finite Abelian group into a finite Abelian group were studied in [29] by V. I. Solodovnikov and by O.A. Logachev, A.A. Sal'nikov, V.V. Yashchenko in [18]. Having applications of functions from $\mathbb{F}_2^n$ to $\mathbb{Z}_4$ in code-division multiple access (CDMA) systems, K.-U Schmidt in [27] (initially appeared in preprint from 2006) generalized the notion of bentness for functions of the form $\mathbb{F}_2^n \to \mathbb{Z}_q$, where $q \geqslant 2$ is a positive integer and studied these functions for the case $q = 4$. The considered functions are named generalized bent (gbent) functions. Note that this generalization deals with the mappings of the form $\mathbb{F}_2^n \to \mathbb{Z}_q$ called generalized Boolean functions, that are also studied from the view of obtaining linear codes with special properties, see [24]. These functions also have applications to the analysis of quantum circuits [6]. In recent years generalized bent functions obtained much attention, in particular, for the case $q = 2^k$. In papers [20, 30] different constructions and properties of generalized bent functions were obtained. The connection between concepts of strong regularity of (edge-weighted) Cayley graph associated to a generalized Boolean function and gbent functions was pointed in [25]. The complete characterization of generalized bent functions from different perspectives was recently presented in works [31, 7, 23]. A comprehensive survey on existing generalizations of bent functions can be found in [32].

The **duality mapping** is a mapping that transforms every (regular generalized) bent function to its dual (generalized) bent. For the Boolean case for every bent function its dual bent function is uniquely defined. It is important to note that the duality mapping is the unique known isometric mapping of the set of bent functions into itself that cannot be extended to a isometry of the whole set of all Boolean functions that preserves bentness. The action of the duality mapping on bent functions within generalizations is increasingly nontrivial since it is typically defined only for the part of bent functions from corresponding generalization which are called *regular*, while more accurate work with them also demands for intermediate notation like *weak regularity* that also appears in this scope.

Self-dual bent functions that are fixed points of the duality mapping form a remarkable class of bent functions since they have the direct relation to their dual bent functions. The definition of self-duality initially was in paper [18] by O.A. Logachev, A.A. Sal'nikov and V.V. Yashchenko. In more details these functions were explored by C. Carlet et al. in 2010 in work [1], where a number of constructions and properties were given and the classification for

small number of variables was provided. Next steps for the classification of qubic self-dual bent functions in 8 variables were made in [5], while quadratic self-dual bent functions were completely characterized in [8]. Constructions and properties of self-dual Boolean bent functions were studied in [10, 13, 21]. The overview of the known metrical properties of self-dual bent functions can be found in [17]. The extension of the concept of self-duality for different generalizations of bent functions was made in several papers. The classification of quadratic self-dual bent functions of the form $\mathbb{F}_p^n \to \mathbb{F}_p$, p odd prime, was made by X.-D. Hou in [9]. In paper [3] the self-duality for bent functions within the same generalization type was studied by A. Çeşmelioğlu et al. In 2018 in paper [28] L. Sok. et al. studied quaternary self-dual bent functions of the form $\mathbb{F}_2^n \to \mathbb{Z}_4$ from the viewpoints of existence, construction, and symmetry. The relation between sign functions of quaternary self-dual bent function in n variables and two self-dual bent functions in n variables was found. Based on this it was proved that there are no quaternary self-dual bent functions in odd number of variables.

In this paper we define the action of linear operator $\mathbb{C}^{2^n} \to \mathbb{C}^{2^n}$ on the generalized Boolean functions in n variables via their sign functions. We study the interconnection between unitary operators that transform the set of all generalized Boolean functions in n variables into itself and the duality mapping. The paper os organised as follows. In Section 2 necessary definitons and notation are given. In Section 3 properties of sign functions of self-dual gbent functions are considered. The main results are in Section 4, namely, Section 4.1, where unitary operators under consideration are characterized, in Sections 4.3 and 4.4 the operators that commute or anti-commute with duality mapping, hence preserve self-duality or replacing self-dual and anti-self-dual gbent functions, respectively, are described. The question whether the duality mapping can be described by the considered set of operators is partially studied in Section 5 The conclusion is in Section 6.

2 Notation

Let $\mathbb{F}_2^n$ be a set of binary vectors of length n. For $x, y \in \mathbb{F}_2^n$ denote $\langle x, y\rangle = \bigoplus_{i=1}^{n} x_i y_i$, where the sign $\oplus$ denotes a sum modulo 2. Denote, following [11], the orthogonal group of index n over the field $\mathbb{F}_2$ as

$$\mathcal{O}_n = \left\{ L \in \mathrm{GL}\,(n, \mathbb{F}_2) : LL^T = I_n \right\},$$

where L^{T} denotes the transpose of L and I_n is an identical matrix of order n over the field $\mathbb{F}_2$.

A *generalized Boolean function* f in n variables is any map from $\mathbb{F}_2^n$ to $\mathbb{Z}_q$, the integers modulo q. The set of generalized Boolean functions in n variables is denoted by $\mathcal{GF}_n^q$, for the case $q = 2$ we use $\mathcal{F}_n$. Let $\omega = e^{2\pi i/q}$. A *sign* function of $f \in \mathcal{GF}_n^q$ is a complex valued function $F = \omega^f$, we will also refer to it as to a complex vector $\left(\omega^{f_0}, \omega^{f_1}, \ldots, \omega^{f_{2^n-1}}\right)$ of length 2^n, where $(f_0, f_1, \ldots, f_{2^n-1})$ is a vector of values of the function f.

The *Hamming weight* $\mathrm{wt}_H(x)$ of the vector $x \in \mathbb{F}_2^n$ is the number of nonzero coordinates of x. The *Hamming distance* $\mathrm{dist}_H(f, g)$ between generalized Boolean functions f, g in n variables is the cardinality of the set $\{x \in \mathbb{F}_2^n | f(x) \neq g(x)\}$. The Lee weight of the element $x \in \mathbb{Z}_q$ is $\mathrm{wt}_L(x) = \min\{x, q - x\}$. The Lee distance $\mathrm{dist}_L(f, g)$ between $f, g \in \mathcal{GF}_n^q$ is

$$\mathrm{dist}_L(f, g) = \sum_{x \in \mathbb{F}_2^n} \mathrm{wt}_L\left(\delta(x)\right),$$

where $\delta \in \mathcal{GF}_n^q$ and $\delta(x) = f(x) + (q-1)g(x)$ for any $x \in \mathbb{F}_2^n$. For Boolean case $q = 2$ the Hamming distance coincides with the Lee distance.

The *(generalized) Walsh–Hadamard transform* of $f \in \mathcal{GF}_n^q$ is the complex valued function:

$$H_f(y) = \sum_{x \in \mathbb{F}_2^n} \omega^{f(x)} (-1)^{\langle x, y \rangle}.$$

A generalized Boolean function f in n variables is said to be *generalized bent* (gbent) if

$$|H_f(y)| = 2^{n/2},$$

for all $y \in \mathbb{F}_2^n$ [27]. If there exists such $\widetilde{f} \in \mathcal{GF}_n^q$ that $H_f(y) = \omega^{\widetilde{f}(y)} 2^{n/2}$ for any $y \in \mathbb{F}_2^n$, the gbent function f is said to be *regular* and $\widetilde{f}$ is called its *dual*. Note that $\widetilde{f}$ is generalized bent as well. A regular gbent function f is said to be *self-dual* if $f = \widetilde{f}$, and *anti-self-dual* if $f = \widetilde{f} + q/2$. Consequently, it is the case only for even q. So throughout this paper we assume that q is a positive even integer. Corresponding sets of gbent functions are denoted by $\mathrm{SB}_q^+(n)$ and $\mathrm{SB}_q^-(n)$.

The *duality mapping* is a mapping that transforms every regular gbent function to its dual one. Thus, it is essentially defined only on regular gbent functions.

3 Eigenvectors of the duality mapping

In this section properties of sign functions of (anti-)self-dual gbent functions will be studied and the connection with the duality mapping will be explicitely pointed.

Let I_n be the identity matrix of size n and $H_n = H_1^{\otimes n}$ be the n-fold tensor product of the matrix H_1 with itself, where

$$H_1 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

This matrix is known as Sylvester Hadamard matrix. It is known the Hadamard property of this matrix

$$H_n H_n^T = 2^n I_{2^n},$$

where H_n^T is transpose of H_n (it holds $H_n^T = H_n$ by symmetricity of H_n). Denote $\mathcal{H}_n = 2^{-n/2} H_n$.

By using Sylvester Hadamard matrix it is possible to define the duality mapping as follows

$$\omega^f \longrightarrow \mathcal{H}_n \omega^f = \omega^{\tilde{f}},$$

where f is a regular gbent function in n variables. Thus, sign functions if self-dual gbent functions are eigenvectors of the aforementioned linear operator that correspond to the eigenvalue 1. At the same time sign functions if anti-self-dual gbent functions are eigenvectors of the aforementioned linear operator that correspond to the eigenvalue (-1). In terms of subspaces these facts imply that sign functions belong to the spaces $\mathrm{Ker}\big(\mathcal{H}_n - I_{2^n}\big) = \mathrm{Ker}\big(H_n - 2^{n/2} I_{2^n}\big)$ and $\mathrm{Ker}\big(\mathcal{H}_n + I_{2^n}\big) = \mathrm{Ker}\big(H_n + 2^{n/2} I_{2^n}\big)$ correspondingly.

Recall an orthogonal decomposition of $\mathbb{R}^{2^n}$ in eigenspaces of H_n from [1] (Lemma 5.2):

$$\mathbb{R}^{2^n} = \mathrm{Ker}\left(H_n + 2^{n/2} I_{2^n}\right) \oplus \mathrm{Ker}\left(H_n - 2^{n/2} I_{2^n}\right),$$

where the symbol $\oplus$ denotes a direct sum of subspaces. Consider the same decomposition

$$\mathbb{C}^{2^n} = \mathrm{Ker}\left(H_n + 2^{n/2} I_{2^n}\right) \oplus \mathrm{Ker}\left(H_n - 2^{n/2} I_{2^n}\right),$$

for a complex space $\mathbb{C}^{2^n}$. It is known that

$$\dim\big(\mathrm{Ker}\left(\mathcal{H}_n + I_{2^n}\right)\big) = \dim\big(\mathrm{Ker}\big(\mathcal{H}_n - I_{2^n}\big)\big) = 2^{n-1},$$

where $\dim(V)$ is the dimension of the subspace $V \subseteq \mathbb{C}^{2^n}$. Moreover, since $\mathcal{H}_n$ is symmetric (Hermitian), the subspaces $\mathrm{Ker}\big(\mathcal{H}_n + I_{2^n}\big)$ and $\mathrm{Ker}\big(\mathcal{H}_n - I_{2^n}\big)$ are mutually orthogonal.

In [15] it was proved that provided $n \geqslant 4$, the linear span of sign functions of self-dual as well as anti-self-dual Boolean bent functions Boolean bent functions in n variables has dimension 2^{n-1}. The same result can be also given for gbent functions:

Proposition 1. *Let $n \geqslant 4$ be an evem number, then the linear span of sign functions of (anti-)self-dual gbent functions in n variables has dimension 2^{n-1}.*

Proof. It is enough to mention that since q is even it holds $(-1) = \omega^{q/2} \in \big\{\omega, \omega^2, \ldots, \omega^{q-1}\big\}$, therefore the set of sign fuctions of (anti-)self-dual Boolean bent functions in n variables is a subset of the set of sign functions of (anti-)self-dual gbent functions in n variables. □

It is worth to note that the example of the basis of the subspace $\mathrm{Ker}\big(\mathcal{H}_n - I_{2^n}\big)$ can be constructed by using the functions obtained via iterative constructions from [1] and [15].

When $n = 2$ there are two self-dual Boolean bent functions, namely x_1x_2 and $x_1x_2 \oplus 1$, which have sign functions $(1, 1, 1, -1)$ and $(-1, -1, -1, 1)$ respectively. These sign functions are linearly dependent vectors in $\mathbb{R}^4$. The set $\mathrm{SB}^-(2)$ consists of functions $x_1x_2 \oplus x_1 \oplus x_2$ and $x_1x_2 \oplus x_1 \oplus x_2 \oplus 1$ with sign functions $(1, -1, -1, -1)$ and $(-1, 1, 1, 1)$ respectively. These sign functions are linearly dependent vectors in $\mathbb{R}^4$ as well. Generalization comprises solution of the system

$$\frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} \omega^{d_1} \\ \omega^{d_2} \\ \omega^{d_3} \\ \omega^{d_4} \end{pmatrix} = \begin{pmatrix} \omega^{d_1} \\ \omega^{d_2} \\ \omega^{d_3} \\ \omega^{d_4} \end{pmatrix},$$

where variables are numbers $d_1, d_2, d_3, d_4 \in \mathbb{Z}_q$ in fact. It is clear that the only solution pattern is

$$\big(\omega^d, \omega^d, \omega^d, \omega^{d+q/2}\big) = \omega^d \cdot (1, 1, 1, -1) \in \mathbb{C}^4,$$

where $d \in \mathbb{Z}_q$. It means that any two sign functions of self-dual gbent functions from $\mathrm{SB}_q^+(2)$ are linearly dependent over $\mathbb{C}$ and $\big|\mathrm{SB}_q^+(2)\big| = q$.

The next result is a generalization of the similar one from [15].

Theorem 1. *Let $n \geqslant 4$ be an even number and $f \in \mathrm{SB}_q^+(n)$. For sign function $\omega^f = \left(F^{00}, F^{01}, F^{10}, F^{11}\right)$, where $F^{00}, F^{01}, F^{10}, F^{11} \in \left\{1, \omega, \omega^2, \ldots, \omega^{q-1}\right\}^{2^{n-2}}$, it holds*

$$\left\langle F^{00}, F^{01}\right\rangle + \left\langle F^{10}, F^{11}\right\rangle = 0,$$
$$\left\langle F^{00}, F^{10}\right\rangle + \left\langle F^{01}, F^{11}\right\rangle = 0.$$

Proof. Let $f \in \mathrm{SB}_q^+(n)$, then by Proposition 1 there exist vectors

$$\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_{2^{n-3}}) \in \mathbb{C}^{2^{n-3}},$$
$$\beta = (\beta_1, \beta_2, \ldots, \beta_{2^{n-3}}) \in \mathbb{C}^{2^{n-3}},$$
$$\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_{2^{n-2}}) \in \mathbb{C}^{2^{n-2}},$$

such that

$$\omega^f = \sum_{i=1}^{2^{n-3}} \alpha_i \mathbf{F}_i^n + \sum_{j=1}^{2^{n-3}} \beta_j \mathbf{G}_j^n + \sum_{k=1}^{2^{n-2}} \gamma_k \left(\mathbf{FG}\right)_k^n,$$

where the sets $S_{\mathbf{F}} = \{\mathbf{F}_i^n\}_{i=1}^{2^{n-3}}$, $S_{\mathbf{G}} = \left\{\mathbf{G}_j^n\right\}_{j=1}^{2^{n-3}}$ and $S_{\mathbf{FG}} = \{(\mathbf{FG})_k^n\}_{k=1}^{2^{n-2}}$ are the sets of sign functions of iteratively constructed self-dual Boolean bent functions that form a basis of the eigenspace of the Sylvester Hadamard matrix (see [15]). Namely, the sets $S_{\mathbf{F}}$, $S_{\mathbf{G}}$, $S_{\mathbf{FG}}$ consist of sign functions

$$\mathbf{F}_i^n = (F_i, F_i, F_i, -F_i),$$
$$\mathbf{G}_j^n = (G_j, -G_j, -G_j, -G_j),$$
$$(\mathbf{FG})_k^n = (A_k, -B_k, B_k, A_k),$$

where sign functions $F_i, A_k \in \mathrm{Ker}\left(\mathcal{H}_{n-2} - I_{2^{n-2}}\right)$ and $G_j, B_k \in \mathrm{Ker}\left(\mathcal{H}_{n-2} + I_{2^{n-2}}\right)$, $i, j = 1, 2, \ldots, 2^{n-3}$, $k = 1, 2, \ldots, 2^{n-2}$, are sign functions that form the bases of the eigenspaces of the matrix H_{n-2}. Define the vectors

$$\mathbf{F} = \sum_{i=1}^{2^{n-3}} \alpha_i F_i, \qquad \mathbf{G} = \sum_{j=1}^{2^{n-3}} \beta_j G_j, \qquad \mathbf{A} = \sum_{k=1}^{2^{n-2}} \gamma_k A_k, \qquad \mathbf{B} = \sum_{k=1}^{2^{n-2}} \gamma_k B_k.$$

Under this notation the sign function ω^f has the form

$$\omega^f = \begin{pmatrix} F^{00} \\ F^{01} \\ F^{10} \\ F^{11} \end{pmatrix} = \begin{pmatrix} \mathbf{F} + \mathbf{G} + \mathbf{A} \\ \mathbf{F} - \mathbf{G} - \mathbf{B} \\ \mathbf{F} - \mathbf{G} + \mathbf{B} \\ -\mathbf{F} - \mathbf{G} + \mathbf{A} \end{pmatrix} \in \left\{1, \omega, \omega^2, \ldots, \omega^{q-1}\right\}^{2^n}.$$

For any $j = 1, 2, \ldots, 2^{n-2}$ denote

$$\begin{aligned}
\big(\mathbf{F}+\mathbf{G}\big)_j + \mathbf{A}_j &= \omega^{t_j},\\
\big(\mathbf{F}-\mathbf{G}\big)_j - \mathbf{B}_j &= \omega^{r_j},\\
\big(\mathbf{F}-\mathbf{G}\big)_j + \mathbf{B}_j &= \omega^{l_j},\\
-\big(\mathbf{F}+\mathbf{G}\big)_j + \mathbf{A}_j &= \omega^{k_j},
\end{aligned}$$

where $t_j, r_j, l_j, k_j \in \mathbb{Z}_q$. Then

$$\begin{aligned}
\mathbf{A}_j &= \frac{1}{2}\big(\omega^{t_j} + \omega^{k_j}\big),\\
\mathbf{B}_j &= \frac{1}{2}\big(\omega^{l_j} - \omega^{r_j}\big),\\
\big(\mathbf{F}+\mathbf{G}\big)_j &= \frac{1}{2}\big(\omega^{t_j} - \omega^{k_j}\big),\\
\big(\mathbf{F}-\mathbf{G}\big)_j &= \frac{1}{2}\big(\omega^{r_j} + \omega^{l_j}\big).
\end{aligned}$$

Note that

$$\langle \mathbf{G}, \mathbf{A}\rangle = \langle \mathbf{F}, \mathbf{B}\rangle = 0.$$

By using this we obtain the expression for the first inner product

$$\begin{aligned}
\big\langle F^{00}, F^{01}\big\rangle + \big\langle F^{10}, F^{11}\big\rangle &= \langle \mathbf{F}+\mathbf{G}+\mathbf{A}, \mathbf{F}-\mathbf{G}-\mathbf{B}\rangle\\
&+ \langle \mathbf{F}-\mathbf{G}+\mathbf{B}, -\mathbf{F}-\mathbf{G}+\mathbf{A}\rangle\\
&= \langle \mathbf{F}, \mathbf{F}\rangle - \langle \mathbf{F}, \mathbf{G}\rangle - \langle \mathbf{F}, \mathbf{B}\rangle\\
&+ \langle \mathbf{G}, \mathbf{F}\rangle - \langle \mathbf{G}, \mathbf{G}\rangle - \langle \mathbf{G}, \mathbf{B}\rangle\\
&+ \langle \mathbf{A}, \mathbf{F}\rangle - \langle \mathbf{A}, \mathbf{G}\rangle - \langle \mathbf{A}, \mathbf{B}\rangle\\
&- \langle \mathbf{F}, \mathbf{F}\rangle - \langle \mathbf{F}, \mathbf{G}\rangle + \langle \mathbf{F}, \mathbf{A}\rangle\\
&+ \langle \mathbf{G}, \mathbf{F}\rangle + \langle \mathbf{G}, \mathbf{G}\rangle - \langle \mathbf{G}, \mathbf{A}\rangle\\
&- \langle \mathbf{B}, \mathbf{F}\rangle - \langle \mathbf{B}, \mathbf{G}\rangle + \langle \mathbf{B}, \mathbf{A}\rangle\\
&= \langle \mathbf{A}, \mathbf{F}\rangle + \langle \mathbf{F}, \mathbf{A}\rangle - \langle \mathbf{G}, \mathbf{B}\rangle - \langle \mathbf{B}, \mathbf{G}\rangle\\
&= \langle \mathbf{A}, \mathbf{F}+\mathbf{G}\rangle + \overline{\langle \mathbf{A}, \mathbf{F}+\mathbf{G}\rangle} + \langle \mathbf{B}, \mathbf{F}-\mathbf{G}\rangle + \overline{\langle \mathbf{B}, \mathbf{F}-\mathbf{G}\rangle}
\end{aligned} \tag{1}$$

while the second one has the form

$$\begin{aligned}\left\langle F^{00}, F^{10}\right\rangle + \left\langle F^{01}, F^{11}\right\rangle &= \langle \mathbf{F}+\mathbf{G}+\mathbf{A}, \mathbf{F}-\mathbf{G}+\mathbf{B}\rangle \\ &+ \langle \mathbf{F}-\mathbf{G}-\mathbf{B}, -\mathbf{F}-\mathbf{G}+\mathbf{A}\rangle \\ &= \langle \mathbf{F},\mathbf{F}\rangle - \langle \mathbf{F},\mathbf{G}\rangle + \langle \mathbf{F},\mathbf{B}\rangle \\ &+ \langle \mathbf{G},\mathbf{F}\rangle - \langle \mathbf{G},\mathbf{G}\rangle + \langle \mathbf{G},\mathbf{B}\rangle \\ &+ \langle \mathbf{A},\mathbf{F}\rangle - \langle \mathbf{A},\mathbf{G}\rangle + \langle \mathbf{A},\mathbf{B}\rangle \\ &- \langle \mathbf{F},\mathbf{F}\rangle - \langle \mathbf{F},\mathbf{G}\rangle + \langle \mathbf{F},\mathbf{A}\rangle \\ &+ \langle \mathbf{G},\mathbf{F}\rangle + \langle \mathbf{G},\mathbf{G}\rangle - \langle \mathbf{G},\mathbf{A}\rangle \\ &+ \langle \mathbf{B},\mathbf{F}\rangle + \langle \mathbf{B},\mathbf{G}\rangle - \langle \mathbf{B},\mathbf{A}\rangle \\ &= \langle \mathbf{A},\mathbf{F}\rangle + \langle \mathbf{F},\mathbf{A}\rangle + \langle \mathbf{G},\mathbf{B}\rangle + \langle \mathbf{B},\mathbf{G}\rangle \\ &= \langle \mathbf{A},\mathbf{F}+\mathbf{G}\rangle + \overline{\langle \mathbf{A},\mathbf{F}+\mathbf{G}\rangle} - \langle \mathbf{B},\mathbf{F}-\mathbf{G}\rangle - \overline{\langle \mathbf{B},\mathbf{F}-\mathbf{G}\rangle} \end{aligned} \tag{2}$$

Consider inner in details the following inner products

$$\begin{aligned}\langle \mathbf{A},\mathbf{F}+\mathbf{G}\rangle &= \sum_{j=1}^{2^n} \mathbf{A}_j \overline{\left(\mathbf{F}+\mathbf{G}\right)}_j = \frac{1}{4}\sum_{j=1}^{2^n}\left(\omega^{t_j}+\omega^{k_j}\right)\left(\overline{\omega^{t_j}}-\overline{\omega^{k_j}}\right) \\ &= \frac{1}{4}\sum_{j=1}^{2^n}\left(1-1+\omega^{k_j}\overline{\omega^{t_j}}-\omega^{t_j}\overline{\omega^{k_j}}\right) = \frac{1}{2}\mathrm{Im}\left(\sum_{j=1}^{2^n}\omega^{k_j}\overline{\omega^{t_j}}\right) i, \\ \overline{\langle \mathbf{A},\mathbf{F}+\mathbf{G}\rangle} &= -\frac{1}{2}\mathrm{Im}\left(\sum_{j=1}^{2^n}\omega^{k_j}\overline{\omega^{t_j}}\right) i,\end{aligned}$$

$$\begin{aligned}\langle \mathbf{B},\mathbf{F}-\mathbf{G}\rangle &= \sum_{j=1}^{2^n} \mathbf{B}_j \overline{\left(\mathbf{F}-\mathbf{G}\right)}_j = \frac{1}{4}\sum_{j=1}^{2^n}\left(\omega^{l_j}-\omega^{r_j}\right)\left(\overline{\omega^{l_j}}+\overline{\omega^{r_j}}\right) \\ &= \frac{1}{4}\sum_{j=1}^{2^n}\left(1-1+\omega^{l_j}\overline{\omega^{r_j}}-\omega^{r_j}\overline{\omega^{l_j}}\right) = \frac{1}{2}\mathrm{Im}\left(\sum_{j=1}^{2^n}\omega^{l_j}\overline{\omega^{r_j}}\right) i, \\ \overline{\langle \mathbf{B},\mathbf{F}-\mathbf{G}\rangle} &= -\frac{1}{2}\mathrm{Im}\left(\sum_{j=1}^{2^n}\omega^{l_j}\overline{\omega^{r_j}}\right) i,\end{aligned}$$

therefore, both (1) and (2) are zero numbers. □

4 Unitary operators and eigenvectors of the duality mapping

In this section we define an action of linear operator $\mathbb{C}^{2^n} \to \mathbb{C}^{2^n}$ on a (generalized) Boolean function in n variables and characterize all unitary operators which transform the set of all (generalized) Boolean functions in n variables into itself and preserve self-duality, thus generalizing in some way the results from [16] on isometric mappings which preserve self-duality of a Boolean bent function and those, which define bijections between the sets of self-dual and anti-self-dual Boolean bent functions.

4.1 Linear operators and generalized Boolean functions

Throught this section we will use standard basis of the space $\mathbb{C}^{2^n}$, which consists of the vectors $\{e_i\}_{i=1}^{2^n} \subset \mathbb{C}^{2^n}$, where e_i has 1 on th i-th position, the rest are zeros.

Let $\varphi : \mathbb{C}^{2^n} \to \mathbb{C}^{2^n}$ be linear operator with matrix A in standard basis of the space $\mathbb{C}^{2^n}$. We shall say that φ *transforms* the generalized Boolean function $f \in \mathcal{GF}_n^q$ with sign function F to the generalized Boolean function $f' \in \mathcal{GF}_n^q$ if the sign function F' of f' is equal to AF, that is $F' = AF = \varphi(F)$. This also comprises the definition of the duality mapping via the Sylvester Hadamard matrix (see Section 3).

Recall that a linear operator φ is said to be *unitary* if $\varphi\varphi^\dagger = \varphi^\dagger\varphi = id$, where $\varphi^\dagger$ is a Hermitian adjoint operator of φ. The matrix of φ is called unitary in this case. Denote by $\mathcal{U}_n^q$ the set of unitary operators $\mathbb{C}^{2^n} \to \mathbb{C}^{2^n}$ which transform the set of generalized Boolean functions in n variables $\mathcal{GF}_n^q$ into itself.

The next result characterizes the set $\mathcal{U}_n^q$. The matrix is called *monomial* or *generalized permutation* if it has exactly one nonzero entry in every row (column).

Theorem 2. *Operators from $\mathcal{U}_n^q$ are characterized by monomial matrices with nonzero elements from the set $\left\{1, \omega, \omega^2, \ldots, \omega^{q-1}\right\}$ and only them.*

Proof. It is obvious that operators with monomial matrices of such form transform the set of q-ary generalized Boolean functions in n variables into itself. Moreover every such matrix is unitary.

Now assume $\varphi \in \mathcal{U}_n^q$ and let $U = (u_{ij})_{i,j=1}^{2^n}$ be its matrix in the canonical basis. Denote by $v_0 \in \mathbb{C}^{2^n}$ a vector with all ones and by $v_i \in \mathbb{C}^{2^n}, i = 1, 2, \ldots, 2^n$, a vector which has 1 on the i-th position, the rest are (-1). Let

$v_{ij} \in \mathbb{C}^{2^n}, i, j = 1, 2, \ldots, 2^n, (i \neq j)$, be a vector which has 1 on the i-th and j-th positions, the rest are (-1)

Fix some $i, j, k \in \{1, 2, \ldots, 2^n\}, (i < j)$. Denote $(Uv_0)_k = \omega^{d_0}, (Uv_i)_k = \omega^{d_i}$, $(Uv_j)_k = \omega^{d_j}$ and $(Uv_{ij})_k = \omega^{d_{ij}}$ for some $d_0, d_i, d_j, d_{ij} \in \mathbb{Z}_q$. Their addition yields

$$\begin{aligned}
(Uv_0)_k + (Uv_i)_k &= 2u_{ki} = \omega^{d_0} + \omega^{d_i}, \\
(Uv_0)_k + (Uv_j)_k &= 2u_{kj} = \omega^{d_0} + \omega^{d_j}, \\
(Uv_0)_k + (Uv_{ij})_k &= 2\left(u_{ki} + u_{kj}\right) = \omega^{d_0} + \omega^{d_{ij}}.
\end{aligned}$$

After grouping of these items we see that

$$u_{ki} = \frac{\omega^{d_0} + \omega^{d_i}}{2}, \quad u_{kj} = \frac{\omega^{d_0} + \omega^{d_j}}{2}, \quad u_{ki} + u_{kj} = \frac{\omega^{d_0} + \omega^{d_{ij}}}{2},$$

that is

$$\omega^{d_0} + \omega^{d_i} + \omega^{d_0} + \omega^{d_j} = \omega^{d_0} + \omega^{d_{ij}},$$

or, equivalently,

$$\omega^{d_0} + \omega^{d_i} + \omega^{d_j} = \omega^{d_{ij}}.$$

Its is clear that it is the case only if $\omega^{d_{ij}}$ coincides with one of three numbers $\omega^{d_0}, \omega^{d_i}, \omega^{d_j}$ and the rest two are the same numbers with opposite signs, that is always possible since q is even.

Basically there are two variants:

Case 1: If $\omega^{d_{ij}} = \omega^{d_0}$ and $\omega^{d_i} + \omega^{d_j} = 0$, then the k-th row of U is

$$U_k = \left(u_{k1}, \ldots, u_{k,i-1}, \frac{\omega^{d_0} - \omega^{d_j}}{2}, u_{k,i+1}, \ldots, u_{k,j-1}, \frac{\omega^{d_0} + \omega^{d_j}}{2}, u_{k,j+1}, \ldots, u_{k,2^n}\right).$$

But in this case

$$\begin{aligned}
|u_{ki}|^2 + |u_{kj}|^2 &= \frac{1}{4}\left(\left|\omega^{d_0} - \omega^{d_j}\right|^2 + \left|\omega^{d_0} + \omega^{d_j}\right|^2\right) \\
&= \frac{1}{4}\left[\left(\omega^{d_0} - \omega^{d_j}\right)\left(\overline{\omega^{d_0}} - \overline{\omega^{d_j}}\right) + \left(\omega^{d_0} + \omega^{d_i}\right)\left(\overline{\omega^{d_0}} + \overline{\omega^{d_j}}\right)\right] \\
&= \frac{1}{4}\left(\omega^{d_0}\overline{\omega^{d_0}} - \omega^{d_0}\overline{\omega^{d_j}} - \omega^{d_j}\overline{\omega^{d_0}} + \omega^{d_j}\overline{\omega^{d_j}}\right) \\
&+ \frac{1}{4}\left(\omega^{d_0}\overline{\omega^{d_0}} + \omega^{d_0}\overline{\omega^{d_j}} + \omega^{d_j}\overline{\omega^{d_0}} + \omega^{d_j}\overline{\omega^{d_j}}\right) \\
&= \frac{1}{4}\left(2 \cdot \omega^{d_0}\overline{\omega^{d_0}} + 2 \cdot \omega^{d_j}\overline{\omega^{d_j}}\right) = \frac{1}{4}(2 + 2) = 1,
\end{aligned}$$

and since U is unitary that implies $\|U_k\|^2 = 1$ for any $k \in \{1, 2, \ldots, 2^n\}$, we derive that all components of U_k except, maybe,

$$u_{ki} = \frac{\omega^{d_0} - \omega^{d_j}}{2},$$

$$u_{kj} = \frac{\omega^{d_0} + \omega^{d_j}}{2},$$

are necessarily equal to zero.

Case 2: Without loss of generality assume that $\omega^{d_{ij}} = \omega^{d_i}$ and $\omega^{d_0} + \omega^{d_j} = 0$, then $u_{kj} = 0$.

Thus for any distinct $i, j \in \{1, 2, \ldots, 2^n\}$ either at least one of items u_{ki}, u_{kj} of the k-th row U_k is zero or in this row there are at most two nonzero items, whose form was considered in Case 1.

If for any row only Case 2 is met, the matrix is obviously monomial. So assume that some row of U, say k-th (in fact, then U has at least two rows of such form), has the form which is described in Case 1.

Consider vector (sign function) $F \in \mathbb{C}^{2^n}$ whose coordinates for $l = 1, 2, \ldots, 2^n$ are defined by

$$F_l = \begin{cases} \omega^{r_1}, & l = i, \\ \omega^{r_2}, & l = j, \\ 1, & \text{otherwise}, \end{cases}$$

where $r_1, r_2 \in \mathbb{Z}_q$ such that $r_1 < r_2$ and $r_2 - r_1 \neq q/2$, denote $\Delta r = r_2 - r_1$. We have

$$(UF)_k = u_{ki}\omega^{r_1} + u_{kj}\omega^{r_2} = \omega^{r_1}\left(\frac{\omega^{d_0} - \omega^{d_j}}{2} + \frac{\omega^{d_0} + \omega^{d_j}}{2}\omega^{\Delta r}\right) = \omega^{s+r_1},$$

for some $s \in \mathbb{Z}_q$. It is clear that it holds if and only if

$$\frac{\omega^{d_0} - \omega^{d_j}}{2} + \frac{\omega^{d_0} + \omega^{d_j}}{2}\omega^{\Delta r} = \omega^s.$$

Recall some trigonometric identities. For any real α, β it holds:

$$\cos\alpha + \cos\beta = 2\cos\left(\frac{\alpha+\beta}{2}\right)\cos\left(\frac{\alpha-\beta}{2}\right),$$
$$\cos\alpha - \cos\beta = -2\sin\left(\frac{\alpha+\beta}{2}\right)\sin\left(\frac{\alpha-\beta}{2}\right),$$
$$\sin\alpha \pm \sin\beta = 2\sin\left(\frac{\alpha\pm\beta}{2}\right)\cos\left(\frac{\alpha\mp\beta}{2}\right),$$
$$\sin(\alpha \pm \beta) = \sin\alpha\cos\beta \pm \cos\alpha\sin\beta,$$
$$\sin 2\alpha = 2\cos\alpha\sin\alpha.$$

Consider doubled real part of ω^s:

$$\begin{aligned}2\mathrm{Re}\left(\omega^s\right) &= \cos\left(\frac{2\pi d_0}{q}\right) - \cos\left(\frac{2\pi d_j}{q}\right) \\ &+ \cos\left(\frac{2\pi\left(d_0+\Delta r\right)}{q}\right) + \cos\left(\frac{2\pi\left(d_j+\Delta r\right)}{q}\right) \\ &= 2\cos\left(\frac{\pi\left(2d_0+\Delta r\right)}{q}\right)\cos\left(\frac{\pi\Delta r}{q}\right) \\ &- 2\sin\left(\frac{\pi\left(2d_j+\Delta r\right)}{q}\right)\sin\left(\frac{\pi\Delta r}{q}\right),\end{aligned}$$

and doubled imaginary part of ω^s:

$$\begin{aligned}2\mathrm{Im}\left(\omega^s\right) &= \sin\left(\frac{2\pi d_0}{q}\right) - \sin\left(\frac{2\pi d_j}{q}\right) \\ &+ \sin\left(\frac{2\pi\left(d_0+\Delta r\right)}{q}\right) + \sin\left(\frac{2\pi\left(d_j+\Delta r\right)}{q}\right) \\ &= 2\sin\left(\frac{\pi\left(2d_0+\Delta r\right)}{q}\right)\cos\left(\frac{\pi\Delta r}{q}\right) \\ &+ \sin\left(\frac{\pi\Delta r}{q}\right)\cos\left(\frac{\pi\left(2d_j+\Delta r\right)}{q}\right).\end{aligned}$$

For simplicity denote $\alpha = \pi\Delta r/q, \beta = \pi\left(2d_0+\Delta r\right)/q$ and $\gamma = \pi\left(2d_j+\Delta r\right)/q$. Since ω^s is a root of unity, its norm is equal to 1, hence

$$\begin{aligned}\mathrm{Re}^2\left(\omega^s\right)+\mathrm{Im}^2\left(\omega^s\right) &= \cos^2\alpha\cos^2\beta - 2\cos\alpha\sin\alpha\cos\beta\sin\gamma + \sin^2\alpha\sin^2\gamma \\ &+ \cos^2\alpha\sin^2\beta + 2\cos\alpha\sin\alpha\sin\beta\cos\gamma + \sin^2\alpha\cos^2\gamma \\ &= \cos^2\alpha\left(\cos^2\beta+\sin^2\beta\right) + \sin^2\alpha\left(\cos^2\gamma+\sin^2\gamma\right) \\ &+ 2\cos\alpha\sin\alpha\left(\sin\beta\cos\gamma - \cos\beta\sin\gamma\right) \\ &= 1 + \sin\left(2\alpha\right)\sin\left(\beta-\gamma\right) = 1,\end{aligned}$$

that is

$$\sin\left(2\alpha\right)\sin\left(\beta-\gamma\right) = 0.$$

Let the firt sine is zero, that is

$$2\alpha = \frac{2\pi\Delta r}{q} = \pi m,$$

for some $m \in \mathbb{Z}$. Then $\Delta r = mq/2$ but since $\Delta r \in \{1, 2, \ldots, q-1\}$, it is the case only for $\Delta r = q/2$, that is a contradiction with the choice of r_1, r_2.

If the second sine is zero, namely

$$\beta - \gamma = \frac{2\pi\,(d_0 - d_j)}{q} = \pi m',$$

for some $m' \in \mathbb{Z}$, again we have either $d_0 = d_j$ or $|d_0 - d_j| = q/2$ since $|d_0 - d_j| \in \{0, 1, \ldots, q-1\}$. But then either $\omega^{d_0} - \omega^{d_j} = 0$ or $\omega^{d_0} + \omega^{d_j} = 0$, that is in the k-th row there is exactly one nonzero element. □

List below some apparent properties of $\mathcal{U}_n^q$ which can be derived from Theorem 2:

Proposition 2. *Every operator from $\mathcal{U}_n^q$ preserves Lee and Hamming distance between generalized Boolean functions and Euclidian distance between their sign functions.*

Proposition 3. *The cardinality of $\mathcal{U}_n^q$ is the following*

$$|\mathcal{U}_n^q| = 2^n! \cdot q^{2^n}.$$

4.2 Matrix representation and connection with Markov's theorem for Boolean case

A mapping φ of the set of all Boolean functions in n variables to itself is called *isometric* if it preserves the Hamming distance between functions, that is

$$\mathrm{dist}_H(\varphi(f), \varphi(g)) = \mathrm{dist}_H(f, g),$$

for any $f, g \in \mathcal{F}_n$. The set of all isometric mappings of the set of all Boolean functions in n variables to itself in [16] was denoted by $\mathcal{I}_n$.

From Markov's theorem (1956) it follows that the general form of isometric mappings of all Boolean functions in n variables to itself is

$$f(x) \longrightarrow f(\pi(x)) \oplus g(x),$$

where π is a permutation on the set $\mathbb{F}_2^n$ and $g \in \mathcal{F}_n$ [19].

Theorem 2 can be reformulated in terms of mappings of (generalized) Boolean functions:

Theorem 3. *The action of any operator from $\mathcal{U}_n^q$ on the set $\mathcal{GF}_n^q$ is uniquely represented in the form*

$$f(x) \longrightarrow f\,(\pi(x)) + g(x),$$

where π is a permuation on $\mathbb{F}_2^n$ and $g \in \mathcal{GF}_n^q$.

Following [16] denote such operator by $\varphi_{\pi,g} \in \mathcal{U}_n^q$. So, for binary case we immediately obtain correspondence between $\mathcal{U}_n^2$ and $\mathcal{I}_n$:

Corollary 1. *For $q = 2$ there is an one-to-one correspondence between the set $\mathcal{U}_n^q$ and the set of isometric mappings of all Boolean functions in n variables into itself ($\mathcal{I}_n$), defined by Markov's theorem.*

Thus the considered set $\mathcal{U}_n^q$ is a some kind of generalization of the set $\mathcal{I}_n$ comprising the framework of sign functions.

Denote by $\{\mathbf{v}_k\}_{k=0}^{2^n-1}$ all binary vectors of length n considered in the lexicographical order.

By Theorem 2, provided that the (standard) basis is fixed, there is an one-to-one correspondence between $\mathcal{U}_n^q$ and the set of monomial matrices of order $2^n \times 2^n$ with nonzero elements from the set $\left\{1, \omega^1, \omega^2, \ldots, \omega^{q-1}\right\}$. Indeed, consider arbitrary mapping $\varphi_{\pi,g} \in \mathcal{U}_n^q$. Let it transforms a function $f \in \mathcal{GF}_n^q$ with sign function

$$F = \left(\omega^{f(\mathbf{v}_0)}, \omega^{f(\mathbf{v}_1)}, \ldots, \omega^{f(\mathbf{v}_{2^n-1})}\right) \in \mathbb{C}^{2^n},$$

to $f' \in \mathcal{GF}_n^q$ with sign function

$$F' = \left(\omega^{f'(\mathbf{v}_0)}, \omega^{f'(\mathbf{v}_1)}, \ldots, \omega^{f'(\mathbf{v}_{2^n-1})}\right) \in \mathbb{C}^{2^n},$$

that is $F' = UF$, where U is a matrix of $\varphi_{\pi,g}$. Namely this matrix is

$$k\begin{pmatrix} & & & \overset{j}{\vdots} & & & \\ & & & 0 & & & \\ & & & \vdots & & & \\ \ldots & 0 & \ldots & \omega^{g(\mathbf{v}_{k-1})} & \ldots & 0 & \ldots \\ & & & \vdots & & & \\ & & & 0 & & & \\ & & & \vdots & & & \end{pmatrix},$$

in which in the k-th row a nonzero element $\omega^{g(\mathbf{v}_{k-1})}$ is in the j-th column, where $(j-1)$ is a number with binary representation $\pi(\mathbf{v}_{k-1})$. So the k-th component of the vector $F' = UF$ is equal to

$$\omega^{f'(\mathbf{v}_{k-1})} = \omega^{f(\pi(\mathbf{v}_{k-1}))} \cdot \omega^{g(\mathbf{v}_{k-1})} = \omega^{f(\pi(\mathbf{v}_{k-1}))+g(\mathbf{v}_{k-1})},$$

for any $k \in \{1, 2, \ldots, 2^n\}$, that is equivalent to

$$f'(x) = f(\pi(x)) + g(x), \quad x \in \mathbb{F}_2^n.$$

4.3 Unitary operators preserving eigenspaces of the duality mapping

Assume that $n \geqslant 4$ is an even integer.

In this subsection we characterize operators from $\mathcal{U}_n^q$ which preserve (anti-)self-duality of gbent function.

At first, similar to [16], the question of how the property of preserving self-duality is connected with preserving anti-self-duality is studied.

Proposition 4. *For an operator $\varphi_{\pi,g} \in \mathcal{U}_n^q$ with a matrix U the following conditions are equivalent:*

1) $\varphi_{\pi,g}$ preserves self-duality;

2) $\varphi_{\pi,g}$ preserves anti-self-duality;

3) $U\mathcal{H}_n = \mathcal{H}_n U$.

Proof. It is enough to note that by Proposition 1 for $n \geqslant 4$ there exist a subset $\{f_i\}_{i=1}^{2^{n-1}} \subseteq \mathrm{SB}_q^+(n)$ with linearly independent sign functions $\{F_i\}_{i=1}^{2^{n-1}} \subseteq \mathrm{Ker}\,(\mathcal{H}_n - I_{2^n})$ and a subset $\{g_i\}_{i=1}^{2^{n-1}} \subseteq \mathrm{SB}_q^-(n)$ with linearly independent sign functions $\{G_i\}_{i=1}^{2^{n-1}} \subseteq \mathrm{Ker}\,(\mathcal{H}_n + I_{2^n})$.

The rest of the proof, comprising these subsets of (anti-)self-dual gbent functions, follows as in [16] (Proposition 2). □

In paper [16] isometric mappings of all Boolean functions in n variables into itself which preserve self-duality were completely characterized, namely it was proved that isometric mapping $f(x) \to f(\pi(x)) \oplus g(x)$ preserves self-duality if and only if

$$\pi(x) = L\,(x \oplus c)\,,\ x \in \mathbb{F}_2^n, \quad g(x) = \langle c, x\rangle \oplus d, \quad x \in \mathbb{F}_2^n,$$

where $L \in \mathcal{O}_n$, $c \in \mathbb{F}_2^n$, $\mathrm{wt}_H(c)$ is even, $d \in \mathbb{F}_2$. It was also shown that isometric mapping of all Boolean functions in n variables into itself preserves self-duality if and only if it preserves anti-self-duality.

The next result characterizes operators from $\mathcal{U}_n^q$ that preserve (anti-)self-duality of gbent function.

Theorem 4. *Operator $\varphi_{\pi,g} \in \mathcal{U}_n^q$ preserves (anti-)self-duality of generalized bent function if and only if*

$$\pi(x) = L\,(x \oplus c)\,, \quad g(x) = \frac{q}{2}\,\langle c, x\rangle + d,\ x \in \mathbb{F}_2^n, \quad x \in \mathbb{F}_2^n,$$

where $L \in \mathcal{O}_n$, $c \in \mathbb{F}_2^n$, $\mathrm{wt}_H(c)$ is even, $d \in \mathbb{Z}_q$.

Proof. Let $f \in \mathrm{SB}_q^+(n) \cup \mathrm{SB}_q^-(n)$ that is $\widetilde{f} = f + \frac{q}{2}\varepsilon$ for some $\varepsilon \in \mathbb{F}_2$. Consider a function $g(x) = f\left(L\left(x \oplus c\right)\right) + \frac{q}{2}\langle c, x\rangle + d$, where $L \in \mathcal{O}_n$, $c \in \mathbb{F}_2^n$, $\mathrm{wt}(c)$ is even, $d \in \mathbb{Z}_q$. Its generalized Walsh–Hadamard transform is

$$\begin{aligned} H_g(y) &= \sum_{x \in \mathbb{F}_2^n} \omega^{g(x)} (-1)^{\langle x, y\rangle} = \sum_{x \in \mathbb{F}_2^n} \omega^{f(L(x \oplus c)) + \frac{q}{2}\langle c, x\rangle + d + \frac{q}{2}\langle x, y\rangle} \\ &= \omega^d \sum_{x \in \mathbb{F}_2^n} \omega^{\frac{q}{2}\langle x, y \oplus c\rangle + f(L(x \oplus c))} = \omega^d \sum_{z \in \mathbb{F}_2^n} \omega^{\frac{q}{2}\left\langle L^{-1} z \oplus c, y \oplus c\right\rangle + f(z)} \\ &= \omega^{d + \frac{q}{2}\langle c, y\rangle + \frac{q}{2}\langle c, c\rangle} \sum_{z \in \mathbb{F}_2^n} \omega^{\frac{q}{2}\langle z, L(y \oplus c)\rangle + f(z)} \\ &= \omega^{d + \frac{q}{2}\langle c, y\rangle} 2^{n/2} \omega^{\widetilde{f}(L(y \oplus c))} = 2^{n/2} \omega^{f(L(y \oplus c)) + \frac{q}{2}\langle c, y\rangle + d + \frac{q}{2}\varepsilon} \\ &= 2^{n/2} \omega^{g(y) + \frac{q}{2}\varepsilon} = 2^{n/2} \omega^{\widetilde{g}(y)}, \end{aligned}$$

hence $\widetilde{g}(y) = g(y) + \frac{q}{2}\varepsilon$ for any $y \in \mathbb{F}_2^n$, so the opposite direction follows.

Assume that U is a matrix of the operator $\varphi_{\pi,g} \in \mathcal{U}_n^q$ preserving (anti-)self-duality. Let the considered mapping has form

$$f(x) \longrightarrow f\left(\pi(x)\right) + g(x),$$

where π is a permuation on $\mathbb{F}_2^n$ and $g \in \mathcal{GF}_n^q$.

Now consider the relation $UH_n = H_n U$, imposed by Proposition 4 in details. Recall that

$$H_n = \begin{pmatrix} (-1)^{\langle \mathbf{v}_0, \mathbf{v}_0\rangle} & (-1)^{\langle \mathbf{v}_0, \mathbf{v}_1\rangle} & \dots & (-1)^{\langle \mathbf{v}_0, \mathbf{v}_{2^n-1}\rangle} \\ (-1)^{\langle \mathbf{v}_1, \mathbf{v}_0\rangle} & (-1)^{\langle \mathbf{v}_1, \mathbf{v}_1\rangle} & \dots & (-1)^{\langle \mathbf{v}_1, \mathbf{v}_{2^n-1}\rangle} \\ \vdots & \vdots & \ddots & \vdots \\ (-1)^{\langle \mathbf{v}_{2^n-1}, \mathbf{v}_0\rangle} & (-1)^{\langle \mathbf{v}_{2^n-1}, \mathbf{v}_1\rangle} & \dots & (-1)^{\langle \mathbf{v}_{2^n-1}, \mathbf{v}_{2^n-1}\rangle} \end{pmatrix}.$$

and U is the matrix

$$(k+1) \begin{pmatrix} & & & \pi\left(\mathbf{v}_i\right) & & & \\ & & & 0 & & & \\ & & & \vdots & & & \\ & & & 0 & & & \\ 0 & \dots & 0 & \omega^{g(\mathbf{v}_k)} & 0 & \dots & 0 \\ & & & 0 & & & \\ & & & \vdots & & & \\ & & & 0 & & & \end{pmatrix},$$

in which in the row with number $(i+1) \in \{1, 2, \dots, 2^n\}$ a nonzero element is in the $(j+1)$-th column, where j is a number with binary representation $\pi\left(\mathbf{v}_k\right)$.

Fix arbitrary $k, j \in \{0, 1, \ldots, 2^n - 1\}$, we have

$$(UH_n)_{k+1,j+1} = \omega^{g(\mathbf{v}_k)}(-1)^{\langle \pi(\mathbf{v}_k), \mathbf{v}_j \rangle}.$$

In order to obtain $(H_n U)_{k+1,j+1}$ consider matrix U in the following form

$$\pi^{-1}(\mathbf{v}_j) \begin{pmatrix} & & & (j+1) & & & \\ & & & 0 & & & \\ & & & \vdots & & & \\ & & & 0 & & & \\ 0 & \ldots & 0 & \omega^{g\left(\pi^{-1}(\mathbf{v}_j)\right)} & 0 & \ldots & 0 \\ & & & 0 & & & \\ & & & \vdots & & & \\ & & & 0 & & & \end{pmatrix}.$$

Then it clear that

$$(H_n U)_{k+1,j+1} = (-1)^{\left\langle \mathbf{v}_k, \pi^{-1}(\mathbf{v}_j) \right\rangle} \omega^{g\left(\pi^{-1}(\mathbf{v}_j)\right)}.$$

Thus for any $k, j \in \{0, 1, \ldots, 2^n - 1\}$ we get the relation

$$\omega^{g(\mathbf{v}_i)}(-1)^{\langle \pi(\mathbf{v}_i), \mathbf{v}_j \rangle} = (-1)^{\left\langle \mathbf{v}_k, \pi^{-1}(\mathbf{v}_j) \right\rangle} \omega^{g\left(\pi^{-1}(\mathbf{v}_j)\right)},$$

or, equivalently, for any $x, y \in \mathbb{F}_2^n$ it is

$$g(x) + \frac{q}{2} \langle \pi(x), y \rangle = \frac{q}{2} \left\langle x, \pi^{-1}(y) \right\rangle + g\left(\pi^{-1}(y)\right), \tag{3}$$

considered by modulo q.

Put zero vector $y = \mathbf{0} \in \mathbb{F}_2^n$ in (3). Then we obtain that g is an affine generalized Boolean function:

$$g(x) = \frac{q}{2} \left\langle x, \pi^{-1}(\mathbf{0}) \right\rangle + g\left(\pi^{-1}(\mathbf{0})\right).$$

Put the expression for g in (3):

$$\frac{q}{2} \left\langle x, \pi^{-1}(\mathbf{0}) \right\rangle + g\left(\pi^{-1}(\mathbf{0})\right) + \frac{q}{2} \langle \pi(x), y \rangle = \frac{q}{2} \left\langle x, \pi^{-1}(y) \right\rangle + \frac{q}{2} \left\langle \pi^{-1}(y), \pi^{-1}(\mathbf{0}) \right\rangle + g\left(\pi^{-1}(\mathbf{0})\right),$$

and after elimination of coinciding terms

$$\frac{q}{2} \left\langle x, \pi^{-1}(\mathbf{0}) \right\rangle + \frac{q}{2} \langle \pi(x), y \rangle = \frac{q}{2} \left\langle x, \pi^{-1}(y) \right\rangle + \frac{q}{2} \left\langle \pi^{-1}(y), \pi^{-1}(\mathbf{0}) \right\rangle. \tag{4}$$

Since the equality (4) should be considered by modulo q, we only care about the parity of components of both sides, thus, for any $x, y \in \mathbb{F}_2^n$ having the following equality

$$\left\langle x, \pi^{-1}(\mathbf{0})\right\rangle \oplus \langle \pi(x), y\rangle = \left\langle x, \pi^{-1}(y)\right\rangle \oplus \left\langle \pi^{-1}(y), \pi^{-1}(\mathbf{0})\right\rangle.$$

At first one can notice that the permutation π must be affine. The reason is that the left part is linear in variable y while the right part is linear in variable x. So let $\pi(x) = L(x \oplus c), x \in \mathbb{F}_2^n$, for some $L \in \mathrm{GL}(n)$ and $c \in \mathbb{F}_2^n$. Then

$$\langle x, c\rangle \oplus \langle L(x \oplus c), y\rangle = \left\langle x, L^{-1}y \oplus c\right\rangle \oplus \left\langle L^{-1}y \oplus c, c\right\rangle,$$
$$\langle L(x \oplus c), y\rangle = \left\langle x, L^{-1}y\right\rangle \oplus \left\langle L^{-1}y, c\right\rangle \oplus \langle c, c\rangle,$$

For $y = \mathbf{0} \in \mathbb{F}_2^n$ we have $\langle c, c\rangle = 0$, hence $\mathrm{wt}_H(c)$ must be an even number. Take it into account and continue

$$\langle Lx, y\rangle \oplus \langle Lc, y\rangle = \left\langle x, L^{-1}y\right\rangle \oplus \left\langle L^{-1}y, c\right\rangle,$$
$$\left\langle \left(L \oplus \left(L^{-1}\right)^T\right)(x \oplus c), y\right\rangle = 0.$$

The last one holds for any $x, y \in \mathbb{F}_2^n$, therefore it should be $L^{-1} = L^T$, that is $L \in \mathcal{O}_n$.

Finally, we can obtain a form of the function g:

$$g(x) = \frac{q}{2}\langle c, x\rangle + g(c), \ x \in \mathbb{F}_2^n,$$

where $g(c)$ can be an arbitrary element of $\mathbb{Z}_q$.

Thus the collection of all aforementined necessary conditions for the parameters $L \in \mathrm{GL}(n), c \in \mathbb{F}_2^n, d \in \mathbb{Z}_q$, yields:

$$\begin{cases} L \in \mathcal{O}_n, \\ \mathrm{wt}_H(c) \text{ is even}, \\ g(x) = \frac{q}{2}\langle c, x\rangle + d, \ x \in \mathbb{F}_2^n, \end{cases} \tag{5}$$

that concludes the proof. □

4.4 Unitary operators which define bijections between the eigenspaces of the duality mapping

Assume that $n \geqslant 4$ is an even integer.

In this subsection we characterize operators from $\mathcal{U}_n^q$ which define one-to-one correspondence between the sets of self-dual and anti-self-dual gbent functions.

Proposition 5. *An operator $\varphi_{\pi,g} \in \mathcal{U}_n^q$ with matrix U defines a bijection between $\mathrm{SB}_q^+(n)$ and $\mathrm{SB}_q^-(n)$ if and only if $U\mathcal{H}_n = -\mathcal{H}_n U$.*

Proof. The opposite direction is clear since if $\mathcal{H}_n A = -A\mathcal{H}_n$, then for any sign functions F, G of $f \in \mathrm{SB}_q^+(n)$ and $g \in \mathrm{SB}_q^-(n)$ respectively it holds

$$\mathcal{H}_n(AF) = -A(\mathcal{H}_n F) = -AF,$$

$$\mathcal{H}_n(AG) = -A(\mathcal{H}_n G) = AG,$$

hence the mapping is a bijection between $\mathrm{SB}_q^+(n)$ and $\mathrm{SB}_q^-(n)$.

By Proposition 1 one can find subsets $\{f_i\}_{i=1}^{2^{n-1}} \subseteq \mathrm{SB}^+(n)$ with linearly independent sign functions $\{F_i\}_{i=1}^{2^{n-1}} \subseteq \mathrm{Ker}(\mathcal{H}_n - I_{2^n})$ and $\{g_i\}_{i=1}^{2^{n-1}} \subseteq \mathrm{SB}^-(n)$ with linearly independent sign functions $\{G_i\}_{i=1}^{2^{n-1}} \subseteq \mathrm{Ker}(H_n + I_{2^n})$ as in the proof of Proposition 4.

The rest of the proof, comprising these subsets of (anti-)self-dual gbent functions, follows as in [16] (Proposition 3). □

In paper [16] isometric mappings of all Boolean functions in n variables into itself which define bijections between the sets of self-dual and anti-self-dual Boolean bent functions in n variables were completely characterized, namely it was proved that isometric mapping $f(x) \to f(\pi(x)) \oplus g(x)$ bijection between $\mathrm{SB}^+(n)$ and $\mathrm{SB}^-(n)$ if and only if

$$\pi(x) = L(x \oplus c),\ x \in \mathbb{F}_2^n,$$

and

$$g(x) = \langle c, x\rangle \oplus d,\ x \in \mathbb{F}_2^n,$$

where $L \in \mathcal{O}_n$, $c \in \mathbb{F}_2^n$, $\mathrm{wt}_H(c)$ is odd, $d \in \mathbb{F}_2$. These conditions almost equal to (5), except the parity of $\mathrm{wt}_H(c)$.

The next result characterizes operators from $\mathcal{U}_n^q$ that comprise one-to-one correspondence between $\mathrm{SB}_q^+(n)$ and $\mathrm{SB}_q^-(n)$.

Theorem 5. *Operator $\varphi_{\pi,g} \in \mathcal{U}_n^q$ defines a bijections between $\mathrm{SB}_q^+(n)$ and $\mathrm{SB}_q^-(n)$ if and only if*

$$\pi(x) = L(x \oplus c), \quad g(x) = \frac{q}{2}\langle c, x\rangle + d, \quad x \in \mathbb{F}_2^n,$$

where $L \in \mathcal{O}_n$, $c \in \mathbb{F}_2^n$, $\mathrm{wt}_H(c)$ is odd, $d \in \mathbb{Z}_q$.

Proof. Let $f \in \mathrm{SB}_q^+(n) \cup \mathrm{SB}_q^-(n)$ that is $\widetilde{f} = f + \frac{q}{2}\varepsilon$ for some $\varepsilon \in \mathbb{F}_2$. Consider a function $g(x) = f(L(x \oplus c)) + \frac{q}{2}\langle c, x\rangle + d$, where $L \in \mathcal{O}_n$, $c \in \mathbb{F}_2^n$, $\mathrm{wt}(c)$

is even, $d \in \mathbb{Z}_q$. Its generalized Walsh–Hadamard transform is

$$\begin{aligned}\mathcal{H}_g(y) &= \sum_{x\in\mathbb{F}_2^n} \omega^{g(x)}(-1)^{\langle x,y\rangle} = \sum_{x\in\mathbb{F}_2^n} \omega^{f(L(x\oplus c))+\frac{q}{2}\langle c,x\rangle+d+\frac{q}{2}\langle x,y\rangle} \\ &= \omega^d \sum_{x\in\mathbb{F}_2^n} \omega^{\frac{q}{2}\langle x,y\oplus c\rangle+f(L(x\oplus c))} = \omega^d \sum_{z\in\mathbb{F}_2^n} \omega^{\frac{q}{2}\left\langle L^{-1}z\oplus c,y\oplus c\right\rangle+f(z)} \\ &= \omega^{d+\frac{q}{2}\langle c,y\rangle+\frac{q}{2}\langle c,c\rangle} \sum_{z\in\mathbb{F}_2^n} \omega^{\frac{q}{2}\langle z,L(y\oplus c)\rangle+f(z)} \\ &= \omega^{d+\frac{q}{2}\langle c,y\rangle} 2^{n/2} \omega^{\widetilde{f}(L(y\oplus c))} = 2^{n/2}\omega^{f(L(y\oplus c))+\frac{q}{2}\langle c,y\rangle+d+\frac{q}{2}\varepsilon} \\ &= 2^{n/2}\omega^{g(y)+\frac{q}{2}\varepsilon} = 2^{n/2}\omega^{\widetilde{g}(y)},\end{aligned}$$

hence $\widetilde{g}(y) = g(y) + \frac{q}{2}\varepsilon$ for any $y \in \mathbb{F}_2^n$. The opposite direction is proved.

Assume that U is a matrix of the operator $\varphi_{\pi,g} \in \mathcal{U}_n^q$ which defines a bijections between $\mathrm{SB}_q^+(n)$ and $\mathrm{SB}_q^-(n)$. As well as in the proof of the Theorem 4, we use that

$$(UH_n)_{k+1,j+1} = \omega^{g(\mathbf{v}_k)}(-1)^{\langle \pi(\mathbf{v}_k),\mathbf{v}_j\rangle},$$

$$(H_nU)_{k+1,j+1} = (-1)^{\left\langle \mathbf{v}_k,\pi^{-1}(\mathbf{v}_j)\right\rangle}\omega^{g\left(\pi^{-1}(\mathbf{v}_j)\right)},$$

for any $k, j \in \{0, 1, \ldots, 2^n - 1\}$.

From Proposition 5 it follows that $UH_n = -H_nU$ that implies $(UH_n)_{k+1,j+1} = -(H_nU)_{k+1,j+1}$ for any $k, j \in \{0, 1, \ldots, 2^n - 1\}$, hence the following relation must hold

$$-\omega^{g(\mathbf{v}_k)}(-1)^{\langle \pi(\mathbf{v}_k),\mathbf{v}_j\rangle} = (-1)^{\left\langle \mathbf{v}_k,\pi^{-1}(\mathbf{v}_j)\right\rangle}\omega^{g\left(\pi^{-1}(\mathbf{v}_j)\right)}$$

or, equivalently,

$$g(x) + \frac{q}{2}\langle \pi(x), y\rangle + \frac{q}{2} = \frac{q}{2}\left\langle x, \pi^{-1}(y)\right\rangle + g\left(\pi^{-1}(y)\right), \tag{6}$$

for any $x, y \in \mathbb{F}_2^n$ considered by modulo q.

The rest of the proof is similar to the proof of Theorem 4 with only the difference in

$$\langle L(x \oplus c), y\rangle = \left\langle x, L^{-1}y\right\rangle \oplus \left\langle L^{-1}y, c\right\rangle \oplus \langle c, c\rangle \oplus 1,$$

and the expression for the function g:

$$g(x) = \frac{q}{2}\langle c, x\rangle + g(c) + \frac{q}{2},\ x \in \mathbb{F}_2^n,$$

where $g(c)$ can be an arbitrary element of $\mathbb{Z}_q$.

Again, the collection of all aforementined necessary conditions, including parameters $L \in \mathrm{GL}(n), c \in \mathbb{F}_2^n, d \in \mathbb{Z}_q$, yields:

$$\begin{cases} L^{-1} = L^T, \\ \mathrm{wt}_H(c) \text{ is odd}, \\ g(x) = \frac{q}{2}\langle c, x\rangle + d, \ x \in \mathbb{F}_2^n, \end{cases}$$

that concludes the proof. □

4.5 The Rayleigh quotient of (generalized) Boolean function

In this subsection operators from the set $\mathcal{U}_n^q$, which preserve and change the sign of the Rayleigh quotient (Rayleigh ratio) of the Sylvester Hadamard matrix defined for every generalized Boolean function in n variables, are studied.

In [1] *the Rayleigh quotient* S_f of a Boolean function $f \in \mathcal{F}_n$ was defined as

$$S_f = \sum_{x,y \in \mathbb{F}_2^n} (-1)^{f(x) \oplus f(y) \oplus \langle x,y \rangle} = \sum_{y \in \mathbb{F}_2^n} (-1)^{f(y)} W_f(y),$$

and when $f \in \mathcal{B}_n$ the *normalized Rayleigh quotient* N_f is a number

$$N_f = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x) \oplus \widetilde{f}(x)} = 2^{-n/2} S_f.$$

It is known [1] (Theorem 3.1) that the value of S_f is at most $2^{3n/2}$ with equality if and only if f is self-dual bent, and at least $\left(-2^{3n/2}\right)$ with equality if and only if f is anti-self-dual bent.

All isometric mappings from the set $\mathcal{I}_n$ that preserve the Rayleigh quotient of every Boolean function in n variables (or change its sign) were characterized in [16]. It was made by showing the direct link between perserving the Rayleigh quoient and preserving the self-duality. Also it was proved that bijectivity between the sets $\mathrm{SB}^+(n)$ and $\mathrm{SB}^-(n)$ is correlated with the change of sign of the Rayleigh quoient.

In [28] the authors studied the Rayleigh quotient of generalized Boolean (bent) functions from $\mathcal{GF}_n^4$. For a generalized Boolean function $f \in \mathcal{GF}_n^4$ they defined

$$R(f) = 2^{-n} \sum_{x,y \in \mathbb{F}_2^n} i^{f(x)-f(y)} (-1)^{\langle x,y \rangle},$$

and proved, see [28] (Theorem 7), the bound $-2^{n/2} \leqslant R(f) \leqslant 2^{n/2}$ with equalities if and only if f is self-dual quaternary bent $(+2^{n/2})$ or self-dual quaternary bent $(-2^{n/2})$.

Define *the Rayleight quotient* R_f of (generalized) Boolean function $f \in \mathcal{GF}_n^q$ as follows

$$R_f = 2^{-n} \sum_{x,y \in \mathbb{F}_2^n} \omega^{f(x)-f(y)} (-1)^{\langle x,y \rangle}.$$

The matrix-vector representation the Rayleight quotient for a generalized Boolean function $f \in \mathcal{FG}_n^q$ with sign function F is

$$R_f = 2^{-n} \sum_{x \in \mathbb{F}_2^n} \overline{\omega^{f(x)}} \sum_{y \in \mathbb{F}_2^n} \omega^{f(y)} (-1)^{\langle x,y \rangle} = \frac{\langle F, H_n F \rangle}{\langle F, F \rangle}.$$

By the same technique as in the proof of [28] (Theorem 7) it is possible to prove that the same bound $-2^{n/2} \leqslant R_f \leqslant 2^{n/2}$ holds with equalities met if and only if f is self-dual gbent $(+2^{n/2})$ or anti-self-dual gbent $(-2^{n/2})$.

The mentioned correlation with preserving of self-duality and bijectivity for Boolean case also stands for the Rayleigh quotient of generalized Boolean function.

Theorem 6. *For $n \geqslant 4$ an operator $\varphi_{\pi,g} \in \mathcal{U}_n^q$*

- *preserves the Rayleigh quotient if and only if it preserves (anti-)self-duality;*
- *changes the sign of the Rayleigh quotient if and only if it is a bijection between the sets $\mathrm{SB}_q^+(n)$ and $\mathrm{SB}_q^-(n)$.*

The proofs of these statements are similar to those provided in [16] (Theorems 3 and 4) and are omitted.

Thus the exact form of operators, which preserve the Rayleigh quotient or change its sign, are described by Theorems 4 and 5.

It also follows that

Corollary 2. *An operator $\varphi_{\pi,g} \in \mathcal{U}_n^q$, which preserves the Rayleigh quotient or changes the sign of the Rayleigh quotient, also preserves gbentness.*

4.6 Classification of quaternary self-dual bent functions in 4 variables

By using the mappings from Theorem 4 we can clarify, for instance, the known classification of quaternary self-dual bent functions in 4 variables given in [28] and formed by 8 classes.

Representative from equivalence class	Size
0220202022000000	24
2022220222020200	16
0330313133110110	48
0330302132010110	24
3123231322030300	96
1321213122010100	96
2123230332121210	48
0220213023100000	48
Number of quaternary self-dual bent functions in four variables	400

Table 1: Classification of quaternary self-dual bent functions in 4 variables from [28]

Namely, the representatives with vectors of values (0330302132010110) and (3123231322030300) from the classes 4 and 5 respectively are related by the transformation

$$f(x) \longrightarrow f(L(x \oplus c)) + \frac{q}{2}\langle c, x\rangle + d,$$

where

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad c = (1001), \quad d = 3.$$

The representatives with vectors of values (2022220222020200) and (2123230332121210) from the classes 2 and 7 respectively are related by the transformation

$$f(x) \longrightarrow f(L(x \oplus c)) + \frac{q}{2}\langle c, x\rangle + d,$$

where

$$L = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad c = (0101), \quad d = 1.$$

Thus, the classification of quaternary self-dual bent functions in 4 variables is given in the Table 2.

Representative from equivalence class	Size
0220202022000000	24
2022220222020200	64
0330313133110110	48
0330302132010110	120
1321213122010100	96
0220213023100000	48
Number of quaternary self-dual bent functions in four variables	400

Table 2: Classification of quaternary self-dual bent functions in 4 variables

5 The duality mapping and unitary operators

In this section for the case of even n we study the question if there exists an operator from the set $\mathcal{U}_n^q$, that transforms every regular gbent function to its dual gbent function.

Theorem 7. *If n is an even number, then in $\mathcal{U}_n^q$ there is no such operator which assigns the dual bent function to every regular bent function from the set $\mathcal{GB}_n^q$.*

Proof. Consider the following set of gbent functions:

$$B = \left\{ \frac{q}{2} f | f \in \mathcal{B}_n \right\} \subset \mathcal{GB}_n^q.$$

It is clear that all gbent functions from B are regular ones with the values from the set $\{0, q/2\}$. Assume the desired operator exists, let it be

$$\varphi_{\pi,g} : f(x) \longrightarrow f\left(\pi(x)\right) + g(x),$$

for some permutation π and generalized Boolean function $g \in \mathcal{GF}_n^q$. Then, in order to transform gebnt functions from the set B to their duals, the function g also must have values in $\{0, q/2\}$. It means that in fact we have a reduction to Boolean case, since all considered generalized Boolean functions, namely that ones from the set B and the function g, have values from the set $\{0, q/2\}$.

Then non-existence of isometric mapping of the set of all Boolean functions in n variables into itself which assigns to every bent functions its dual implies non-existence of the considered unitary operator. It is known [14] that there is no such isometric mapping, hence the result follows.

□

Thus, Theorem 7 is a generalization of the known result of non-existence for the Boolean case, but here we consider all mappings from the set $\mathcal{U}_n^q$.

It is interesting to study the same question for the case of an odd number of variables n.

6 Conclusion

In this paper the action of linear operators of the form $\mathbb{C}^{2^n} \to \mathbb{C}^{2^n}$ on the generalized Boolean functions in n variables via their sign functions was defined. The interconnection between unitary operators that transform the set of all generalized Boolean functions in n variables into itself and the duality mapping was studied. The known classification of quaternary self-dual bent functions is clarified. It follows that the set $\mathcal{U}_n^q$ can be seen as an initial expansion of the set of automorphisms of the Boolean functions in n variables to generalized Boolean functions. For the future study it can be interesting to go beyond the set $\mathcal{U}_n^q$ that is to consider operators that transform some desired subsets of Boolean functions into itself but not neccesarily all generalized Boolean functions. Examples of such problems deal with gbent or self-dual gbent functions. The question of determing the connection between the set $\mathcal{U}_n^q$ and duality mapping for odd n is an open one.

Acknowledgments. The work is supported by the Mathematical Center in Akademgorodok under agreement No. 075-15-2019-1613 with the Ministry of Science and Higher Education of the Russian Federation and Laboratory of Cryptography JetBrains Research.

References

[1] Carlet C., Danielson L.E., Parker M.G., Solé. P., "Self-dual bent functions", *Int. J. Inform. Coding Theory*, **1** (2010), 384–399.

[2] Carlet C., *Boolean Functions for Cryptography and Coding Theory*, Cambridge Univ. Press, London, 2020, 620.

[3] Çeşmelioğlu A., Meidl W. Pott A., "On the dual of (non)-weakly regular bent functions and self-dual bent functions", *Adv. Math. Commun.*, **7**:4 (2013), 425–440.

[4] Cusick T.W., Stănică P., *Cryptographic Boolean functions and applications*, Acad. Press, London, 2017, 288.

[5] Feulner T., Sok L., Solé P., Wassermann A., "Towards the Classification of Self-Dual Bent Functions in Eight Variables", *Des. Codes Cryptogr.*, **68**:1 (2013), 395–406.

[6] Gangopadhyay S., Poonia V.S., Aggarwal D., Parekh R., "Generalized Boolean Functions and Quantum Circuits on IBM-Q", 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2019.

[7] Hodžić S., Meidl W., Pasalic E., "Full Characterization of Generalized Bent Functions as (Semi)-Bent Spaces, Their Dual, and the Gray Image", *IEEE Trans. Inform. Theory*, **64**:7 (2018), 5432–5440.

[8] Hou X.-D., "Classification of self dual quadratic bent functions", *Des. Codes Cryptogr.*, **63**:2 (2012), 183–198.

[9] Hou X.-D., "Classification of p-ary self dual quadratic bent functions, p odd", *Journal of Algebra*, **391** (2013), 62–81.

[10] Hyun J.Y., Lee H., Lee Y., "MacWilliams duality and Gleason-type theorem on self-dual bent functions", *Des. Codes Cryptogr.*, **63**:3 (2012), 295–304.

[11] Janusz G.J., "Parametrization of self-dual codes by orthogonal matrices", *Finite Fields Appl.*, **13**:3 (2007), 450–491.

[12] Kumar P.V., Scholtz R.A., Welch L.R., "Generalized bent functions and their properties", *J. Comb. Theory Series A*, **40** (1985), 90–107.

[13] Luo G., Cao X., Mesnager S., "Several new classes of self-dual bent functions derived from involutions", *Cryptogr. Commun.*, **11**:6 (2019).

[14] Kutsenko A.V., "On some properties of known isometric mappings of the set of bent functions", *Prikl. Diskr. Mat. Suppl. (Applied Discrete Math. Supplement)*, **10** (2020), 43–44.

[15] Kutsenko A., "Metrical properties of self-dual bent functions", *Des. Codes Cryptogr.*, **88**:1 (2020), 201–222.

[16] Kutsenko A., "The group of automorphisms of the set of self-dual bent functions", *Cryptogr. Commun.*, **12**:5 (2020), 881–898.

[17] Kutsenko A., Tokareva N., "Metrical properties of the set of bent functions in view of duality", *Prikl. Diskr. Mat. (Applied Discrete Math.)*, **49** (2020), 18–34.

[18] Logachev O.A., Sal'nikov A.A., Yashchenko V.V., "Bent functions on a finite Abelian group", *Discrete Math. Appl.*, **7**:6 (1997), 547–564.

[19] Markov A. A., "On transformations without error propagation", *Selected Works, Vol. II: Theory of Algorithms and Constructive Mathematics. Mathematical Logic. Informatics and Related Topics, MTsNMO, Moscow*, 2003, 70–93, In Russian.

[20] Martinsen T., Meidl W., Stănică P., "Partial spread and vectorial generalized bent functions", *Des. Codes Cryptogr.*, **85**:1 (2017), 1–13.

[21] Mesnager S., "Several New Infinite Families of Bent Functions and Their Duals", *IEEE Trans. Inf. Theory*, **60**:7 (2014), 4397–4407.

[22] Mesnager S., *Bent Functions: Fundamentals and Results,*, Springer, Berlin, 2016, 544 p.

[23] Mesnager S., Tang C., Qi Y., Wang L., Wu B., Feng K., "Further Results on Generalized Bent Functions and Their Complete Characterization", *IEEE Trans. Inform. Theory*, **64**:7 (2018), 5441–5452.

[24] Paterson K.G., "Generalized Reed–Muller Codes and Power Control in OFDM Modulation", *IEEE Trans. Inform. Theory*, **46**:1 (2000), 104–120.

[25] Riera C., Stănică P., Gangopadhyay S., "Generalized bent Boolean functions and strongly regular Cayley graphs", *Discrete Appl. Math.*, **283** (2020), 367–374.

[26] Rothaus O.S., "On bent functions", *J. Combin. Theory. Ser. A*, **20**:3 (1976), 300–305.

[27] Schmidt K.-U., "Quaternary constant-amplitude codes for multicode CDMA", *IEEE Trans. Inform. Theory*, **55**:4 (2009), 1824–1832.

[28] Sok L., Shi M., Solé. P., "Classification and Construction of quaternary self-dual bent functions", *Cryptogr. Commun.*, **10**:2 (2018), 277–289.

[29] Solodovnikov V.I., "Bent functions from a finite Abelian group into a finite Abelian group", *Discret. Math. Appl.*, **12**:2 (2002), 111–126.

[30] Stănică P., Martinsen T., Gangopadhyay S., Singh B. K., "Bent and generalized bent Boolean functions", *Des. Codes Cryptogr.*, **69**:1 (2013), 77–94.

[31] Tang C., Xiang C., Qi Y., Feng K., "Complete characterization of generalized bent and2k-bent Boolean functions", *IEEE Trans. Inform. Theory*, **63** (2017), 4668–4674.

[32] Tokareva N.N., "Generalizations of bent functions — a survey", *J. Appl. Ind. Math.*, **5**:1 (2011), 110–129.

[33] Tokareva N., *Bent Functions, Results and Applications to Cryptography*, Acad. Press. Elsevier, 2015, 230.

[34] Wada T., "Characteristic bit sequences applicable to constant amplitude orthogonal multicode systems", *IEICE Trans. Fundamentals*, **E83-A**:11 (2000), 2160–2164.

Postquantum Cryptography

Some Remarks on the Security of Isogeny-based Cryptosystems

Sergey Grebnev

QApp, Russia
sg@qapp.tech

Abstract

We review methods proposed for attacking supersingular isogeny-based cryptosystems and apply them to Forsythia – a recently proposed for standardization in Russia key exchange protocol. In particular, we show that the parameters choosen for Forsythia and SIKE are adequate or even too conservative with respect to a novel security model taking into account the success probability of cryptanalytic algorithms.

Keywords: Supersingular isogeny key exchange, parallel collision search, Pollard's rho, SIDH, Forsythia, SIKE.

1 Introduction

Supersingular-isogeny based cryptography, being one of the youngest branches of post-quantum cryptography, has emerged from the early works of Couveignes (1997), Charles-Goren-Lauter (2005), Rostovtsev and Stolbunov (2006) into a practical and quite well-studied field. The paper by De Feo, Jao and Plût [3] was the first to describe SIDH — an efficient and secure key exchange protocol. Basing upon SIDH, a key encapsulation mechanism named SIKE [9] was proposed for the NIST post-quantum standardisation competition and has reached the third round as an alternative candidate.

Following the initiation of the process of search for the post-quantum cryptographic mechanisms prototypes for possible standardisation in Russia, the *Forsythia* protocol [8] was proposed. The protocol is basically a version of SIDH with its own parameters sets, as well as its own starting curve. The main goal of this paper is to study various practical security models proposed for assessment of the parameters of supersingular isogeny-based cryptosystems, and to show that the parameters chosen for Forsythia provide adequate security levels.

We start with a brief description of isogeny-based cryptosystems.

2 Isogeny-based cryptosystems

The protocol is performed by two parties: an *initiator* A and a *responder* B.

Parameters

The basic parameters of the protocol are as follows:

- a prime p, $p = l_A^{e_A} l_B^{e_B} \cdot f - 1$, where l_A, l_B are small primes (e.g., 2 and 3), $(l_A, f) = (l_B, f) = 1$;
- the field $GF(p^2)$;
- a supersingular elliptic curve $E_0(GF(p^2))$ (*starting curve*). We have $\#E_0(GF(p^2)) = (l_A^{e_A} l_B^{e_B} \cdot f)^2$. By construction (see [3]), $E[l_A^{e_A}]$ has $l_A^{e_A - 1}(l_A + 1)$ cyclic subgroups of order $l_A^{e_A}$, every one of them defining an isogeny (being the kernel of an isogeny), the same holds for $E[l_B^{e_B}]$.

Recall that isomorphic curves have the same j-invariant. Since the construction of an isomorphism between two elliptic curves is a simple task, the isogeny problem is actually the problem of finding isogeny between two classes of isomorphic curves, each of them being represented by a j-invariant.

SIDH: the key exchange protocol

Fix the public parameters:

- a prime p, $p = l_A^{e_A} l_B^{e_B} \cdot f - 1$, where l_A, l_B are small primes (e.g., 2 and 3), $(l_A, f) = (l_B, f) = 1$;
- the field $GF(p^2)$;
- a supersingular elliptic curve $E_0(GF(p^2))$ (*starting curve*);
- the bases $\{P_A, Q_A\}$ and $\{P_B, Q_B\}$ which generate $E_0[l_A^{e_A}]$ and $E_0[l_B^{e_B}]$ respectively, that is, $\langle P_A, Q_A \rangle = E_0[l_A^{e_A}]$ and $\langle P_B, Q_B \rangle = E_0[l_B^{e_B}]$.

The party A chooses a random element $n_A \in_R \mathbb{Z}/l_A^{e_A}\mathbb{Z}$ and constructs an isogeny $\varphi_A : E_0 \to E_A$ with the kernel $K_A := \langle P_A + [n_A]Q_A \rangle$. The party A also computes the image $\{\varphi_A(P_B), \varphi_A(Q_B)\}$ and sends these points to B alongside with the elliptic curve E_A (that is, its description).

The party B chooses a random element $n_B \in_R \mathbb{Z}/l_B^{e_B}\mathbb{Z}$ and constructs an isogeny $\varphi_B : E_0 \to E_B$ with the kernel $K_B := \langle P_B + [n_B]Q_B \rangle$. The party

B also computes the image $\{\varphi_B(P_A), \varphi_B(Q_A)\}$ and sends these points to A alongside with the elliptic curve E_B (that is, its description).

The party A, having received from B the tuple $E_B, \varphi_B(P_A), \varphi_B(Q_A)$, constructs an isogeny $\varphi'_A : E_B \to E_{AB}$ with the kernel $\langle \varphi_B(P_A) + [n_A]\varphi_B(Q_A) \rangle$; the party B proceeds in the same manner. The shared key is the j-invariant of the curve

$$E_{AB} = \varphi'_B(\varphi_A(E_0)) = \varphi'_A(\varphi_B(E_0)) = E_0 / \langle P_A + [n_A]Q_A, P_B + [n_B]Q_B \rangle .$$

The practical security of the protocol relies on the hardness of the following problem.

Problem 1. *Computational Supersingular Isogeny – CSSI: let $\phi_1 : E_0 \to E_1$ – an isogeny with the kernel $R_1 + [n_1]S_1$, where n_1 is chosen uniformly at random from the interval $[1, l_1^{e_1}]$. Given E_1 and images $\phi_1(R_2), \phi_1(S_2)$ of the points, find the generator of $\langle R_1 + [n_1]S_1 \rangle$.*

Forsythia: an instantiation of SIDH

Forsythia[1] is a post-quantum key exchange protocol proposed for standardisation in Russia. It is basically an instantiation of SIDH with the following features:

- the scheme is based upon the original SIDH protocol [3];
- the scheme supports ephemeral-only key exchange;
- the scheme has its own starting curve $E_{19}(GF(p)) : y^2 = x^3 - 2^3 \cdot 19x + 2 \cdot 19^2$, which is chosen as proposed in [12];
- the scheme provides three levels of security by specifying three characteristics of the base field:
  - for 80 bits of security: $p_{271} = 2^{132} \cdot 3^{85} \cdot 11 - 1$;
  - for 128 bits of security: $p_{415} = 2^{208} \cdot 3^{129} \cdot 5 - 1$;
  - for 256 bits of security: $p_{754} = 2^{372} \cdot 3^{239} \cdot 7 - 1$.

Remark 1. *We note that the powers of 2 in all the p_i are even, thus allowing us to efficiently employ 4-isogenies in the implementation of the protocol.*

[1] *Forsythia* is a genus of flowering plants in the olive family *Oleaceae*. The name was choosen to be in the line with the names of standardized Russian key exchange protocols *Limonnik* and *Echinacea* which stand for flowering (medicinal) plants.

Remark 2. *We note that Forsythia should only be used in the ephemeral-only key exchange settings, otherwise, an efficient attack from [6] allows an anctive adversary to recover the static secret key in* $O(\log p)$ *sessions. An application of the Fujisaki-Okamoto transformation, allow to convert Forsythia to both a CCA-secure and secure against the attacks from [6]; however, this transformation requires a significant overhead. An authenticated key exchange scheme implementing the Fujisaki-Okamoto transformation was proposed, for example, in [7].*

Having described Forsythia, we proceed with a review of attacks on isogeny-based cryptosystems.

3 Attacks on isogeny-based cryptosystems

We omit the indices A, B for brevity. Thus, a simplified case of the CSSI problem may be stated as follows. Let us have a (secret) l^e-isogeny $\phi : E \to E/G$ for a subgroup $G \subset E$ of the order $l^e \approx p^{1/2}$. The problem is to find the generator of G (or, equivalently, the isogeny ϕ).

Basic method

Since every supersingular elliptic curve $E(GF(p^2))$ has $(l+1)l^{e-1}$ cyclic subgroups of order l^e, the brute-force attack requires $O(l^e)$ or $O(p^{1/2})$ testings.

Meet-in-the-middle

We follow the discussion in [1]. Suppose for simplicity that e is even. We construct a pair of trees such that the leaves of the first define classes of isomorphisms of the curves, $l^{e/2}$-isogenous to E; leaves of the second – classes of isomorphisms of the curves, $l^{e/2}$-isogenous to E/G. Each set contains no more that $(l+1)l^{e/2-1}$ classes.

By brute force testing we detect $l^{e/2}$-isogenies $\phi_1 : E \to E'$ and $\phi_2 : E/G \to E''$ such that there exists an isomorphism $\psi : E' \to E''$. At last, we have l^e-isogeny $\phi = \widehat{\phi_2} \circ \psi \circ \phi_1$.

Required memory is estimated by $O(p^{1/4})$ cells, time — $O(p^{1/4})$ operations.

Parallel collision search

Parallel collision search (van Oorschot–Wiener method) of [14] is efficiently applied to the CSSI problem in [1, 2].

The most efficient methods for collision search of a pseudorandom function f are iterational in the sense that they are based upon the calculation of sequences of the form $x_i = f(x_{i-1})$, $i \in \mathbb{N}$, hence the range of f must be contained within its domain.

The main idea of the method is that every processor generates its own sequence $x_i = f(x_{i-1})$ until a *distinguished point* x_d, which satisfies an easily verifiable condition (fo example, a fixed number of lower bits are zero), is found. x_d is saved to a common memory at the address computed as a function of the distinguished point. If a distinguished point is found twice — we have a collision of f.

Let $S = \{0,1\} \times \{0, \ldots, (l+1)l^{e/2-1} - 1\}$, $E_0 = E$, $E_1 = E/G$. Each pair $(i, y) \in S$ defines a subgroup of elliptic curve E_i.

Example 1. *For $l = 2$ (cf. [1]) the correspondence between the pairs $(i, y) = (i, (b, k)) \in \{0,1\} \times \{0,1,2\} \times \{0, \ldots, l^{e/2-1} - 1\}$ and cyclic subgroups $\langle R_i \rangle \subset E_i$ is given by*

$$R_i = \begin{cases} P_i + [b2^{e/2-1} + k]Q_i, & \text{if } b = 0, 1 \\ [2k]P_i + Q_i, & \text{if } b = 2, \end{cases}$$

where $\langle P_i, Q_i \rangle = E_i[2^{e/2}]$.

Let $h : S \to E_0(GF(p^2)) \cup E_1(GF(p^2))$, $h : (i, y) \mapsto R_i$, and let the iteration function $f : S \to S$ be a function that on an input pair (i, y) computes an isogeny $l^{e/2}$ with the degree $\langle R_i \rangle$, computes the j-invariant $j(E_i/\langle R_i \rangle)$ and maps it to S by some pseudorandom function $g : GF(p^2) \to S$.

There exist an unique "golden" collision for f, which provides a solution of the CSSI problem ([2, §2.3]). The complexity of finding the "golden" collision is estimated in [14] as

$$T = \frac{2.5}{m}\sqrt{|S|^3/w} \cdot t; \tag{1}$$

m is the number of processors, $|S|$ is the size of the scope of iteration function, w is the size of memory availiable, t is the complexity of iteration function.

We have that $|S| \approx p^{1/4}$, and hence the complexity is estimated as

$$O\left(\frac{p^{3/8}}{m\ w^{1/2}}\right) \tag{2}$$

operations of computation of the iteration function (in our case the computation of a $l^{e/2}$-isogeny) (cf. [2]).

Quantum computer

The *claw-finding* algorithm of [13] for given functions $g_1 : X_1 \to Y$, $g_2 : X_2 \to Y$ finds $(x_1, x_2) \in X_1 \times X_2$ such that $g_1(x_1) = g_2(x_2)$.

Suppose that $\#X_1 \approx \#X_2 \approx N$, $\#Y \gg N$, then the time required is $O(N^{2/3})$ operations with $O(N^{2/3})$ memory cells.

In the case of CSSI, X_1 is the set of $l^{e/2}$-isogenies from $E = E_1$; X_2 is the set of $l^{e/2}$-isogenies from $E/G = E_2$, $g_i(\phi) = j(\phi(E_i))$. We have $\#X_1 = \#X_2 \approx p^{1/4}$, hence the time required – $O(p^{1/6})$ (and $O(p^{1/6})$ memory).

Grover's method, applied to the CSSI problem in [10], requires $O(p^{1/4})$ operations, memory requirements are just – $O(1)$.

A recent paper [11] describes a quantum variant of the golden collision search method with the complexity $O(p^{3/14})$ gates and $O(p^{1/14})$ memory.

The analysis of quantum algorithms in [10] implies that in the realistic model of quantum computer (with memory less than 2^{64} cells) Grover's method is more efficient.

4 Costs of attacks and security considerations

Definition 1. *At this stage, we define* quantum *(respectively,* classical*)* security *of the scheme as the length of the key of an (abstract) block cipher, for which the complexity of finding the secret key is equivalent to the complexity of solving the CSSI problem with a quantum (respectively, classical) computer.*

Following [1, 10], we conclude that the best classical method for the CSSI problem is the parallel collision search, the best quantum method — Grover's algorithm. The parameters proposed for Forsythia in [8] were calculated according to these estimations.

Following [4], we suppose that a key sampling of the Kuznyechik block cipher requires $\approx 2^{20}$ quantum gates. The approach of [10] gives a lower estimate for the CSSI problem sampling as 2^{23} quantum gates.

Thus, we have that Grover's algorithm has equivalent complexity estimates for a Forsythia instance with a parameter p and a block cipher with the key length

$$n_Q(p) \approx \log_2 p/2 + 3. \tag{3}$$

The novel golden collision search algorithm has equivalent complexity for a Forsythia instance with a parameter p and a block cipher with the key length

$$n_Q(p) \approx 3/7 \log_2 p + 3. \tag{4}$$

In order to compute the security against classical attack, we denote the number of classical operations required for the sampling of a block cipher key t_1. Suppose that a single iteration function (l^e-isogeny in our case) of the parallel collision search requires t_2 operations, we have that the equivalent key length for Forsythia instance with a parameter p is

$$n_C(p) = 3/8 \log_2 p + \log_2 t_2 - \log_2 t_1 - 1/2 \log_2 w. \tag{5}$$

Thus, in order to achieve n bits of security, we have to take p such that $\min\{n_C(P), n_Q(p)\} \geq n$.

5 The choice of parameters

The prime characteristic of the base field p is chosen as $p = 2^{e_2}3^{e_3}f - 1$ such that the factors $2, 3$ are balanced: $e_2 \approx e_3 \cdot \log_2 3$, and the factor f is a small integer. Also we require that $\left(\frac{-19}{p}\right) = -1$ (this is necessary for the starting curve E_{19} to be supersingular, cf. [12]).

p	Formula	Classical security	Quantum security (Grover)	Quantum security (Golden collision)
p_{271}	$2^{132} \cdot 3^{85} \cdot 11 - 1$	82	130	119
p_{415}	$2^{208} \cdot 3^{129} \cdot 5 - 1$	135	202	180
p_{754}	$2^{372} \cdot 3^{239} \cdot 7 - 1$	262	371	326

Remark 3. *Classical security was computed by (5) for a hypothetical supercomputer with memory availiable $w \approx 2^{64}$ cells (for comparison, the supercomputer Fugaku, which heads the current Top500 list [15], has $\approx 2^{23}$ cores and $\approx 2^{49}$ memory cells) with $t_2 = 2^{22}$, $t_1 = 2^{10}$.*

Remark 4. *We use the best estimates for quantum golden collision security calculated with the software from [11]. We suppose that the quantum circuits are limited by 2^{64} gates.*

6 Security with respect to success probability

Consider now the security definition from a recent talk by S. Galbraith [5].

Definition 2. *Let X be a computational problem, with instances x produced by an algorithm $Gen(1^\lambda)$. Let ε_0 be some fixed upper bound on success probabilities of interest (possibly a function of λ). Then X has λ-bit security level if, for every adversary A, if $A(x)$ runs in time t and succeeds with probability $\varepsilon < \varepsilon_0$ we have $t/\varepsilon \geq 2^\lambda$.*

The probability $P(t)$ of success of the parallel collision search method over a search space S with $\#S = n$ after t iterations by all processors with z memory cells is estimated (cf. [14, Lemma 2]) as

$$P(t) = \begin{cases} 1 - \exp(-t^2/(2n)), \text{ if } t < z, \\ 1 - (1 - \frac{z}{n})^{t-z} \exp(-z^2/(2n)), \text{ otherwise.} \end{cases} \tag{6}$$

We assess the bit security of various Forsythia instances in the sense of Definition 2 in the following manner. Consider now a hypothetical computer with 2^{64} memory cells. Let us suppose it is capable of running $T = 2^{80}$ operations (that is, an iteration function computations) to break Forsythia with p_{415}, p_{754} and $T = 2^{64}$ operations[2] to break Forsythia with p_{271}. The computer runs the parallel collision search method. In order to reach the acclaimed bit security, we have the upper bound on success probability ε_0, respectively, equal to 2^{-48}, 2^{-176} and 2^{-16}. Thus, in order to compute the required base prime p for Forsythia, we perform the following.

1. Substitute $n = p^{1/4}$ into equation (6) and rewrite it for a new variable $\theta = t/\sqrt{n}$:

$$P(\theta) = \begin{cases} 1 - \exp(-\theta^2/(2)), \text{ if } \theta < z/\sqrt{n}, \\ 1 - (1 - \frac{z}{n})^{\theta\sqrt{n}-z} \exp(-z^2/(2n)), \text{ otherwise.} \end{cases} \tag{7}$$

2. Solve (7) for θ with $P = \varepsilon_0$.

3. Compute the bit-size l of the p required to obtain the claimed security level as

$$l = 4(\log_2 T - log_2(\theta)).$$

After computation, we have:

Claimed security level	Forsythia prime (bit-size)	Calculated prime (bit-size)
80 bits	**271**	286
128 bits	415	414
256 bits	754	670

[2]Setting $T = 2^{80}$ in this case implies success probability equal to 1 and gives no useful information.

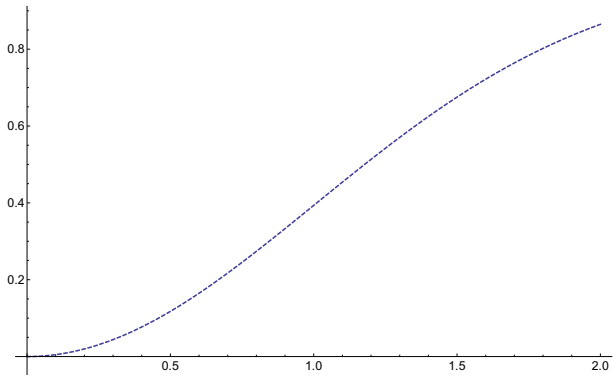

Figure 1: The plot of $P(\theta)$ for p_{271}.

Thus, we note that p_{415} and p_{754} still satisfy their claimed security levels, while p_{271} (shown in **bold**) is just slightly below the security margin.

Repeating the calculations for SIKE parameters [9], with the memory storage 2^{80} availiable (as in [1]), we have:

Claimed NIST security level	SIKE prime (bit-size)	Calculated prime (bit-size)
I	434	414
III	610	541
V	751	670

Thus, we see that the SIKE primes are chosen somewhat conservatively in the new model as well. Note that we have only considered SIKE primes for security levels I, III and V, since these security levels are defined via the complexity of analysis of a block cipher.

7 Conclusion

We have studied the security of the Forsythia isogeny-based key exchange protocol in various models. We have shown that the parameters chosen for Forsythia satisfy their claimed security levels in the "traditional" security model and, except for p_{271}, which is just slightly below the security margin, in the novel security model of [5] against an adversary which may execute 2^{80} (or 2^{64}) operations. We have also shown that the SIKE parameters for NIST security levels I, III and V are chosen somewhat conservatively even with respect to the new model.

References

[1] Adj G., Cervantes-Vázquez D., Chi-Domínguez J.-J., Menezes A., Rodríguez-Henríquez F., "On the cost of computing isogenies between supersingular elliptic curves", *Cryptology ePrint Archive*, **2018/313**, 2018.

[2] Costello C., Longa P., Naehrig M., Renes J., Virdia F., "Improved Classical Cryptanalysis of SIKE in Practice", *Cryptology ePrint Archive*, **2019/298**, 2019.

[3] De Feo L., Jao D., Plût J., "Towards Quantum-Resistant Cryptosystems From Supersingular Elliptic Curve Isogenies", *J. Mathematical Cryptology*, **8(3)** (2014), 209-–247.

[4] Denisenko D., Marshalko G., Nikitenkova M., Rudskoy V., Shishkin V., "Estimation of Grover's algorithm implementation for searching GOST R 34.12-2015 block cipher keys", *Jour. Exp. Theor. Phys.*, **155**:4 (2019), 645, In Russian.

[5] Galbraith S., "Security levels in cryptography", 2020, https://www.math.auckland.ac.nz/~sgal018/ACISP.pdf.

[6] Galbraith S., Petit C., Shani B., Ti Y.B., "On the Security of Supersingular Isogeny Cryptosystems", *Cryptology ePrint Archive*, **2016/859**, 2016.

[7] Grebnev S., "Limonnitsa: making Limonnik-3 post-quantum", *Matem. Vopr. Kriptografii*, **11**:2 (2020), 25–42.

[8] Grebnev S., Klyucharev P., Koreneva A., Koshelev D., Taraskin O., Tulebaev A., "Forsythia: a supersingular isogeny-based key exchange protocol", *Working draft*, 2021, `https://www.ruscrypto.ru/resource/archive/rc2021/files/02\_grebnev_klucharev_koreneva_koshelev_taraskin_tulebayev.pdf`, In Russian.

[9] Jao D., Azarderakhsh R., Campagna M., Costello C., De Feo L., Hess B., Jalali A., Koziel B., LaMacchia B., Longa P., Naehrig M., Renes J., Soukharev V., Urbanik D., "Supersingular Isogeny Key Encapsulation", 2017, https://sike.org/#nist-submission.

[10] Jaques S., Schanck J.M., "Quantum cryptanalysis in the RAM model: Claw-finding attacks on SIKE", *Cryptology ePrint Archive*, **2019/103**, 2019.

[11] Jaques S., Schrottenloher A., "Low-gate Quantum Golden Collision Finding.", *SAC 2020 – Selected Areas in Cryptography, Oct 2020, Online, Canada*, 2020, https://hal.inria.fr/hal-03046039/document.

[12] Koshelev D., "Starting supersingular elliptic curve for isogeny-based cryptography", 2020, https://www.researchgate.net/profile/Dimitri_Koshelev, In Russian.

[13] Seiichiro T., "Claw Finding Algorithms Using Quantum Walk", 2008, http://arxiv.org/abs/0708.2584.

[14] van Oorschot P.C., Wiener M.J., "Parallel Collision Search with Cryptanalytic Applications.", *J. Cryptology*, **12** (1999), 1-–28.

[15] "Supercomputer Fugaku", 2020, https://www.top500.org/system/179807/.

The Hadamard Square of Concatenated Linear Codes

Ivan Chizhov[1,2,3] and Alexandra Davletshina[4]

[1]Lomonosov Moscow State University, Russia
[2]Federal Research Center "Informatics and Control" of Russian Academy of Science, Russia
[3]JSC "NPK Kryptonite", Russia
[4]JSC "InfoTeCS", Russia
ichizhov@cs.msu.ru, sdav94@rambler.ru

Abstract

The paper is devoted to the Hadamard square of concatenated linear codes. Such codes consist of codewords that are obtained by concatenation part of the codewords from other codes. It is proved that if the sum of Hadamard squares' dimensions of the codes used in the concatenation is slightly less than the dimension of the entire space, then the Hadamard square of the concatenated code is equal to the Cartesian product of the Hadamard square of code-components.

It means that the cryptanalysis for many code-based post-quantum cryptographic mechanisms built on concatenated codes is equivalent to the cryptanalysis of these mechanisms built on code-components. So using the concatenation of codes from different classes instead of one class of codes, generally speaking, does not increase the cryptographic strength of the mechanisms.

Keywords: concatenated linear codes, Hadamard square, Hadamard product, Schur product, component-wise product, McEliece public-key cryptosystem, post-quantum cryptography

1 Introduction

The Hadamard (Schur) product or the coordinate-wise product of linear codes has been studied for a long time. In the beginning, it was used to construct algebraic decoders correcting errors for some linear codes [18]. Recently, it is increasingly used in cryptography. Many constructions of secret sharing schemes and cryptographic protocols for secure multi-party computation [4] use the Hadamard product of linear codes. Attacks on post-quantum code-based cryptographic mechanisms are one of the main applications of this operation over linear codes. So, worth noting the attack [2] on the McEliece cryptosystem based on Reed-Muller binary codes, or the attack [22] on the same cryptosystem, but based on Reed-Solomon subcodes. Numerous examples of the Hadamard product application for constructing attacks on code-based cryptosystem given in works [7, 8, 9, 10, 17]. For the first time, the

efficient algorithm was constructed in [13] that distinguishes Goppa codes from random binary codes using this operation.

For practical applications be essential to describe the Hadamard square of the linear code and establish its properties as a linear code. For example, in [3] it was proved that the Hadamard square of the linear code fills the entire space with a probability close to one. This property is often used to construct attacks on post-quantum public-key cryptosystems; for example, see works [1, 7, 10, 17]. Some cryptographic mechanisms are based on linear codes, the Hadamard square of which is not equal to the entire space. Then the algebraic or combinatorial structure of the Hadamard square of the linear code becomes important.

Recently, several attacks [5, 6, 11, 17] have been constructed on post-quantum cryptographic mechanisms based on the concatenation of linear codes from different classes. Such linear codes consist of codewords which are obtained by combining part of the codewords from other codes. Moreover, for these attacks to work correctly, it is required that the Hadamard square of the combined code is equal to the Cartesian product of Hadamard squares of the codes used in the combination. The researchers noted that this property is fulfilled almost always in the experiments, but there is no theoretical for this fact proved was provided.

In this paper, the theoretical gap is eliminated. And it is proved that if the sum of the Hadamard squares' dimensions of the codes used in the concatenation is slightly less than the dimension of the entire space, then the Hadamard square of the concatenated code is equal to the Cartesian product of Hadamard squares of code-components.

2 The main result

Let V_q^n be the linear space of all vectors of length n over $GF(q)$. *Block linear* $[n,k]_q$*-code* over $GF(q)$ or just *code* is a k-dimensional linear subspace $\mathcal{C}$ of V_q^n. In this case, n is called the *length* of the code, and k is called the *dimension* of code. When the dimension of the code $\mathcal{C} \subseteq V_q^n$ is not essential to us, it will be called the $[n]_q$-code $\mathcal{C}$. Vectors $c \in \mathcal{C}$ are called *codewords* of the code $\mathcal{C}$.

We say that the $[n]_q$-code $\mathcal{C}$ is generated by the $(k \times n)$-matrix G with elements from $GF(q)$ if the linear combination of the rows of the matrix G over $GF(q)$ coincides with $\mathcal{C}$. This fact we write as $\mathcal{C} = \langle G \rangle$. Moreover, if matrix G has the minimum rank among all matrices generating code $\mathcal{C}$, then it is called the *generator* matrix of the code $\mathcal{C}$.

The vector $h = (h_1, \ldots, h_n) \in V_q^n$ is called *parity check* of the code $\mathcal{C}$, if for any vector $c = (c_1, \ldots, c_n) \in \mathcal{C}$ holds equality

$$h_1 \cdot c_1 + \ldots + h_n \cdot c_n = 0,$$

here all operations are performed in the field $GF(q)$. It is clear that the set of all parity checks of code $\mathcal{C}$ is a linear subspace of V_q^n, i.e. the linear code. This code is called the *dual* code to code $\mathcal{C}$. We denote the code dual to $\mathcal{C}$ as $\mathcal{C}^\perp$.

The generator matrix H of code $\mathcal{C}^\perp$ is called the *parity-check matrix* of code $\mathcal{C}$. Note that from the definition of the parity-check matrix H of code $\mathcal{C}$, it follows that for any $c \in \mathcal{C}$ holds the equalities

$$Hc^T = 0, \quad cH^T = 0.$$

The minimum distance(see [16]) of the linear code $\mathcal{C}$ is called the number

$$d_{\mathcal{C}} = \min_{c \in \mathcal{C},\ c \neq 0} \mathrm{wt}(c),$$

here $\mathrm{wt}(c)$ is the Hamming weight (the number of nonzero coordinates) of the vector c. The minimum distance of code $\mathcal{C}^\perp$, which is dual to code $\mathcal{C}$, is denoted as $d_{\mathcal{C}}^\perp$.

The Cartesian product of vectors $c = (c_1, \ldots, c_n) \in V_q^n$ and $b = (b_1, \ldots, b_m) \in V_q^m$ is called vector

$$c \times b = (c_1, \ldots, c_n, b_1, \ldots, b_m) \in V_q^{m+n}.$$

Accordingly, the *Cartesian product* $[n]_q$-code $\mathcal{C}$ and $[m]_q$-code $\mathcal{B}$ is called $[n+m]_q$-code $\mathcal{C} \times \mathcal{B}$ consisting of vectors

$$\mathcal{C} \times \mathcal{B} = \{c \times b | c \in \mathcal{C},\ b \in \mathcal{B}\}.$$

The concatenation $cat(\mathcal{C}_1, \ldots, \mathcal{C}_u)$ of codes $\mathcal{C}_1, \ldots, \mathcal{C}_u$ is called the set of codes $\mathcal{C}$, which are generated by a matrix of the form

$$(G_1 \| \ldots \| G_u),$$

here $\|$ is the concatenation of matrix columns, and the $(k \times n_i)$-matrix G_i generates the code $\mathcal{C}_i$, $i = 1, 2, \ldots, u$. It is clear that $\mathcal{C} \in cat(\mathcal{C}_1, \ldots, \mathcal{C}_u)$ is $[n_1 + \ldots + n_u]_q$-code.

Also, for any code $\mathcal{C} \in cat(\mathcal{C}_1, \ldots, \mathcal{C}_u)$, the following inclusion is true

$$\mathcal{C} \subseteq \mathcal{C}_1 \times \ldots \times \mathcal{C}_u.$$

Hadamard product of two vectors $c, b \in V_q^n$ is called the vector $c \circ b$ obtained as a result of the component-wise product of coordinates of these vectors:

$$c \circ b = (c_1, \ldots, c_n) \circ (b_1, \ldots, b_n) = (c_1 b_1, \ldots, c_n b_n).$$

Definition 1. *Let $\mathcal{C}$ and $\mathcal{B}$ are $[n]_q$-codes. Then* Hadamard product (Schur product, component-wise product) *$\mathcal{C} \circ \mathcal{B}$ of codes $\mathcal{C}$ and $\mathcal{B}$ will be called the $[n]_q$-code, consisting of the linear span of the following vectors $\{c \circ b | c \in \mathcal{C},\ b \in \mathcal{B}\}$. If $\mathcal{C} = \mathcal{B}$, then code $\mathcal{C} \circ \mathcal{C} = \mathcal{C}^2$ is called* Hadamard square *of code $\mathcal{C}$.*

For the Hadamard square of codes that are the concatenation of other codes, the following proposition is true.

Proposition 1. *Let $\mathcal{C} \in cat(\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_u)$ for some codes $\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_u$. Then the following inclusion is true*

$$\mathcal{C}^2 \subseteq \mathcal{C}_0^2 \times \mathcal{C}_1^2 \times \ldots \times \mathcal{C}_u^2. \tag{1}$$

We will be interested in the following problem. Under what condition the inclusion (1) turns into equality. The paper's main result is the following 1.

Theorem 1. *Let u be a positive integer, and for each $i = 0, 1, \ldots, u$ the code $\mathcal{C}_i$ be a $[n_i]_q$-code. Let also $[N, k]_q$-code $\mathcal{C} \in cat(\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_u)$.*

If $d_{\mathcal{C}}^{\perp} \neq 2$, $k \geq 4$, $N \leq \frac{k(k+1)}{2}$, $N \cdot \log_q(2 - q^{-1}) \leq \frac{k(k-3)}{2}$ and

$$N - \log_q \frac{3k+4}{4} \geq \dim \mathcal{C}_0^2 + \dim \mathcal{C}_1^2 + \ldots + \dim \mathcal{C}_u^2,$$

then we have

$$\mathcal{C}^2 = \mathcal{C}_0^2 \times \mathcal{C}_1^2 \times \ldots \times \mathcal{C}_u^2. \tag{2}$$

3 Hadamard square and quadratic forms

It turns out to be a convenient interpretation of the Hadamard square of the linear code with a point of view of quadratic forms over $GF(q)$. Such an approach allowed the authors of [3] to establish the behavior of the dimension of Hadamard square of a random linear code.

Definition 2. *A quadratic form over $GF(q)$ is called homogeneous quadratic polynomial over this field*

$$q(x_1, \ldots, x_k) = \sum_{1 \leq i < j \leq k} a_{i,j} x_i x_j + \sum_{i=1}^{k} b_i x_i^2,$$

here $a_{i,j} \in GF(q), 1 \leq i < j \leq k$, u $b_i \in GF(q), 1 \leq i \leq k$.

Let denotes by $\mathcal{Q}_k(q)$ the set of all quadratic forms over $GF(q)$ in k variables. Consider a $(k \times n)$-matrix G, let $g_i \in V_q^k$ be the column of the matrix G with index i. Define a mapping $\ell_G : \mathcal{Q}_k(q) \to V_q^n$ in the following way:

$$\ell_G(f) = \big(f(g_1), \ldots, f(g_n)\big).$$

In this case, the Hadamard square of the linear $[n,k]_q$-code $\mathcal{C}$ generated by the matrix G is the image of the linear operator ℓ_G (see, for example, [3, 19]):

$$\mathcal{C}^2 = \operatorname{Im} \ell_G. \tag{3}$$

The following proposition attends directly from proposition 1.

Proposition 2. *Let $\mathcal{C} \in cat(\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_u)$ for some codes $\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_u$. Then we have*

$$\dim \mathcal{C}^2 = \dim \operatorname{Im} \ell_{(G_0 \| G_1 \| \ldots \| G_u)} \leq \sum_{i=0}^{u} \dim \operatorname{Im} \ell_{G_i}, \tag{4}$$

where $(G_0 \| G_1 \| \ldots \| G_u)$ is generator matrix of code $\mathcal{C}$.

Moreover, equality in (4) is achieved if and only if

$$\mathcal{C}^2 = \mathcal{C}_0^2 \times \mathcal{C}_1^2 \times \ldots \times \mathcal{C}_u^2.$$

Let $\ker \ell_G$ be a kernel of linear operator ℓ_G. Since $\dim \mathcal{Q}_k(q) = \frac{k(k+1)}{2}$, then the equality

$$\dim \operatorname{Im} \ell_G = \frac{k(k+1)}{2} - \dim \ker \ell_G$$

holds.

So the following proposition is true.

Proposition 3. *Let $\mathcal{C} \in cat(\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_u)$ for some codes $\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_u$. Then we have*

$$\dim \ker \ell_{(G_0 \| G_1 \| \ldots \| G_u)} \geq \frac{k(k+1)}{2} - \sum_{i=0}^{u} \dim \mathcal{C}_i^2, \tag{5}$$

where $(G_0 \| G_1 \| \ldots \| G_u)$ is generator matrix of code $\mathcal{C}$.

Moreover, equality in (5) is achieved if and only if

$$\mathcal{C}^2 = \mathcal{C}_0^2 \times \mathcal{C}_1^2 \times \ldots \times \mathcal{C}_u^2.$$

Proof. The proof follows from the equality

$$\dim \ker \ell_{(G_0 \| G_1 \| \ldots \| G_u)} = \frac{k(k+1)}{2} - \dim \operatorname{Im} \ell_{(G_0 \| G_1 \| \ldots \| G_u)},$$

and from proposition 2 given (3). □

4 The proof of main theorem

This section is devoted to the proof of the main results of the paper. Let us first establish the truth of the most general theorem.

Theorem 2. *Let X_i, $i = 0, \ldots, u$, be $(k \times n_i)$-matrices over $GF(q)$. Matrix X_i, $i = 0, \ldots, u$, generates linear code $\mathcal{C}_i$. Denote by $N = n_0 + n_1 + \ldots + n_u$. Let $\mathcal{C}$ be $[N, k]_q$-code over $GF(q)$ generated by the matrix $X = (X_0 \| X_1 \| \ldots \| X_u)$. Let us require that the matrix X does not contain identical columns.*

If $k \geq 4$, $N \leq \frac{k(k+1)}{2}$, $N \cdot \log_q(2 - q^{-1}) \leq \frac{k(k-3)}{2}$ and

$$N - \log_q \frac{3k+4}{4} \geq \dim \mathcal{C}_0^2 + \mathcal{C}_1^2 + \ldots + \dim \mathcal{C}_u^2,$$

then we have

$$\mathcal{C}^2 = \mathcal{C}_0^2 \times \mathcal{C}_1^2 \times \ldots \times \mathcal{C}_u^2.$$

Proof. At first, we prove the useful technical lemma.

Lemma 1. *Consider a discrete random variable ξ with a finite number of values $\{a_1, \ldots, a_s\}$. Let p_i be the probability of occurrence of the value a_i. We will assume that a_1 is the minimum possible value of ξ. Let us denote by $\mathcal{M}\xi$ the mathematical expectation of the random variable ξ. If $\mathcal{M}\xi \leq a_1$, then for any i we have either $p_i = 0$, or $a_i = a_1$.*

Proof. Indeed, by definition

$$a_1 \geq \mathcal{M}\xi = \sum_{i=1}^{s} a_i p_i \Leftrightarrow a_1 \sum_{i=1}^{s} p_i \geq \sum_{i=1}^{s} a_i p_i \Leftrightarrow 0 \geq \sum_{i=1}^{s} (a_i - a_1) p_i.$$

Now $p_i \geq 0$ and $a_i - a_1 \geq 0$ for $i = 2, \ldots, s$, since a_1 is minimum value of random variable ξ. Therefore $\sum_{i=1}^{s} (a_i - a_1) p_i = 0$. But it is only possible if for each $i = 2, \ldots, s$, either $p_i = 0$ or $a_i = a_1$. □

Consider $\ker \ell_X$.

Let be given a uniform distribution on the set of $(k \times N)$-matrices $X = (X_0 \| \ldots \| X_u)$, such that the matrix X has no zero columns and repeated columns. Then $\ker \ell_X$ will be a random variable defined on the set of random matrices X. According to proposition 3 holds the following inequality

$$\dim \ker \ell_X \geq \frac{k(k+1)}{2} - \sum_{i=0}^{u} \dim \mathcal{C}_i^2.$$

This means that if we prove that

$$\mathcal{M} \dim \ker \ell_X \leq \frac{k(k+1)}{2} - \sum_{i=0}^{u} \dim \mathcal{C}_i^2,$$

then from Lemma 1, it will follow that the random variable $\dim \ker \ell_X$ with nonzero probability can take only the value

$$\dim \ker \ell_X = \frac{k(k+1)}{2} - \sum_{i=0}^{u} \dim \mathcal{C}_i^2.$$

Therefore, according to proposition 3, the truth of the theorem will follow from this.

Thus, it is necessary to estimate the mathematical expectation of a random variable $\dim \ker \ell_X$. Now $|\ker \ell_X| = q^{\dim \ker \ell_X}$, therefore, we will estimate the mathematical expectation of the cardinality of $\ker \ell_X$. By definition $f \in \ker \ell_X$, if and only if $f(X_0) = f(X_1) = \ldots = f(X_u) = 0$.

Let I_f be a random variable that takes the value one if $f(X_0) = f(X_1) = \ldots = f(X_u) = 0$, and 0 in other cases. Then

$$|\ker \ell_X| = \sum_{f \in \mathcal{Q}_k(q)} I_f.$$

Since the mathematical expectation is linear, the following equality is true

$$\mathcal{M}|\ker \ell_X| = \sum_{f \in \mathcal{Q}_k(q)} \mathcal{M} I_f.$$

Notice that

$$\mathcal{M} I_f = 0 \cdot \Pr\{I_f = 0\} + 1 \cdot \Pr\{I_f = 1\}.$$

Therefore

$$\mathcal{M}|\ker \ell_X| = \sum_{f \in \mathcal{Q}_k(q)} \Pr\{I_f = 1\}.$$

Let for all $f \in \mathcal{Q}_k(q)$ holds the inequality

$$\Pr\{I_f = 1\} \leq q^{-\sum_{i=0}^{u} \dim \mathcal{C}_i^2}, \tag{6}$$

then

$$\mathcal{M}|\ker \ell_X| \leq |\mathcal{Q}_k(q)| \cdot q^{-\sum_{i=0}^{u} \dim \mathcal{C}_i^2}.$$

However, then, taking into account $|\ker \ell_X| = q^{\dim \ker \ell_X}$, for $\dim \ker \ell_X$, we get

$$\mathcal{M} \dim \ker \ell_X \leq \dim \mathcal{Q}_k(q) - \sum_{i=0}^{u} \dim \mathcal{C}_i^2.$$

Since $\dim \mathcal{Q}_k(q) = \frac{k(k+1)}{2}$, we get

$$\mathcal{M} \dim \ker \ell_X \leq \frac{k(k+1)}{2} - \sum_{i=0}^{u} \dim \mathcal{C}_i^2.$$

So, to prove the theorem, it is necessary to establish that the inequality (6) holds for any quadratic form f.

Consider the quadratic form f which takes the value 0 on the set of values X, $|X| = N$. Let its weight be w. Then there are $\binom{q^k-w}{N}$ options for choosing from set of arguments of subset $Y = X_0 \cup X_1 \cup \ldots \cup X_u$ of cardinality $N = n_0 + n_1 + \ldots + n_u$, on which form f takes 0. Then the fraction of such subsets Y among all possible subsets of cardinality N will be equal to $\binom{q^k-w}{N} / \binom{q^k}{N}$. This means that

$$\Pr\{I_f = 1 | wt(f) = w\} = \frac{\binom{q^k-w}{N}}{\binom{q^k}{N}}.$$

Then by the law of total probability

$$P = \Pr\{I_f = 1\} = \sum_{w=0}^{q^k} \Pr\{wt(f) = w\} \Pr\{I_f = 1 | wt(f) = w\}.$$

Suppose that f is chosen randomly and with equal probability from $\mathcal{Q}_k(q)$, then the probability $\Pr\{wt(f) = w\}$ can be calculated by the formula

$$\Pr\{wt(f) = w\} = \frac{Q_w}{q^{\dim \mathcal{Q}_k(q)}} = \frac{Q_w}{q^{k(k+1)/2}},$$

where Q_w is number of quadratic forms of weight w.

Then we get

$$P = \sum_{w=0}^{q^k} \frac{Q_w}{q^{k(k+1)/2}} \frac{\binom{q^k-w}{N}}{\binom{q^k}{N}} = \frac{1}{q^{k(k+1)/2}} \cdot \sum_{w=0}^{q^k} Q_w \frac{\binom{q^k-w}{N}}{\binom{q^k}{N}}. \tag{7}$$

Let

$$Q = \sum_{w=0}^{q^k} Q_w \frac{\binom{q^k-w}{N}}{\binom{q^k}{N}}.$$

Further, $Q_w \neq 0$ only for $w = 0, q^k - q^{k-1}, q^k - q^{k-1} - \tau q^{k-1-h}(q-1)$ where $h = 1, \ldots, \lfloor k/2 \rfloor$ and $\tau = 1, -1$ (see [14, 15, 21]).

So, the following fractions need to be estimated

$$\frac{\binom{q^{k-1}}{N}}{\binom{q^k}{N}}, \quad \frac{\binom{q^{k-1}+q^{k-1-h}(q-1)}{N}}{\binom{q^k}{N}}, \quad \frac{\binom{q^{k-1}-q^{k-1-h}(q-1)}{N}}{\binom{q^k}{N}}.$$

Lemma 2. *If* $0 < a < q$, $n > 0$ *and* $N > 0$, *then we have*

$$\frac{\binom{a\cdot q^{k-1}}{N}}{\binom{q^k}{N}} \le a^N q^{-N}.$$

Proof. Consider equalities

$$\frac{\binom{a\cdot q^{k-1}}{N}}{\binom{q^k}{N}} = \frac{(a\cdot q^{k-1})!(q^k-N)!}{(a\cdot q^{k-1}-N)!q^k!} = \prod_{i=1}^{N} \frac{a\cdot q^{k-1}-N+i}{q^k-N+i}.$$

Further,

$$\frac{a\cdot q^{k-1}-N+i}{q^k-N+i} = \frac{a\cdot q^{k-1}-q^k+q^k-N+i}{q^k-N+i} = 1-\frac{q^{k-1}(q-a)}{q^k-N+i}. \qquad (8)$$

Since $a < q$, the fraction $q^{k-1}(q-a)/(q^k-N+i)$ is not negative, so the smaller it is, the (8) is more. Thus, the maximum of expression (8) is reached at $i = N$.

For $1 \le i \le N$ we have

$$1-\frac{q^{k-1}(q-a)}{q^k-N-n+i} \le 1-\frac{q^{k-1}(q-a)}{q^k} = \frac{a}{q}.$$

Therefore

$$\frac{\binom{a\cdot q^{k-1}}{N}}{\binom{q^k}{N}} \le a^N q^{-N}.$$

□

Let a takes one of the values $1, 1\pm q^{-h}(q-1)$, where $1 \le h \le \lfloor k/2 \rfloor$. Since $q \ge 1$, then $1+q^{-h}(q-1) > 0$. Further, for $h \ge 1$, the inequality $1-q^{-h}(q-1) \ge 1-q^{-1}(q-1) = q^{-1} > 0$ holds.

Hence, according to Lemma 2, we have

$$\frac{\binom{q^{k-1}}{N}}{\binom{q^k}{N}} \le q^{-N}, \quad \frac{\binom{q^{k-1}\pm q^{k-1-h}(q-1)}{N}}{\binom{q^k}{N}} \le q^{-N}(1\pm q^{-h}(q-1))^N.$$

Thus,

$$Q \le 1+q^{-N}Q^0+q^{-N}\sum_{h=1}^{\lfloor k/2\rfloor}\left[Q_h^-(1+q^{-h}(q-1))^N+Q_h^+(1-q^{-h}(q-1))^N\right],$$

where $Q^0 = Q_{q^k-q^{k-1}}$, $Q_h^- = Q_{q^k-q^{k-1}-q^{k-1-h}(q-1)}$ and $Q_h^+ = Q_{q^k-q^{k-1}+q^{k-1-h}(q-1)}$. Since $1+q^{-h}(q-1) \ge 1-q^{-h}(q-1)$ for $h \ge 1$, then

$$Q \le 1+q^{-N}Q^0+q^{-N}\sum_{h=1}^{\lfloor k/2\rfloor}\left[Q_h^-+Q_h^+\right](1+q^{-h}(q-1))^N. \qquad (9)$$

Let us estimate Q^0 and $Q_h^- + Q_h^+$ for $1 \le h \le \lfloor k/2 \rfloor$.

According to works [14, 15, 21], holds

$$Q_h^{\pm} = \frac{1}{2} q^{h^2} (q^h \mp 1) \frac{\prod_{i=k-2h+1}^{k} (q^i - 1)}{\prod_{i=1}^{h} (q^{2i} - 1)}.$$

First, note that $Q_h^+ \le Q_h^-$, so we only estimate Q_h^-.

Choose any ε, $k^{-1} \le \varepsilon \le \frac{1}{4}$. Then $1 \le \varepsilon k \le k/4$. Let $h \le \varepsilon \cdot k$. In this case $2h < k - 2h + 1$. Then we get

$$Q_h^- \le \frac{1}{2} q^{h^2} \frac{\prod_{i=k-2h+1}^{k} q^i}{(q^h - 1) \prod_{i=1}^{h-1} (q^{2i} - 1)} = \frac{1}{2} \frac{q^{h^2 + 2hk - h(2h-1)}}{(q^h - 1) \prod_{i=1}^{h-1} (q^{2i} - 1)}.$$

Now we use the inequality $q^x - 1 \ge q^{x-1}$, which is valid for any $x \ge 1$ and $q \ge 2$.

$$(q^h - 1) \cdot \prod_{i=1}^{h-1} (q^{2i} - 1) \ge q^{h-1} \cdot \prod_{i=1}^{h-1} q^{2i-1} = q^{h-1+2\sum_{i=1}^{h-1} i - \sum_{i=1}^{h-1} 1} = q^{h(h-1)}.$$

Then for $2 \le h \le \varepsilon \cdot k$ we get

$$Q_h^- \le \frac{1}{2} \cdot q^{h^2 + 2hk - h(2h-1) - h(h-1)} = \frac{1}{2} \cdot q^{2h(k+1-h)}.$$

Let us find the extremum of $\phi(h) = 2h(k + 1 - h)$. For $k + 1 \ge 2h$ derivative $\phi'(h) = 2(k + 1 - h) - 2h = 2(k + 1 - 2h)$ is not negative, therefore $\phi(h)$ does not decrease on the interval $[1, (k + 1)/2]$.

Hence, for $2 \le h \le \varepsilon \cdot k \le k/4 < (k + 1)/2$ we get $2h(k + 1 - h) \le 2\varepsilon k(k + 1 - \varepsilon k) = 2\varepsilon(1 - \varepsilon)k^2 + 2\varepsilon k$.

If $h = 1$, then

$$Q_1^- = \frac{1}{2} q(q+1) \frac{\prod_{i=k-1}^{k} (q^i - 1)}{\prod_{i=1}^{1} (q^{2i} - 1)} = \frac{1}{2} q(q+1) \frac{(q^k - 1)(q^{k-1} - 1)}{q^2 - 1} \le \frac{1}{2} \cdot q^{2k}.$$

Since $1 \le \varepsilon k$ and $\varepsilon < 1$, then $2k \le 2\varepsilon(1 - \varepsilon)k^2 + 2\varepsilon k$.

Then for $k^{-1} \le \varepsilon \le \frac{1}{4}$ and $1 \le h \le \varepsilon k$ we get

$$Q_h^{\pm} \le \frac{1}{2} \cdot q^{2\varepsilon(1-\varepsilon)k^2 + 2\varepsilon k}.$$

Thus for $1 \le h \le \varepsilon k$ we get

$$Q_h^- + Q_h^+ \le q^{2\varepsilon(1-\varepsilon)k^2 + 2\varepsilon k}.$$

And for the remaining Q_h^-, $h > \varepsilon k$, and Q^0, we have the trivial inequalities

$$Q_h^- + Q_h^+ \le q^{k(k+1)/2}, \quad Q^0 \le q^{k(k+1)/2}.$$

Then from (9) for every $k^{-1} \le \varepsilon \le 4^{-1}$ implies

$$Q \le 1+(1+(1/2-\varepsilon)k)q^{-N}q^{k(k+1)/2}+q^{-N}q^{2\varepsilon(1-\varepsilon)k^2+2\varepsilon k}\sum_{h\le\varepsilon k}(1+q^{-h}(q-1))^N.$$

Notice that $1 + q^{-h}(q-1) \le 2 - q^{-1} = \alpha_q$. Then

$$Q \le 1 + (1 + (1/2 - \varepsilon)k)q^{-N}q^{k(k+1)/2} + \varepsilon \cdot k \cdot q^{-N}q^{2\varepsilon(1-\varepsilon)k^2+2\varepsilon k}\alpha_q^N.$$

Further, if $k \ge 4$, then

$$1 + \left(\frac{1}{2} - \varepsilon\right) k = 1 - \varepsilon k + \frac{k}{2} \le \frac{k}{2}.$$

. Thus,

$$Q \le 1 + \frac{k}{2}q^{-N}q^{k(k+1)/2} + \frac{k}{4}q^{-N(1-\log_q \alpha_q)}q^{2\varepsilon(1-\varepsilon)k^2+2\varepsilon k}. \tag{10}$$

Let us choose ε so that

$$\frac{k(k+1)}{2} - N \ge -N(1 - \log_q \alpha_q) + 2\varepsilon(1-\varepsilon)k^2 + 2\varepsilon k. \tag{11}$$

This is equivalent to the following inequality

$$N \log_q \alpha_q + (2\varepsilon(1-\varepsilon) - 1/2)\, k^2 + (2\varepsilon - 1/2)k \le 0. \tag{12}$$

The left side of the inequality is the square polynomial of ε. Not so hard to prove that (12) holds on the union of intervals

$$\left(-\infty, \frac{1}{2} - \frac{\sqrt{k+1+2b-1}}{2k}\right] \bigcup \left[\frac{1}{2} + \frac{\sqrt{k+1+2b+1}}{2k}, +\infty\right),$$

where $b = N \log_q \alpha_q$. The second half-interval cannot contain points of the segment $[k^{-1}, 4^{-1}]$ since $4^{-1} < 2^{-1}$. Therefore, let us require that the second contains it. For this, it is enough that

$$k^{-1} \le \frac{1}{2} - \frac{\sqrt{k+1+2N\log_q \alpha_q} - 1}{2k}.$$

The last is equivalent to

$$N \log_q \alpha_q \le \frac{k(k-3)}{2}. \tag{13}$$

Thus, if (13) holds, then we have (11). From (11) and (10), the inequality follows

$$Q \le 1 + \frac{3k}{2} q^{-N} q^{k(k+1)/2}.$$

But then from (7) we get an estimate for the probability

$$P \le \frac{3k}{4} q^{-N} + q^{-k(k+1)/2}.$$

If $N \le k(k+1)/2$ then $q^{-N} \ge q^{-k(k+1)/2}$, therefore we finally get

$$P \le \frac{3k+4}{4} q^{-N}.$$

Then, to satisfy (6), it is necessary to require that

$$-N + \log_q \frac{3k+4}{4} \le -\sum_{i=0}^{u} \dim \mathcal{C}_i^2 \Leftrightarrow N - \log_q \frac{3k+4}{4} \ge \sum_{i=0}^{u} \dim \mathcal{C}_i^2.$$

The theorem is completely proved. □

Now, to prove the main theorem 1, let us apply the statement of Theorem 2 to the generator matrix $G = (G_0 \| G_1 \| \dots \| G_u)$ of the code $\mathcal{C} \in cat(\mathcal{C}_0, \mathcal{C}_1, \dots, \mathcal{C}_u)$. The inequality $d_{\mathcal{C}}^{\perp} > 2$ guarantees that G does not contain identical columns. It is also by definition of set $cat(\mathcal{C}_0, \mathcal{C}_1, \dots, \mathcal{C}_u)$, for $i = 0, 1, \dots, u$ matrix G_i generates code $\mathcal{C}_i$.

5 Application to cryptanalysis of some post-quantum cryptographic mechanisms

Concatenated codes are sometimes used to construct post-quantum cryptographic mechanisms based on error-correcting codes.

So in work [20], it is proposed to construct the McEliece cryptosystem, use codes from the set $cat(\mathcal{C}_0, \mathcal{C}_1, \dots, \mathcal{C}_u)$, where C_i, $i = 0, 1, \dots, u$, is Reed–Muller code $RM(r, m)$.

An effective attack on this variant of the McEliece cryptosystem is proposed in [5]. In this case, for the attack to succeed, equality (2) must hold for code $\mathcal{C}$. The authors of the attack could only verify the equality (2) experimentally. Theorem 1 strictly allows proving this fact. So, for example, for the original parameters proposed in [20], $\dim \mathcal{C}_i^2 = \dim RM(6, 10) = 848$, $k = 176$, $N = 4 \cdot 1024 = 4096$, we get

$$4096 - \log_2 \frac{3 \cdot 176 + 4}{4} > 4096 - 8 = 4088 > 4 \cdot 848 = 3392,$$

which guarantees the success of the attack [5].

In [12], authors propose to build the cryptosystem based on codes from the family $cat(RM(r,m),\Gamma)$, where Γ is a binary Goppa code. Moreover, the codes $RM(r,m)$ and Γ are chosen so that their dimensions coincide. In [6], an attack on this cryptosystem is constructed in some adversary models. Among other things, the attack uses the fact of equality (2). Let 2^m is the length of $RM(r,m)$, n_1 is the length of Γ, and k is the dimension of these codes. Since it is not enough what is known about the Hadamard square of Γ, then we restrict its dimension to n_1. We get

$$2^m + n_1 - \log_2 \frac{3k+4}{4} \geq n_1 + \dim RM(2r,m).$$

So, if

$$\dim RM(2r,m) \leq 2^m - \log_2 \frac{3k+4}{4}, \tag{14}$$

then, in this case, it is possible to prove the efficiency of the attack from [6] rigorously. For example, for the code $RM(6,10)$, the inequality (14) holds since

$$848 \leq 1024 - \log_2 133 \approx 1016.$$

Note also the attack from [11], where the McEliece cryptosystem is constructed on the class of codes $cat(\mathcal{C}_0,\mathcal{C}_1,\ldots,\mathcal{C}_u)$ for a more general case of choosing codes $\mathcal{C}_i$. The center point of attack is equality (2). The authors note that they have experimentally verified its implementation, including for non-binary codes. It turned out that it is almost always fulfilled. Theorem 1 substantiates the experimental data from work [11].

Finally, consider the attack from [17]. It is devoted to the McEliece cryptosystem built on the Reed–Muller code $RM(r,m)$, in which random coordinates are added to each codeword so that the code's linearity is preserved. In Section 5.2 in Remark 1, the authors note that they experimentally established the following fact. If we add t random coordinates to the code $RM(r,m)$, then the Hadamard square of the new code $\mathcal{B}$ will have the dimension

$$\dim \mathcal{B}^2 = \dim RM(2r,m) + t.$$

Theorem 1 allows us to prove this fact. So code $\mathcal{B}$ can be considered a code from the family $cat(RM(r,m),\mathcal{C})$, where $\mathcal{C}$ is generated by a submatrix containing only added random columns of generator matrix of $\mathcal{B}$. However then the length of $\mathcal{C}$ is equal to t, therefore $\dim \mathcal{C}^2 \leq t$. It means that if the inequality (14) holds, then Theorem 1 implies the equality

$$\dim \mathcal{B}^2 = \dim RM(2r,m) + \dim \mathcal{C}^2.$$

It remains only to note that based on Theorem 2.2 of the article [3] for random linear codes with high probability $\dim \mathcal{C}^2 = t$. Moreover, for instance, when the set of added columns has the maximum rank t, then $\dim \mathcal{C}^2 = t$ with probability 1.

6 Conclusion remarks

The main theorem allows us to conclude that for some types of cryptographic mechanisms, the use of concatenation of codes from different classes instead of one class of codes, generally speaking, does not increase the cryptographic strength of the mechanism.

The authors hope that using the proven fact that the Hadamard square of the concatenated code is equal to the Cartesian product of Hadamard squares of the code-components, it will be possible to clarify several known attacks and build new attacks on post-quantum code-based cryptographic mechanisms.

This work was partially supported by the Russian Foundation for Basic Research under grant no. 18-29-03124.

References

[1] M. Bardet, M. Bertin, A. Couvreur, A. Otmani, "Practical Algebraic Attack on DAGS", *Code-Based Cryptography*, Lecture Notes in Computer Science, Springer International Publishing, Cham, 2019, 86-101.

[2] M. A. Borodin, I. V. Chizhov, "Effective attack on the McEliece cryptosystem based on Reed–Muller codes", *Diskr. Mat.*, **26**:1 (2014), 10–20; *Discrete Math. Appl.*, **24**:5 (2014), 273–280.

[3] I. Cascudo, R. Cramer, D. Mirandola, G. Zémor, "Squares of Random Linear Codes", *IEEE Transactions on Information Theory*, **61**:3 (2015), 1159-1173.

[4] H. Chen, R. Cramer, C. Dwork, "Algebraic geometric secret sharing schemes and secure multi-party computations over small fields", *Advances in cryptology - CRYPTO 2006*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, 521–536.

[5] I. Chizhov, S. Koniukhov, A. Davletshina, "Effective structural attack on McEliece-Sidelnikov public-key cryptosystem", *International Journal of Open Information Technologies*, **8**:7 (2020), 1–10, In Russian.

[6] I. Chizhov, E. Popova, "Structural attack on McEliece-Sidelnikov type public-key cryptosystem based on a combination of random codes with Reed-Muller codes", *International Journal of Open Information Technologies*, **8**:6 (2020), 24–33, In Russian.

[7] A. Couvreur, P. Gaborit, V. Gauthier-Umaña, A. Otmani, J.-P. Tillich, "Distinguisher-based attacks on public-key cryptosystems using Reed–Solomon codes", *Des. Codes Cryptogr*, **73** (2014), 641–666.

[8] A. Couvreur, I. Márquez-Corbella, R. Pellikaan, "Cryptanalysis of Public-Key Cryptosystems That Use Subcodes of Algebraic Geometry Codes", *Coding Theory and Applications*, CIM Series in Mathematical Sciences, Springer International Publishing, Cham, 2015, 133-140.

[9] A. Couvreur, A. Otmani, J.-P. Tillich, "Polynomial Time Attack on Wild McEliece Over Quadratic Extensions", *IEEE Transactions on Information Theory*, **63**:1 (2017), 404-427.

[10] A. Couvreur, A. Otmani, J.-P. Tillich, V. Gauthier–Umaña, "A Polynomial-Time Attack on the BBCRS Scheme", *Public-Key Cryptography – PKC 2015*, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2015, 175-193.

[11] V. M. Deundyak, Y. V. Kosolapov, "On the strength of asymmetric code cryptosystems based on the merging of generating matrices of linear codes", *2019 XVI International Symposium "Problems of Redundancy in Information and Control Systems" (REDUNDANCY)* (2019 XVI International Symposium "Problems of Redundancy in Information and Control Systems" (REDUNDANCY)), 2019, 143-148.

[12] E. Egorova, G. Kabatiansky, E. Krouk, C. Tavernier, "A new code-based public-key cryptosystem resistant to quantum computer attacks", *J. Phys.: Conf. Ser.*, **1163** (2019), 012061.

[13] J. Faugère, V. Gauthier-Umaña, A. Otmani, L. Perret, J.-P. Tillich, "A Distinguisher for High-Rate McEliece Cryptosystems", *IEEE Transactions on Information Theory*, **59**:10 (2013), 6830-6844.

[14] S. Li, "On the weight distribution of second order Reed–Muller codes and their relatives", *Designs, Codes and Cryptography*, **87**:10 (2019), 2447-2460.

[15] R. J. McEliece, "Quadratic forms over finite fields and second-order Reed-Muller codes", *JPL Space Programs Summary*, **3** (1969), 37–58.

[16] F. J. McWilliams, N. J. A. Sloane, *The theory of error-correcting codes. I and II.*, North-Holland mathematical library; v. 16, North-Holland Pub. Co., North Holland, New York, 1977.

[17] A. Otmani, H. Kalachi, "Square Code Attack on a Modified Sidelnikov Cryptosystem", *Codes, Cryptology, and Information Security*, Lecture Notes in Computer Science, Springer International Publishing, 2015, 173–183.

[18] R. Pellikaan, "On decoding by error location and dependent sets of error positions", *Discrete Mathematics*, **106–107** (1992), 369-381.

[19] H. Randriambololona, "An Upper Bound of Singleton Type for Componentwise Products of Linear Codes", *IEEE Transactions on Information Theory*, **59**:12 (2013), 7936-7939.

[20] V. M. Sidel'nikov, "Open coding based on Reed–Muller binary codes", *Diskr. Mat.*, **6**:2 (1994), 3–20; *Discrete Math. Appl.*, **4**:3 (1994), 191–207.

[21] N. Sloane, E. Berlekamp, "Weight enumerator for second-order Reed-Muller codes", *IEEE Transactions on Information Theory*, **16**:6 (1970), 745–751.

[22] C. Wieschebrink, "Cryptanalysis of the Niederreiter public key scheme based on GRS subcodes", *Lecture Notes in Computer Science*, **6061 LNCS** (2010), 61–72.

Public key cryptography

Small Scalar Multiplication on Weierstrass Curves using Division Polynomials

Sergey Agievich, Stanislav Poruchnik, and Vladislav Semenov

Research Institute for Applied Problems of Mathematics and Informatics
Belarusian State University, Belarus
agievich@bsu.by, semenov.vlad.by@gmail.com, poruchnikstanislav@gmail.com

Abstract

This paper deals with elliptic curves in the short Weierstrass form over large prime fields and presents algorithms for computing small odd multiples of a given point P on a curve. Our algorithms make use of division polynomials and are more efficient than the naive algorithm based on repeated additions with $2P$. We show how to perform scalar multiplication in the variable base settings using the precomputed small multiples. By employing the window method and avoiding conditional branches, we achieve the constant-time property for the final scalar multiplication algorithm. Small multiples are computed in either Jacobian or affine coordinates. To obtain affine coordinates, we use Montgomery's trick of simultaneous multiplicative inversion of several field elements. The conversion to affine coordinates slows down the precomputations but speeds up the main scalar multiplication loop. We show that the conversion makes sense and gives an overall performance boost in practical settings.

Keywords: elliptic curve, short Weierstrass form, division polynomial, scalar multiplication.

1 Preliminaries

Elliptic curves in the short Weierstrass form are historically the first curves of ECC (Elliptic Curve Cryptography, [6]). They appeared in the pioneering papers by V. Miller [12] and N. Koblitz [9], they formed the basis of a dozen cryptography standards.

Comparing the efficiency of arithmetic operations, Weierstrass curves are not the fastest. They are inferior to Montgomery and Edwards curves [13, 2, 3, 4], the current champions. Despite this, Weierstrass curves continue to be widely used. It is not only due to legacy issues. Unlike Montgomery and Edwards curves, Weierstrass curves can possess the cofactor-one property, which makes them very convenient for use in various cryptographic protocols.

Let $\mathbb{F}$ be a large prime finite field. An elliptic curve over $\mathbb{F}$ in the short Weierstrass form is defined by the equation

$$E\colon y^2 = x^3 + ax + b, \quad a, b \in \mathbb{F}, \quad 4a^3 + 27b^2 \neq 0.$$

Affine points of the curve, that is, pairs $(x, y) \in \mathbb{F}^2$ satisfying E, are added using special rules. A resulting sum can be a special point at infinity, denoted by O, and this point can also be a summand. The addition is defined in such a way that affine points complemented by O form an abelian group. In this group, O is zero and $(x, -y)$ is the inverse of (x, y).

Let G be an affine point, $\mathbb{G} = \langle G \rangle$ be the corresponding cyclic group, q be an order of $\mathbb{G}$ and l be the length of q in bits. In cryptography, (E, G) are chosen so that q is a large prime close to $|\mathbb{F}|$. We further assume that this is indeed the case.

The main operation of ECC is scalar multiplication, that is, computing dP given $P \in \mathbb{G} \setminus \{O\}$ and $d \in \{1, 2, \ldots, q-1\}$. We consider the so-called *variable base* settings, when P is volatile and precomputations with it are not possible. These settings cover constructing shared secrets in Diffie-Hellman-type protocols or verifying signatures of ElGamal and Schnorr types. The scalar d is often secret.

Computations with points of an elliptic curve are reduced to computations with their coordinates (elements of $\mathbb{F}$), which are described by arithmetic (over $\mathbb{F}$) circuits. The main contribution to the circuit complexity is made by the multiplicative operations: $\mathtt{I}$ — inversion, $\mathtt{M}$ — multiplication of arbitrary elements, $\mathtt{S}$ — squaring. The notation $i\mathtt{I} + m\mathtt{M} + s\mathtt{S}$ means that a circuit contains i operations $\mathtt{I}$, m operations $\mathtt{M}$ and s operations $\mathtt{S}$. For example, on Weierstrass curves, the addition of affine points and their doubling can be done with the complexity $1\mathtt{I} + 2\mathtt{M} + 1\mathtt{S}$ and $1\mathtt{I} + 2\mathtt{M} + 2\mathtt{S}$ respectively.

The operation $\mathtt{I}$ is the most expensive, for practical dimensions its complexity is 80–100 times higher than that of $\mathtt{M}$. To reduce the use of $\mathtt{I}$, affine points (x, y) are converted into projective points $(X, Y, Z) \in \mathbb{F}^3$. We use Jacobian projective points: $X/Z^2 = x$, $Y/Z^3 = y$. The coordinate Z acts as a normalizing factor that "absorbs" the inconvenient operation $\mathtt{I}$. On a Weierstrass curve with $a = -3$ (this is the optimal choice), the operations $J \leftarrow J + J$ (addition of Jacobian points providing the Jacobian result), $J \leftarrow 2J$ (doubling of Jacobian points), $J \leftarrow J + A$ (addition of Jacobian points with affine ones) can be done with the complexity $11\mathtt{M} + 5\mathtt{S}$, $3\mathtt{M} + 5\mathtt{S}$ and $7\mathtt{M} + 4\mathtt{S}$ respectively.

There are many methods for scalar multiplication $(d, P) \mapsto dP$. In this paper, we use the window method which consists of two stages:

I. For a small *window width* w and a *base* $\mathcal{B} \subseteq \{\pm 1, \pm 2, \ldots, \pm(2^w - 1)\}$, the small multiples $\{nP \colon n \in \mathcal{B}\}$ are computed.
II. The resulting point dP is computed using point doublings and additions with small multiples. A sequence of multiples added at each step is determined using a *recording* algorithm that processes the binary representation of d. The bits of d are processed either from right (less significant) to left (more significant) or from left to right.

We propose algorithms that implement both stages of the window method with $\mathcal{B} = \{\pm 1, \pm 3, \ldots, \pm(2^w - 1)\}$. This base was introduced in [14] and then used in [7] accompanied by a different recording (*recoding* in the original) algorithm. Both recording algorithms of [14] and [7] are right-to-left. We use the left-to-right direction. This direction is not the best (see discussion in [7]) from the perspective of power analysis, a branch of side-channel attacks, but in this paper, we do not take power analysis into consideration.

Our algorithms of the first stage are based on division polynomials. Their use for scalar multiplication was proposed by V. Miller in the already mentioned paper [12]. It appears that Miller's proposal was first implemented in [8] by adapting an algorithm for computing elliptic nets from [15]. The proposal is not used in practice since the resulting circuits for computing dP have high complexity. On the other hand, the circuits for computing small multiples of P have acceptable complexity and, moreover, these circuits are more efficient than the naive circuit based on repeated additions with $2P$ (see below). Note that we use division polynomials in a rather straightforward manner compared to a more sophisticated approach of [8, 15] where the double-and-add method based on division polynomials is actually designed.

In our algorithm of the second stage, conditional branches are avoided. In cryptography, algorithms without branches are called *constant-time*. Only constant-time algorithms are considered safe since branches usually induce fluctuations of the runtime with potential leakage of sensitive data (in our case, bits of d).

The small multiples of P that are computed at the first stage and used at the second stage can be either Jacobian or affine points. To compute affine points, we use a circuit for simultaneous inversion of several field elements with only one `I` operation and some extra multiplications. This circuit was proposed by P. Montgomery in [13] and is often referred to as Montgomery's trick. When using affine points, the first stage is slower but the second stage is faster.

The following table summarizes the complexity of the proposed algorithms applying for Weierstrass curves in the short form with $a = -3$. In

the table, $k = \lceil l/w \rceil$. The second column presents the form of the small multiples: in Jacobian coordinates (J) or in affine (A). The complexity of the first stage actually covers the computation of only $2^{w-1} - 1$ points nP, $n = 3, 5, \ldots, 2^w - 1$, since the remaining points of the form nP, $n \in \mathcal{B}$, can be computed through field negations, that is, without multiplicative operations.

Stage	Small multiples	Complexity
I	J	$(19 \cdot 2^{w-2} - 11)\mathtt{M} + (7 \cdot 2^{w-2} - 1)\mathtt{S}$
	A	$\mathtt{I} + (25 \cdot 2^{w-2} - 13)\mathtt{M} + (5 \cdot 2^{w-1} - 3)\mathtt{S}$
II	J	$\mathtt{I} + (3(k-1)w + 11k - 8)\mathtt{M} + (6(k-1)w + 5k - 4)\mathtt{S}$
	A	$\mathtt{I} + (3(k-1)w + 8k - 5)\mathtt{M} + (6(k-1)w + k)\mathtt{S}$

Let us discuss the first line of the table. It means that the computation of one small multiple takes time $\approx \frac{19}{2}\mathtt{M} + \frac{7}{2}\mathtt{S}$. On the other hand, the usual algorithm for computing small multiples is to compute the Jacobian point $2P$ and then add it with P, $3P$, $5P$, and so on. Each addition, except the first, is performed in time $\approx 11\mathtt{M} + 5\mathtt{S}$. As we can see, using division polynomials to compute small multiples gives improvement.

The rest of the paper is organized as follows. In Section 2 we recall the notion of division polynomials and outline a way to use them for computing small multiples. The exact algorithms and their complexity are provided in Section 3. In Section 4 we present the constant-time window algorithm for scalar multiplication in variable base settings. Using popular heuristics about the complexity ratio between I, M, S and additive field operations, we estimate the optimal window width for $l = 256, 384, 512$ and different coordinate systems of small multiples (Jacobian or affine). It turns out that affine coordinates are better in all considered cases.

2 Division polynomials

Division polynomials $\psi_n(x, y)$ describe the coordinates of multiples of an elliptic curve point. More precisely, if $P = (x, y)$ is an affine point, $n \geq 2$ and $nP \neq O$, then

$$nP = \left(x - \frac{\psi_{n-1}(P)\psi_{n+1}(P)}{\psi_n(P)^2}, \frac{\psi_{n+2}(P)\psi_{n-1}(P)^2 - \psi_{n-2}(P)\psi_{n+1}(P)^2}{4y\psi_n(x,y)^3} \right).$$

The polynomials $\psi_n(x, y)$ are computed recursively. It would be convenient to define the recursion in terms of auxiliary polynomials $W_n(x, y^2)$ such that

$$\psi_n(x, y) = \begin{cases} 2yW_n(x, y^2), & n \text{ is even}, \\ W_n(x, y^2), & n \text{ is odd}. \end{cases}$$

Assuming (x, y) is fixed, denote $W_n = W_n(x, y^2)$. With that,

$$W_n = \begin{cases} -1, & n = -1, \\ 0, & n = 0, \\ 1, & n = 1, \\ 1, & n = 2, \\ 3(x^2+a)^2 + 4(3bx - a^2), & n = 3, \\ 2(x^4(x^2+5a) + bx(20x^2 - 4a) - 5a^2x^2 - 8b^2 - a^3), & n = 4, \\ W_m \left(W_{m+2}W_{m-1}^2 - W_{m-2}W_{m+1}^2\right), & n = 2m, \\ \left((2y)^4 W_m W_{m+2}\right) W_m^2 - \left(W_{m-1}W_{m+1}\right) W_{m+1}^2, & n = 4k+1, \quad m = 2k, \\ \left(W_m W_{m+2}\right) W_m^2 - \left((2y)^4 W_{m-1}W_{m+1}\right) W_{m+1}^2, & n = 4k+3, \quad m = 2k+1. \end{cases}$$

For odd n, the affine coordinates of the point nP can be represented as follows:

$$\begin{aligned} nP &= \left(\frac{X_n}{W_n^2}, \frac{Y_n}{W_n^3}\right), \\ X_n &= xW_n^2 - (2y)^2 W_{n-1}W_{n+1}, \\ Y_n &= y\left(W_{n+2}W_{n-1}^2 - W_{n-2}W_{n+1}^2\right). \end{aligned}$$

Besides, we automatically get the Jacobian form of nP: $nP = (X_n, Y_n, W_n)$.

Carrying out the computation for n up to $2^w - 1$, we obtain an arithmetic circuit with two non-constant inputs, that is, the affine coordinates of P, and $3 \cdot (2^{w-1} - 1)$ outputs, that is, the Jacobian coordinates of the points $3P, 5P, \ldots, (2^w - 1)P$.

The resulting circuit can be extended with a circuit taking Jacobian coordinates and returning affine ones. In the additional circuit, we use the following algorithm proposed by P. Montgomery and mentioned in Section 1. The algorithm inverts k nonzero field elements in time $\mathtt{I} + (3k - 3)\mathtt{M}$.

Algorithm `MontInv`

Input: $(u_1, \ldots, u_k)$, $u_i \in \mathbb{F}$, $u_i \neq 0$.

Output: $(u_1^{-1}, \ldots, u_k^{-1})$.

Steps:

1. $v_1 \leftarrow u_1$.
2. For $i = 2, \ldots, k$: $v_i \leftarrow v_{i-1}u_i$.
3. $t \leftarrow v_k^{-1}$.
4. For $i = k, k-1, \ldots, 2$:
   (1) $u_i^{-1} \leftarrow v_{i-1}t$;
   (2) $t \leftarrow tu_i^{-1}$.
5. $u_1^{-1} \leftarrow t$.

6. Return $(u_1^{-1}, \ldots, u_k^{-1})$.

Further, to slightly simplify the transition from Jacobian to affine coordinates, we use the following form of nP:

$$nP = \left(\frac{X_n}{W_n^2}, \frac{Y'_n}{W_n^4}\right),$$
$$Y'_n = y\left(W_n W_{n+2} W_{n-1}^2 - W_{n-2} W_n W_{n+1}^2\right).$$

3 Small scalar multiplication

3.1 Jacobian coordinates

Let us analyze equations for computing the Jacobian points nP, $n = 3, 5, \ldots, 2^w - 1$. We make a list of intermediate and final expressions to be computed, determine which expressions need to be computed before others, count the arithmetic complexity of the computations. We count not only multiplicative operations of the field $\mathbb{F}$ but also simpler ones: $\mathtt{A}$ — addition or subtraction, $\mathtt{m}$ — multiplication by a small (≤ 5) constant, $\mathtt{half}$ — division by 2. Multiplication by the curve coefficient a or b is treated as the operation $\mathtt{M}$. We suppose that the expressions a^2, $a^3 + 8b^2$ are precomputed and can be used along with a and b at no cost.

The results of the analysis are given in Table 1.

Table 1: Computing small multiples in Jacobian coordinates

Expression(s)	n	Require(s)	Complexity
W_n	$1, 2, 3, 4$	a, b, a^2, $a^3 + 8b^2$, x	$4\mathtt{M} + 3\mathtt{S} + 7\mathtt{m} + 10\mathtt{A}$
$y^2, (2y)^2$		y	$1\mathtt{S} + 1\mathtt{m}$
W_{2n}	$3, 4, \ldots, 2^{w-1}$	W_n, $W_{n+2}W_{n-1}^2 - W_{n-2}W_{n+1}^2$	$(2^{w-1} - 2)\mathtt{M}$
W_{2n+1}	$2, 3, \ldots, 2^{w-1}$	$(2y)^4 W_n W_{n+2}$, W_n^2, W_{n+1}^2, $W_{n-1}W_{n+1}$	$(2^{w-1} - 1)(2\mathtt{M} + \mathtt{A}) - \mathtt{M}$
W_n^2	$1, 2, \ldots, 2^w$	W_n	$(2^w - 2)\mathtt{S}$
$W_n W_{n+2}$	$1, 2, \ldots, 2^{w-1}$ $2, 4, \ldots, 2^w - 2$	W_n, W_{n+2}, W_n^2, W_{n+2}^2	$(3 \cdot 2^{w-2} - 3)(\mathtt{S} + 3\mathtt{A} + \mathtt{half})$
$(2y)^2 W_n W_{n+2}$	$2, 4, \ldots, 2^w - 2$	$(2y)^2$, $W_n W_{n+2}$	$(2^{w-1} - 1)\mathtt{M}$
$(2y)^4 W_n W_{n+2}$	$2, 4, \ldots, 2^{w-1}$	$(2y)^2$, $(2y)^2 W_n W_{n+2}$	$2^{w-2}\mathtt{M}$
$W_{n+2}W_{n-1}^2 - W_{n-2}W_{n+1}^2$	$3, 4, \ldots, 2^{w-1}$ $3, 5, \ldots, 2^w - 1$	W_{n-2}, W_{n+2}, W_{n-1}^2, W_{n+1}^2	$(3 \cdot 2^{w-1} - 7)\mathtt{M} +$ $(3 \cdot 2^{w-2} - 2)\mathtt{A}$
X_n	$3, 5, \ldots, 2^w - 1$	x, W_n^2, $(2y)^2 W_{n-1} W_{n+1}$	$(2^{w-1} - 1)(\mathtt{M} + \mathtt{A})$
Y_n	$3, 5, \ldots, 2^w - 1$	y, $W_{n+2}W_{n-1}^2 - W_{n-2}W_{n+1}^2$	$(2^{w-1} - 1)\mathtt{M}$

Calculating the complexity, we take into account the following simplifications:

$$
\begin{aligned}
&W_5 = (2y)^4 W_2 W_4 - W_3 W_3^2,\\
&W_1^2 = W_2^2 = 1,\\
&W_n W_{n+2} = ((W_n + W_{n+2})^2 - W_n^2 - W_{n+2}^2)/2,\\
&W_1 W_3 = W_3,\\
&W_2 W_4 = W_4,\\
&W_5 W_2^2 - W_1 W_4^2 = W_5 - W_4^2,\\
&W_6 W_3^2 - W_2 W_5^2 = W_6 W_3^2 - W_5^2.
\end{aligned}
$$

The total complexity of the circuit for computing small multiples in Jacobian coordinates:

$$(19 \cdot 2^{w-2} - 11)\mathtt{M} + (7 \cdot 2^{w-2} - 1)\mathtt{S} + 8\mathtt{m} + (2^{w+2} - 3)\mathtt{A} + (3 \cdot 2^{w-2} - 3)\mathtt{half}.$$

The circuit is presented in detail in the algorithm `SmallMultJ`. Here we take into account that certain expressions are used multiple times. Such expressions are cached implicitly, that is, they are stored locally and then reused without recalculation. Cached expressions are enclosed in square brackets. Expressions in the brackets to the left of $\leftarrow$ are inserted into the cache, and expressions to the right are retrieved from it.

Algorithm `SmallMultJ`

Input: $P = (x, y) \in \mathbb{G} \setminus \{O\}$, w $(3 \le w < \log_2 q)$.

Output: $3P, 5P, \ldots, (2^w - 1)P$ (in Jacobian coordinates).

Steps:

1. $[y^2] \leftarrow y^2$.
2. $[(2y)^2] \leftarrow 4 \cdot [y^2]$.
3. Compute W_3:
   (1) $[x^2] \leftarrow x^2$;
   (2) $[bx] \leftarrow b \cdot x$;
   (3) $[W_3] \leftarrow 3 \cdot ([x^2] + a)^2 - 4 \cdot ([a^2] - 3 \cdot [bx])$.
4. Compute W_4:
   (1) $[ax] \leftarrow a \cdot x$;
   (2) $[x^3] \leftarrow [y^2] - [ax] - b$;
   (3) $[W_4] \leftarrow 2 \cdot ([x^3]^2 + 4 \cdot [bx] \cdot (5 \cdot [x^2] - a) + 5 \cdot [ax] \cdot ([x^3] - [ax]) - [a^3 + 8b^2])$.
5. $[W_2^2] \leftarrow 1$.
6. $[W_3^2] \leftarrow [W_3]^2$.
7. $[W_4^2] \leftarrow [W_4]^2$.

8. $[W_1W_3] \leftarrow [W_3]$.
9. $[W_2W_4] \leftarrow [W_4]$.
10. $[(2y)^2W_2W_4] \leftarrow [(2y)^2] \cdot [W_2W_4]$.
11. $[(2y)^4W_2W_4] \leftarrow [(2y)^2] \cdot [(2y)^2W_2W_4]$.
12. $[W_5] \leftarrow [(2y)^4W_2W_4] - [W_1W_3] \cdot [W_3^2]$.
13. $[W_5^2] \leftarrow ([W_5])^2$.
14. $[W_5W_2^2 - W_1W_4^2] \leftarrow [W_5] - [W_4^2]$.
15. $[W_6W_3^2 - W_2W_5^2] \leftarrow [W_6] \cdot [W_3^2] - [W_5^2]$.
16. For $n = 3, 4, \ldots, 2^{w-1}$:
    (1) if $n \geq 5$:
        $[W_{n+2}W_{n-1}^2 - W_{n-2}W_{n+1}^2] \leftarrow [W_{n+2}] \cdot [W_{n-1}^2] - [W_{n-2}] \cdot [W_{n+1}^2]$;
    (2) $[W_{2n}] \leftarrow [W_n] \cdot [W_{n+2}W_{n-1}^2 - W_{n-2}W_{n+1}^2]$;
    (3) $[W_{2n}^2] \leftarrow ([W_{2n}])^2$;
    (4) $[W_nW_{n+2}] \leftarrow (([W_n] + [W_{n+2}])^2 - [W_n^2] - [W_{n+2}^2])/2$;
    (5) if n is odd:
        a) $[W_{2n+1}] \leftarrow [W_nW_{n+2}] \cdot [W_n^2] - [(2y)^4W_{n-1}W_{n+1}] \cdot [W_{n+1}^2]$;
        else:
        a) $[(2y)^2W_nW_{n+2}] \leftarrow [(2y)^2] \cdot [W_nW_{n+2}]$;
        b) $[(2y)^4W_nW_{n+2}] \leftarrow [(2y)^2] \cdot [(2y)^2W_nW_{n+2}]$;
        c) $[W_{2n+1}] \leftarrow [(2y)^4W_nW_{n+2}] \cdot [W_n^2] - [W_{n-1}W_{n+1}] \cdot [W_{n+1}^2]$;
    (6) if $n \neq 2^{w-1}$:
        $[W_{2n+1}^2] \leftarrow ([W_{2n+1}])^2$.
17. For $n = 3, 5, \ldots, 2^{w-1} + 1$:
    (1) $[X_n] \leftarrow x \cdot [W_n^2] - [(2y)^2W_{n-1}W_{n+1}]$.
18. For $n = 2^{w-1} + 3, 2^{w-1} + 5, \ldots, 2^w - 1$:
    (1) $t \leftarrow (([W_{n-1}] + [W_{n+1}])^2 - [W_{n-1}^2] - [W_{n+1}^2])/2$;
    (2) $[X_n] \leftarrow x \cdot [W_n^2] - [(2y)^2] \cdot t$.
19. For $n = 3, 5, \ldots, 2^{w-1} - 1$:
    (1) $[Y_n] \leftarrow y \cdot [W_{n+2}W_{n-1}^2 - W_{n-2}W_{n+1}^2]$.
20. For $n = 2^{w-1} + 1, 2^{w-1} + 3, \ldots, 2^w - 1$:
    (1) $[Y_n] \leftarrow y \cdot ([W_{n+2}] \cdot [W_{n-1}^2] - [W_{n-2}] \cdot [W_{n+1}^2])$.
21. Return the Jacobian points $nP = ([X_n], [Y_n], [W_n])$, $n = 3, 5, \ldots, 2^w - 1$.

The algorithm `SmallMultJ` is constant-time. Indeed, its branching conditions depend only on the window width w (public parameter), but not on the base point P.

The algorithm uses $18 \cdot 2^{w-2} + 1$ field registers, that is, memory cells for storing elements of $\mathbb{F}$. Some registers can be reused but we do not consider this optimization here.

3.2 Affine coordinates

To compute small multiples nP, $n = 3, 5, \ldots, 2^w - 1$ in affine coordinates, we modify the previous circuit as follows.

1. The expressions $W_n W_{n+2}$ are computed for $n = 1, 2, \ldots, 2^w - 1$. For $n = 1, 2$ the computation again has no cost since $W_1 W_3 = W_3$ and $W_2 W_4 = W_4$. For $3 \le n \le 2^w - 2$ we again use the scheme $W_n W_{n+2} = ((W_n + W_{n+2})^2 - W_n^2 - W_{n+2}^2)/2$. For $n = 2^w - 1$ we directly multiply W_n by W_{n+2} as $W_{2^w+1}^2$ is not computed. The overall complexity: $(2^w - 4)(\mathtt{S} + 3\mathtt{A} + \mathtt{half}) + \mathtt{M}$.
2. Instead of the expressions $W_{n+2} W_{n-1}^2 - W_{n-2} W_{n+1}^2$ we compute $W_n W_{n+2} W_{n-1}^2$ and $W_{n-2} W_n W_{n+1}^2$. We multiply $W_n W_{n+2}$ by W_{n-1}^2 and $W_{n-2} W_n$ by W_{n+1}^2. Complexity: $(3 \cdot 2^{w-1} - 5)\mathtt{M}$. Here we take into account the simplification: $W_3 W_5 W_2^2 = W_3 W_5$.
3. The expressions W_{2n}, $n = 3, \ldots, 2^{w-1}$, are computed by subtracting $W_{n-2} W_n W_{n+1}^2$ from $W_n W_{n+2} W_{n-1}^2$. Complexity: $(2^{w-1} - 2)\mathtt{A}$.
4. The expressions W_n^{-2}, $n = 3, 5, \ldots, 2^w - 1$, are computed simultaneously using `MontInv`. After that we compute W_n^{-4} by squaring W_n^{-2}. Complexity: $\mathtt{I} + (3 \cdot (2^{w-1} - 1) - 3)\mathtt{M} + (2^{w-1} - 1)\mathtt{S}$.
5. Instead of Y_n we compute Y'_n, $n = 3, 5, \ldots, 2^w - 1$. The computations use y, $W_n W_{n+2} W_{n-1}^2$ and $W_{n-2} W_n W_{n+1}^2$. For each n it takes one multiplication and for $n > 2^{w-1}$ it additionally takes one subtraction. Complexity: $(2^{w-1} - 1)\mathtt{M} + 2^{w-2}\mathtt{A}$.
6. The conversion into affine coordinates consists of multiplying X_n by W_n^{-2} and Y'_n by W_n^{-4}, $n = 3, 5, \ldots, 2^w - 1$. Complexity: $(2^w - 2)\mathtt{M}$.

The total complexity of the modified circuit:

$$\mathtt{I} + (25 \cdot 2^{w-2} - 13)\mathtt{M} + (5 \cdot 2^{w-1} - 3)\mathtt{S} + 8\mathtt{m} + (19 \cdot 2^{w-2} - 6)\mathtt{A} + (2^w - 4)\mathtt{half}.$$

The circuit is detailed in the following algorithm.

Algorithm `SmallMultA`

Input: $P = (x, y) \in \mathbb{G} \setminus \{O\}$, w $(3 \le w < \log_2 q)$.

Output: $3P, 5P, \ldots, (2^w - 1)P$ (in affine coordinates).

Steps:

1. $[y^2] \leftarrow y^2$.
2. $[(2y)^2] \leftarrow 4 \cdot [y^2]$.
3. Compute W_3:
   (1) $[x^2] \leftarrow x^2$;
   (2) $[bx] \leftarrow b \cdot x$;

(3) $[(x^2+a)^2] \leftarrow ([x^2]+a)^2$;
(4) $[W_3] \leftarrow 3 \cdot [(x^2+a)^2] - 4 \cdot ([a^2] - 3 \cdot [bx])$.

4. Compute W_4:
   (1) $[ax] \leftarrow a \cdot x$;
   (2) $[x^3] \leftarrow [y^2] - [ax] - b$;
   (3) $[x^6] \leftarrow ([x^3])^2$;
   (4) $[W_4] \leftarrow 2 \cdot ([x^6] + 4 \cdot [bx] \cdot (5 \cdot [x^2] - a) + 5 \cdot [ax] \cdot ([x^3] - [ax]) - [a^3 + 8b^2])$.
5. $[W_2^2] \leftarrow 1$.
6. $[W_3^2] \leftarrow [W_3]^2$.
7. $[W_4^2] \leftarrow [W_4]^2$.
8. $[W_1 W_3] \leftarrow [W_3]$.
9. $[W_2 W_4] \leftarrow [W_4]$.
10. $[(2y)^2 W_2 W_4] \leftarrow [(2y)^2] \cdot [W_2 W_4]$.
11. $[(2y)^4 W_2 W_4] \leftarrow [(2y)^2] \cdot [(2y)^2 W_2 W_4]$.
12. $[W_5] \leftarrow [(2y)^4 W_2 W_4] - [W_1 W_3] \cdot [W_3^2]$.
13. $[W_5^2] \leftarrow ([W_5])^2$.
14. For $n = 3, 4, \ldots, 2^{w-1}$:
    (1) $[W_n W_{n+2}] \leftarrow (([W_n] + [W_{n+2}])^2 - [W_n^2] - [W_{n+2}^2])/2$;
    (2) if $n = 3$:
        $[W_{2n}] \leftarrow [W_n W_{n+2}] - [W_{n-2} W_n] \cdot [W_{n+1}^2]$;
        else:
        $[W_{2n}] \leftarrow [W_n W_{n+2}] \cdot [W_{n-1}^2] - [W_{n-2} W_n] \cdot [W_{n+1}^2]$;
    (3) $[W_{2n}^2] \leftarrow ([W_{2n}])^2$.
    (4) if n is odd:
        a) $[W_{2n+1}] \leftarrow [W_n W_{n+2}] \cdot [W_n^2] - [(2y)^4 W_{n-1} W_{n+1}] \cdot [W_{n+1}^2]$;
        else:
        a) $[(2y)^2 W_n W_{n+2}] \leftarrow [(2y)^2] \cdot [W_n W_{n+2}]$;
        b) $[(2y)^4 W_n W_{n+2}] \leftarrow [(2y)^2] \cdot [(2y)^2 W_n W_{n+2}]$;
        c) $[W_{2n+1}] \leftarrow [(2y)^4 W_n W_{n+2}] \cdot [W_n^2] - [W_{n-1} W_{n+1}] \cdot [W_{n+1}^2]$;
    (5) if $n \neq 2^{w-1}$:
        $[W_{2n+1}^2] \leftarrow ([W_{2n+1}])^2$.
15. $[W_3^{-2}], [W_5^{-2}], \ldots, [W_{2^w-1}^{-2}] \leftarrow \texttt{MontInv}([W_3^2], [W_5^2], \ldots, [W_{2^w-1}^2])$.
16. For $n = 3, 5, \ldots, 2^{w-1} + 1$:
    (1) $[X'_n] \leftarrow x - [(2y)^2 W_{n-1} W_{n+1}] \cdot [W_n^{-2}]$.
17. For $n = 2^{w-1} + 3, 2^{w-1} + 5, \ldots, 2^w - 1$:
    (1) $t \leftarrow (([W_{n-1}] + [W_{n+1}])^2 - [W_{n-1}^2] - [W_{n+1}^2])/2$;
    (2) $[X'_n] \leftarrow x - [(2y)^2] \cdot t \cdot [W_n^{-2}]$.
18. For $n = 3, 5, \ldots, 2^{w-1} - 1$:
    (1) $[Y'_n] \leftarrow y \cdot [W_{2n}] \cdot ([W_n^{-2}])^2$.

19. For $n = 2^{w-1}+1, 2^{w-1}+3, \ldots, 2^w-3$:
    (1) $[W_nW_{n+2}] \leftarrow (([W_n]+[W_{n+2}])^2 - [W_n^2] - [W_{n+2}^2])/2$;
    (2) $[Y'_n] \leftarrow y \cdot ([W_nW_{n+2}] \cdot [W_{n-1}^2] - [W_{n-2}W_n] \cdot [W_{n+1}^2]) \cdot ([W_n^{-2}])^2$.
20. $[Y'_{2^w-1}] \leftarrow y \cdot ([W_{2^w-1}] \cdot [W_{2^w+1}] \cdot [W_{2^w-2}^2] - [W_{2^w-3}W_{2^w-1}] \cdot [W_{2^w}^2]) \cdot ([W_{2^w-1}^{-2}])^2$.
21. Return the affine points $nP = ([X'_n], [Y'_n])$, $n = 3, 5, \ldots, 2^w-1$.

The algorithm `SmallMultA` is constant-time for the same reasons as for `SmallMultJ`. The algorithm uses $17 \cdot 2^{w-2} + 6$ field registers.

4 Scalar multiplication

Let us proceed with scalar multiplication, that is, computing dP from an affine point $P = (x, y)$ and a scalar $d \in \{1, 2, \ldots, q-1\}$. Recall that q is a large (odd) prime and the length of q in bits equals l.

We start by choosing some window width w and computing the small multiples nP, $n = 3, 5, \ldots, 2^w-1$, using either the `SmallMultJ` or `SmallMultA` algorithm. Next, we compute the negative small multiples $-nP$, $n = 1, 3, \ldots, 2^w-1$, with a small overhead ($2^{w-1}\mathtt{A}$).

The computation of dP is performed in Jacobian coordinates. We use point doublings ($J \leftarrow 2J$) and additions with the points $\pm nP$ (either $J \leftarrow J+J$ or $J \leftarrow J+A$ depending on the form of the small multiples). The resulting Jacobian point is converted into the affine one ($A \leftarrow J$). To determine small multiples added at each step, we record d as follows.

First, we write d in base 2^w:

$$d = \sum_{i=0}^{k-1} d_i 2^{wi}. \qquad (\star)$$

Here $d_i \in \{0, 1, \ldots, 2^w-1\}$ are digits of the representation and $k = \lceil l/w \rceil$ is their number.

Second, for odd d, the digits d_i are adjusted to get into the set $\mathcal{B} = \{\pm 1, \pm 3, \ldots, \pm(2^w-1)\}$ such that $(\star)$ is still valid. The adjustment is as follows. For $i = k-1, k-2, \ldots, 1$, the parity of the digit d_i is tested. If the digit is even, it is increased by 1 and the previous digit is simultaneously decreased by 2^w:

$$(d_i, d_{i-1}) \leftarrow (d_i + 1, d_{i-1} - 2^w).$$

This equation can be written in the constant-time (applicable to d_i of any parity) form:

$$(d_i, d_{i-1}) \leftarrow (d_i + \mathtt{even}(d_i), d_{i-1} - \mathtt{even}(d_i)2^w).$$

Here $\texttt{even}(d_i) = 1 - d_i \bmod 2$.

Third, if the scalar d is even, then it is replaced with the odd scalar $q - d$. We compute the point $(q-d)P = -dP$ and negate it afterwards.

Altogether, we get the following algorithm.

Algorithm `ScalarMult`

Input: $P \in \mathbb{G} \setminus \{O\}$, $d \in \{1, 2, \ldots, q-1\}$.

Output: dP (in affine coordinates).

Steps:

1. $\delta \leftarrow d \bmod 2$, $d \leftarrow (1-\delta)q + (2\delta - 1)d$.
2. Choose a window width w ($3 \le w < \log_2 q$).
3. $P[1] \leftarrow P$, $(P[3], P[5], \ldots, P[2^w - 1]) \leftarrow \texttt{alg}(P, w)$, $\texttt{alg} \in \{\texttt{SmallMultJ}, \texttt{SmallMultA}\}$.
4. $(P[-1], P[-3], \ldots, P[-2^w + 1]) \leftarrow (-P[1], -P[3], \ldots, -P[2^w - 1])$.
5. Represent d as $\sum_{i=0}^{k-1} d_i 2^{wi}$, $d_0, d_1, \ldots, d_{k-1} \in \{0, 1, \ldots, 2^w - 1\}$.
6. $(d_{k-1}, d_{k-2}) \leftarrow (d_{k-1} + \texttt{even}(d_{k-1}), d_{k-2} - \texttt{even}(d_{k-1})2^w)$.
7. $Q \leftarrow P[d_{k-1}]$.
8. For $i = k-2, k-3, \ldots, 1$:
   (1) $(d_i, d_{i-1}) \leftarrow (d_i + \texttt{even}(d_i), d_{i-1} - \texttt{even}(d_i)2^w)$;
   (2) $Q \leftarrow 2^w Q$ ($J \leftarrow 2J$, w times);
   (3) $Q \leftarrow Q + P[d_i]$ ($J \leftarrow J + J$ or $J \leftarrow J + A$).
9. $Q \leftarrow 2^w Q$.
10. $Q \leftarrow Q + P[d_0]$.
11. $Q \leftarrow (-1)^\delta Q$.
12. Convert Q to affine coordinates ($A \leftarrow J$).
13. Return Q.

Note that in doublings $J \leftarrow 2J$ and additions $J \leftarrow J + J$ or $J \leftarrow J + A$, the exceptional case when some of the operands equals O is not possible. This is because the condition $Q \ne O$ is an invariant of `ScalarMult`. Let us prove this fact.

First, Q is obviously not equal to O after Steps 7, 10, 11. Second, Q cannot become O after doubling since Q belongs to the group $\mathbb{G}$ of odd order q. Third, Q cannot become O after Step 8.3. Indeed, after this step the point Q has the form eP, where $e = \sum_{j=i}^{k-1} d_j 2^{w(j-i)}$. The equality $Q = O$ means that either $e = 0$ or $e = q$. The case $e = 0$ is impossible because e is odd. The case $e = q$ is impossible since $i \ge 1$ and, therefore, $e < 2^{w(k-1)} < q$.

Conventional algorithms for doubling and adding points on Weierstrass curves are constant-time provided that exceptional cases are not possible.

Using these algorithms in `ScalarMult`, we achieve for it the constant-time property. We take into account that `ScalarMult` does not contain conditional branches and that the nested algorithms `SmallMultJ` and `SmallMultA` are constant-time.

It should be noted that in addition to conditional branches, there is another factor of non-constant running time. Modern microprocessors load data through the cache memory and the loading time may vary depending on the cache state. Since `ScalarMult` does use the array $P[i]$ to store small multiples, additional measures should be taken to prevent running time fluctuations.

One of the natural measures is to avoid storing negative points $P[-n]$ by switching from the operations $J \leftarrow J+J$, $J \leftarrow J+A$ to $J \leftarrow J \pm J$ and $J \leftarrow J \pm A$ while preserving the constant-time property. In result, `ScalarMult` requires either $3 \cdot 2^{w-2}$ or $2 \cdot 2^{w-2}$ field registers to store Jacobian or affine small multiples $P[n]$, $n = 1, 3, \ldots, 2^{w-1}$, respectively.

Let `ScalarMult`$[$`alg`$, w]$ be the algorithm `ScalarMult` instantiated with `alg` $\in \{$`SmallMultJ`, `SmallMultA`$\}$ and a window width w. Let a curve with $a = -3$ be used and the operations $J \leftarrow 2J$, $J \leftarrow J + J$ and $J \leftarrow J + A$ be implemented with the complexity 3M+6S+4m+6A+`half`, 11M+5S+4m+9A and 7M+4S+4m+9A respectively.[1] Let the complexity of $A \leftarrow J$ be I+S+3M. The choice $a = -3$ made provides the fastest time for the operation $J \leftarrow 2J$ without affecting the complexity of other operations. Moreover, since a is small, multiplication by a in `alg` costs 1m, not 1M.

For the case `alg` = `SmallMultA`, let the cascade $(J \leftarrow 2J, J \leftarrow J + A)$ at the junction of the steps 8.2, 8.3 and 9, 10 be additionally optimized. The cascade is treated as a special operation $J \leftarrow 2J + A$ which is implemented in time 11M + 7S + 27A according to [11, Appendix A.3].

In these settings, the final algorithms have the following complexity:

Operation in $\mathbb{F}$	`ScalarMult`$[$`SmallMultJ`$, w]$	`ScalarMult`$[$`SmallMultA`$, w]$
I	1	2
M	$\lceil l/w-1 \rceil (3w+11) + 19 \cdot 2^{w-2} - 8$	$\lceil l/w-1 \rceil (3w+8) + 25 \cdot 2^{w-2} - 10$
S	$\lceil l/w-1 \rceil (6w+5) + 7 \cdot 2^{w-2}$	$\lceil l/w-1 \rceil (6w+1) + 5 \cdot 2^{w-1} - 2$
m	$\lceil l/w-1 \rceil (4w+4) + 8$	$\lceil l/w-1 \rceil (4w-4) + 8$
A	$\lceil l/w-1 \rceil (6w+9) + 2 \cdot 2^{w+2} - 3$	$\lceil l/w-1 \rceil (6w+21) + 19 \cdot 2^{w-2} - 6$
`half`	$\lceil l/w-1 \rceil w + 3 \cdot 2^{w-2} - 3$	$\lceil l/w-1 \rceil (w-1) + 2^w - 4$

To simplify the expressions above, let us apply the following heuristic often used in practice:

$$\mathtt{I} = 100\mathtt{M}, \quad \mathtt{S} = 0.8\mathtt{M}, \quad \mathtt{m} = \mathtt{A} = \mathtt{half} = 0\mathtt{M},$$

[1] See [1] for detailed circuits. They are named `dbl-1998-hnm`, `add-2007-bl` and `madd-2007-bl`.

or $(100, 0.8, 0)$ for short. Let us also consider two additional heuristics: $(100, 0.67, 0)$ and $(100, 0.67, 0.05)$. With a heuristic h, the complexity of an algorithm is expressed as the number of M operations, *M-complexity*, denoted as $\mathtt{M}(h)$.

Table 2 presents M-complexity of the algorithms $\mathtt{ScalarMult}[\mathtt{alg}, w^*]$ for $l = 256, 384, 512$ and different heuristics listed above. Here w^* is the window width that provides the smallest M-complexity for a given alg and l. The optimal width w^* is the same for all three heuristics. The table shows that the choice $\mathtt{alg} = \mathtt{SmallMultA}$ is preferable to $\mathtt{alg} = \mathtt{SmallMultJ}$ providing both smaller M-complexity and smaller optimal window width. Note that the smaller the window width, the less memory is required.

Table 2: M-complexity of scalar multiplication algorithms

l	Algorithm	M-complexity (rounded to the nearest integer)		
		$\mathtt{M}(100, 0.8, 0)$	$\mathtt{M}(100, 0.67, 0)$	$\mathtt{M}(100, 0.67, 0.05)$
256	$\mathtt{ScalarMult}[\mathtt{SmallMultJ}, 5]$	3043	2803	2985
	$\mathtt{ScalarMult}[\mathtt{SmallMultA}, 4]$	2840	2631	2824
	$\mathtt{MontLadder}[\mathtt{WeierCurve}]$	2724	2590	2731
	$\mathtt{MontLadder}[\mathtt{MontCurve}]$	2200	2067	2182
384	$\mathtt{ScalarMult}[\mathtt{SmallMultJ}, 6]$	4379	4029	4293
	$\mathtt{ScalarMult}[\mathtt{SmallMultA}, 5]$	4085	3769	4048
	$\mathtt{MontLadder}[\mathtt{WeierCurve}]$	4030	3829	4041
	$\mathtt{MontLadder}[\mathtt{MontCurve}]$	3250	3050	3223
512	$\mathtt{ScalarMult}[\mathtt{SmallMultJ}, 6]$	5739	5271	5622
	$\mathtt{ScalarMult}[\mathtt{SmallMultA}, 5]$	5328	4907	5278
	$\mathtt{MontLadder}[\mathtt{WeierCurve}]$	5335	5068	5350
	$\mathtt{MontLadder}[\mathtt{MontCurve}]$	4299	4033	4264

The table additionally covers the algorithm from [10] that performs scalar multiplication on Montgomery curves by the left-to-right Montgomery ladder according to [13]. This algorithm denoted as $\mathtt{MontLadder}[\mathtt{MontCurve}]$ is considered one of the most efficient in the constant-time class, its complexity is

$$l(5\mathtt{M} + 4\mathtt{S} + \mathtt{m} + 8\mathtt{A}) + \mathtt{I} + \mathtt{M}.$$

Here we treat multiplication by a (presumably small) curve coefficient as the operation m and multiplication by a coordinate of P as the operation M (the latter because scalar multiplication is performed in the variable base settings).

The table also covers the algorithm from [5] which we denote as $\mathtt{MontLadder}[\mathtt{WeierCurve}]$. This algorithm also uses the Montgomery ladder

but over short Weierstrass curves. This is probably the fastest constant-time algorithm for these curves, its complexity is

$$l(7\mathtt{M} + 4\mathtt{S} + 10\mathtt{A} + 1\mathtt{half}) + \mathtt{I} + 8\mathtt{M} + 6\mathtt{S} + 3\mathtt{m} + 6\mathtt{A} + 2\mathtt{half}.$$

We suppose here that the simple finalization technique [5, Section 2.4.1] and the S-M tradeoff [5, Figure 3] are used. We also suppose that the constant 1/3 is precomputed so that division by 3 costs 1M.

The table shows that the algorithm ScalarMult[SmallMultA, w^*] is competitive to MontLadder[WeierCurve] especially for large l. A drawback of ScalarMult[SmallMultA, w^*] is a rather large memory requirements compared to only 6 field registers

Conclusion

Combining division polynomial-driven algorithms for small scalar multiplication on elliptic curves in the short Weierstrass form with several well-known optimization techniques we obtain constant-time algorithms for scalar multiplication that are competitive to the recent developments in the subject based on the Montgomery ladder. The integrated techniques are: the window method, skipping even small multiples, left-to-right scalar recording avoiding exceptional cases, Montgomery's trick for simultaneous inversion of several field elements, the fast point doubling-addition.

References

[1] Bernstein D. J., Lange T., "Explicit-formulas database", 2007, http://hyperelliptic.org/EFD.

[2] Bernstein D. J., Lange T., "Faster addition and doubling on elliptic curves", *Lecture Notes in Computer Science*, Advances in Cryptology – ASIACRYPT 2007, **4833**, ed. Kurosawa K., Springer, Berlin, Heidelberg, 2007, 29–50.

[3] Bernstein D. J., Birkner P., Joye M., Lange T., Peters C., "Twisted Edwards curves", *Lecture Notes in Computer Science*, Advances in Cryptology – AFRICACRYPT 2008, **5023**, ed. Vaudenay S., Springer, Berlin, Heidelberg, 2008, 389–405.

[4] Edwards H. M., "A normal form for elliptic curves", *Bulletin (New Series) of the American Mathematical Society*, **44**:3 (2007), 393–422.

[5] Hamburg M., "Faster Montgomery and double-add ladders for short Weierstrass curves", Cryptology ePrint Archive, Report 2020/437, 2020, https://eprint.iacr.org/2020/437.

[6] Hankerson, D. and Menezes, A. J. and Vanstone, S., "Guide to Elliptic Curve Cryptography", 2003.

[7] Joye M., Tunstall M., "Exponent Recoding and Regular Exponentiation Algorithms", *Lecture Notes in Computer Science*, Progress in Cryptology, Second International Conference on Cryptology in Africa — AFRICACRYPT 2009, **5580**, ed. Preneel B., Springer, Berlin, Heidelberg, 2008, 334–349.

[8] Kanayama N., Liu Y., Okamoto E., Saito K., Teruya T., Uchiyama S., "Implementation of an Elliptic Curve Scalar Multiplication Method Using Division Polynomials", *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, **E97.A**:1, 300–302.

[9] Koblitz N., "Elliptic Curve Cryptosystems", *Mathematics of computation*, **48**:177 (1987), 203–209.

[10] Langley A., Hamburg M., Turner S., "Elliptic Curves for Security", Request for Comments, **7748** (2016), https://rfc-editor.org/rfc/rfc7748.txt.

[11] Longa P., Miri A., "New Multibase Non-Adjacent Form Scalar Multiplication and its Application to Elliptic Curve Cryptosystems (extended version)", Cryptology ePrint Archive, Report 2008/052, 2008, https://eprint.iacr.org/2008/052.

[12] Miller V. S., "Use of Elliptic Curves in Cryptography", *Lecture Notes in Computer Science*, Advances in Cryptology – CRYPTO'85 Proceedings, **218**, ed. Williams H. C., Springer, Berlin, Heidelberg, 1986, 417-426.

[13] Montgomery P., "Speeding the Pollard and Elliptic Curve Methods of Factorization", *Mathematics of Computation*, **48**:177 (1987), 243–264.

[14] Okeya K., Takagi T., "The Width-w NAF Method Provides Small Memory and Fast Elliptic Scalar Multiplications Secure against Side Channel Attacks", *Lecture Notes in Computer Science*, Topics in Cryptology, The Cryptographers' Track at the RSA Conference — CT-RSA 2003, **2612**, ed. Joye, M., Springer, Berlin, Heidelberg, 2003, 328–342.

[15] Stange K. E., "The Tate Pairing Via Elliptic Nets", *Lecture Notes in Computer Science*, Pairing-Based Cryptography – Pairing 2007, **4575**, ed. Takagi T., Okamoto T., Okamoto E., Okamoto T., Springer, Berlin, Heidelberg, 2007, 329–348.

One "Short" Signature Scheme's Security Properties

Anton Guselev

Technical Committee on Standardisation "Cryptographic Information Security" (TC 026), Russia
guselev_am@tc26.ru

Abstract

At CTCrypt 2020 a digital signature scheme that allow to produce short digital signatures was presented. The scheme was made by the modification of one described in GOST R 34.10-2012 that is the variant of classical ElGamal framework. In the article the security of the scheme considered from the provable security point of view. However no practical variants to attack the scheme were presented in the article, the particular level of bit security was not estimated.

In this article we present three attacks that significantly reduce the security of the scheme. Characteristics of the attacks are used to estimate the bit security of the scheme.

Keywords: digital signature scheme, security level evaluation.

1 Introduction

At CTCrypt 2020 the group of authors proposed the description of the signature scheme (see [1]), that allows to form shorter signatures in comparison with the length of signatures, provided by the all variants of the signature schemes from GOST R 34.10-2012.

As pointed out in [1], the developers were guided by the need to use the features of the currently available software (hardware) used for the implementation of signature schemes from GOST R 34.10-2012, such as modern processors architecture or parameters of the random number generators used in generating the private key or one-time (ephemeral) keys.

The proposed scheme is based on the scheme from GOST R 34.10-2012, which implies a hash function with a hash code length of 256 bits. In further sections notations from[1] and GOST R 34.10-2012 will be used, stated base scheme further will be addressed as "GOST signature scheme".

In [1] three transformations are suggested to reduce the signature length:

– reduction of the length of the signature component r by applying a compressive mapping;

– reduction of the length of the signature component r due to fixation of some number of least significant bits. In this case some additional calculations (enumeration of values k) are supposed to be done in order to choose the value r that meets specified requirements;

– reduction of the length of the signature component s (or r) by “cutting off” some least bits. In this case some calculations to restore the initial value s (or r) from the “cuted” value are to be done.

1.1 Our contribution

It is obvious, that the mentioned transformations lead to decreasing of the security level of the initial GOST signature scheme. The theoretical security of the “short” signature scheme is justified in [1]. At the same time the actual assessment of the scheme security level is not carried out in the work. In this article some methods of analysis of the “short” signature scheme are presented. The methods characteristics will be used to estimate the actual security level of the scheme.

1.2 Paper organisation

The remainder of the paper is organized as follows. In Section 2 basic definitions and descriptions of the schemes from [1] are introduced. In Section 3 security evaluations of the described schemes are considered. Novel attacks with the evaluation of their characteristic are presented in the section. The attack characteristics are used to evaluate security level of the schemes from [1]. All the conclusions drawn up in Section 4.

2 Design principles

Before proceeding to the description of the design principles used in [1] to create the «short» signature scheme, let us recall the general principles of operation of the GOST signature scheme.

The following notions will be used. If the variable x gets the value of the variable y we denote $x \leftarrow y$. To show that the value is chosen uniformly at random from the set A and this value is assigned with veriable k we use the notion $k \xleftarrow{\mathcal{U}} A$. For $x \in \mathbb{Z}_q$ the notion $\bar{x}$ will mean the bit representation of x. For point C of some elliptic curve, the notion x_C denotes x-coordinate of

point C. Also some hash-function $\mathrm{H}(\cdot)$ will require to represent the signature schemes.

Each user of the signature scheme has private key d and the corresponding public key $Q = dP$, where P is an element of a subgroup of prime order q (where q is n bit length) of points of elliptic curve given over $\mathbb{Z}_p$, where p is some prime number.

The following algorithms should be implied to sign the message $\mathtt{Msg} \in V^*$ and to verify the signature.

Generic sign algorithm

Input: $(d, \mathtt{Msg})$

Output: signature $\bar{r}\|\bar{s}$

1: $e \leftarrow \mathrm{H}(\mathtt{Msg})$
2: $k \xleftarrow{\mathcal{U}} \mathbb{Z}_q$
3: $r \leftarrow x_{kP} \pmod q$
4: $s \leftarrow ke + rd$
5: **return** $\bar{r}\|\bar{s}$

Generic verification algorithm

Input: $(Q, \mathtt{Msg}, \bar{r}\|\bar{s})$

Output: $\{0, 1\}$

1: $e \leftarrow \mathrm{H}(\mathtt{Msg})$
2: $R \leftarrow e^{-1}sP - e^{-1}rQ$
3: **if** $x_R \neq r$ **then**
4: **return** 0 ▷ The signature is false
5: **else**
6: **return** 1 ▷ The signature is correct

2.1 One-way function based approach

The basic idea of the first method from [1] is to reduce the bit length of $\bar{r}$ by applying some one-way function $f : \mathbb{Z}_p \to \mathbb{Z}_q^*$ to r. According to [1] $f(r) = \phi(\mathrm{H}_1(\bar{r}) \pmod{2^b})$ is one-way function, where $\mathrm{H}_1 : V^* \to \mathbb{Z}_{2^{256}}$ and $\phi : \mathbb{Z}_{2^b} \to \mathbb{Z}_q^*$. Note that $f(r)$ has only b significant bits for any r.

In this case the following algorithms should be implied to sign the message $\mathtt{Msg}$ and to verify the signature. This signature scheme further will be addressed as Scheme 1.

Sign algorithm

Input: $(d, \mathtt{Msg})$

Output: signature $\bar{r}\|\bar{s}$

1: $e \leftarrow \mathrm{H}(\mathtt{Msg})$
2: $k \xleftarrow{\mathcal{U}} \mathbb{Z}_q$.
3: $r \leftarrow f(x_{kP} \pmod q)$
4: $s \leftarrow ke + rd$
5: **return** $\bar{r}\|\bar{s}$

Verification algorithm

Input: $(Q, \mathtt{Msg}, \bar{r}\|\bar{s})$

Output: $\{0, 1\}$

1: $e \leftarrow \mathrm{H}(\mathtt{Msg})$
2: $R \leftarrow e^{-1}sP - e^{-1}rQ$
3: **if** $f(x_R) \neq r$ **then**
4: **return** 0 ▷ The signature is false
5: **else**
6: **return** 1 ▷ The signature is correct

In [1] it is noted that bit length of the signature will be $n + b + 1$.

Note that the idea of reduction of the signature length by applying a one-way function to the signature component was proposed by C.P. Schnorr in 1989 [3]. One of the main features of the signature scheme proposed in [3] that its security level can't be reduced considering collision attacks against

hash-functions in use. It is important to notice that the scheme from [1] has no such feature.

2.2 Bit-fixation based approach

The basic idea of the second method from [1] is to use the representation $\bar{r} = \bar{r}^* \| const$, where the bit length of the string $const$ is predefined value l. In this case calculation of the value $\bar{r}$ require additional computations. During the computations the enumeration of the values $k \in \mathbb{Z}_q$ are to be made.

The following algorithms should be implied to sign the message `Msg` and to verify the signatures.

Sign algorithm

Input: $(d, \texttt{Msg})$

Output: signature $\bar{r} \| \bar{s}$

1: $e \leftarrow \mathrm{H}(\texttt{Msg})$
2: $k \xleftarrow{\mathcal{U}} \mathbb{Z}_q$
3: $r \leftarrow x_{kP} \pmod q$
4: **if** $\mathrm{LSB}_l(\bar{r}) \neq const$ **then**
5: **goto** step 1
6: **else**
7: $s \leftarrow ke + rd$
8: $\bar{r}^* \leftarrow \mathrm{MSB}_{n-l}(\bar{r})$
9: **return** $\bar{r}^* \| \bar{s}$

Verification algorithm

Input: $(Q, \texttt{Msg}, \bar{r} \| \bar{s})$

Output: $\{0, 1\}$

1: $e \leftarrow \mathrm{H}(\texttt{Msg})$.
2: $\bar{r} \leftarrow \bar{r}^* \| const$
3: $R \leftarrow e^{-1}sP - e^{-1}rQ$
4: **if** $x_R \neq r$ **then**
5: **return** 0 ▷ The signature is false
6: **else**
7: **return** 1 ▷ The signature is correct

In [1] it is noted that the bit length of the signature will be $2n - l$.

2.3 Bit-cutting based approach

The third approach proposed in [1] is to cut some bits of the parameter s (or r). In this case, during the verification of the signature it is necessary to perform additional computations in order to restore and check the truncated bits.

Let’s assume that the bit length of the parameter s is truncated by t bits. The following algorithms should be implied to sign the message `Msg` and to verify the signature.

Sign algorithm

Input: $(d, \texttt{Msg})$

Output: signature $\bar{r}\|\bar{s}$

1: $e \leftarrow \mathrm{H}(\texttt{Msg})$
2: $k \xleftarrow{\mathcal{U}} \mathbb{Z}_q$
3: $r \leftarrow x_{kP}$
4: $s \leftarrow ke + rd$
5: $\bar{s}^* \leftarrow \mathrm{MSB}_{n-t}(\bar{s})$
6: **return** $\bar{r}\|\bar{s}$

Verification algorithm

Input: $(Q, \texttt{Msg}, \bar{r}\|\bar{s})$

Output: $\{0, 1\}$

1: $e \leftarrow \mathrm{H}(\texttt{Msg})$
2: $cnt \leftarrow 0$
3: $cnt \leftarrow cnt + 1.$
4: **if** $cnt \geq 2^t$ **then**
5: **return** 0 ▷ The sign is false
6: **else**
7: $\bar{s} \leftarrow \bar{s}^*\|\overline{cnt}$
8: $R \leftarrow e^{-1}sP - e^{-1}rQ$
9: **if** $x_R = r$ **then**
10: **return** 1 ▷ The sign is correct
11: **else**
12: **goto** step 3

In [1] it is noted that the bit length of the signature will be $2n - t$.

2.4 Combined approach

To achieve the minimal length of the signature in [1] it is suggested to combine the three approaches. In this case the sign and verification algorithms could be denoted as follows.

Sign algorithm

Input: $(d, \texttt{Msg})$

Output: signature $\bar{r}\|\bar{s}$

1: $e \leftarrow \mathrm{H}(\texttt{Msg}).$
2: $k \xleftarrow{\mathcal{U}} \mathbb{Z}_q$
3: $r \leftarrow f(x_{kP} \pmod q)$
4: **if** $\mathrm{LSB}_l(\bar{r}) \neq const$ **then**
5: **goto** step 1
6: **else**
7: $s \leftarrow ke + rd.$
8: $\bar{r}^* \leftarrow \mathrm{MSB}_{n-l}(\bar{r})$
9: $\bar{s}^* \leftarrow \mathrm{MSB}_{n-t}(\bar{s})$
10: **return** $\bar{r}\|\bar{s}$

Verification algorithm

Input: $(Q, \texttt{Msg}, \bar{r}\|\bar{s})$

Output: $\{0, 1\}$

1: $e \leftarrow \mathrm{H}(\texttt{Msg})$
2: $\bar{r} \leftarrow \bar{r}^*\|const$
3: $cnt \leftarrow 0$
4: $cnt \leftarrow cnt + 1$
5: **if** $cnt \geq 2^t$ **then**
6: **return** 0 ▷ The sign is false
7: **else**
8: $\bar{s} \leftarrow \bar{s}^*\|\overline{cnt}$
9: $R \leftarrow e^{-1}sP - e^{-1}rQ$
10: **if** $f(x_R) = r$ **then**
11: **return** 1 ▷ The sign is correct
12: **else**
13: **goto** step 4

This signature scheme further will be addressed as Scheme 2.
In [1] it is noted that the bit length of the signature will be $n+b+1-t-l$.

3 Security evaluations

In [1] security of the “short” signature scheme is considered from the provable security point of view. The main results are based upon the approach

provided during the justification of security of ECDSA in [2]. In particular, it is shown that the proposed signature scheme is secure the discrete logarithm problem is hard for the case in consideration. A common approach based on forking lemma [4] is considered to achieve the final results. It is also noted in [1] that when the parameters $b = 100$, $t = 18$ and $l = 18$ are used, the advantage of the adversary to forge the signature does not exceed 2^{-35}.

3.1 Novel attacks

A common attack approach against any signature schemes based upon collision attacks against hash-function that are in use for message processing. The application of this approach to analyse the proposed “short” signature scheme will not lead to a reduction of the security level in comparison to the GOST signature scheme.

At the same time, the features of the signature length reduction methods described above can be used to mount additional attacks.

3.1.1 Attack against Scheme 1

The aim of the attack is to find a preimage for the one-way function for the fixed value of r. This attack could be used to forge a signature for an arbitrary message. The adversary needs to have **only one** message/signature pair computed by a legitimate user to mount the attack. Let’s assume that the user computed the signature $\bar{r}\|\bar{s}$ for the message $\mathtt{Msg}$. So, in order to forge a signature for an arbitrary message $\mathtt{Msg}_1$ without the private key d the following algorithm could be performed.

Algorithm 1 Attack against Scheme 1

Input: $(\mathtt{Msg}, \mathtt{Msg}_1, \bar{r}\|\bar{s})$
Output: forged signature $\bar{s}_1\|\bar{r}$

$e \leftarrow \mathrm{H}(\mathtt{Msg})$
$e_1 \leftarrow \mathrm{H}(\mathtt{Msg}_1)$
$T \leftarrow e^{-1}sP - e^{-1}rQ.$ ▷ $T = kP$, but k is unknown
$i \leftarrow 0$
while $f(x_A) \neq r$ **do**
    $i \leftarrow i + 1$
    $A = e_1^{-1}eT + e_1^{-1}eiP$
$s_1 \leftarrow s + ie$
return $\bar{r}\|\bar{s}_1$

Let us show that the forged signature $\bar{s}_1\|\bar{r}$ will be valid signature for $\mathtt{Msg}_1$. To do this, let us demonstrate that the equality $f(x_R) = r$ holds. Note

that

$$s_1 = s + ie = ke + rd + ie = e(k + i) + rd,$$

then

$$R = e_1^{-1}s_1P - e_1^{-1}rQ = e_1^{-1}(e(k + i) + rd)P - e_1^{-1}rdP = \\ = e_1^{-1}ekP + e_1^{-1}eiP = A.$$

Given the equation $f(x_A) = r$, we obtain that $f(x_R) = r$, i.e. the forged signature is valid.

Note that computation the second preimage in the case of $b = 100$ would require no more than 2^{100} calculations of the multiple points and the hash-functions.

3.1.2 An attack based on bit fixation

Let’s consider Scheme 2 in which the bit-cutting procedure (see sec. 2.3) is not implied. The signature is achieved only by the implementation of one-way function and by fixing of some bits.

In this case the set of all values r suitable for the calculation of the signature may be reduced. We may find values r that are different from the one used during the sign stage, but they may be used during verification procedure and signature will be valid. In particular, lets consider a signature $\bar{r}^*\|\bar{s}$, where

$$r = x_{kP}, \quad \bar{r} = \bar{r}^*\|\underbrace{0\ldots0}_{l},$$

for some k that was specially chosen (by the user) during the sign procedure. Then if we choose such k' that $x_{k'P} = r'$, $\bar{r}' = \bar{r}^*\|\bar{u}$, where $0 < u < 2^l$, then in view of the procedure of cutting least significant bits of r, the signature $(\bar{r}^*\|\bar{s})$ proceeded from r', and not from r, will also be a valid signature for the given message. It is should be noted that the described method will achieve success in case the comparison of cut bits with given constant is not performed during the verification procedure.

Let’s consider the representation $\bar{r}' = \bar{r}^*\|\bar{u}$, where $0 < u < 2^l$. Having the last inequality we may state that the cardinality of the set of admissible values r suitable for the analysis increase from 1 to $2^l - 1$. So the probability to find i useful for the attack increases.

Let’s describe how to apply the mentioned idea on practice.

Let’s assume that we want to forge a signature for the message $\mathtt{Msg}_1$ such that $\mathrm{H}(\mathtt{Msg}_1) = e_1$. To do so we imply the enumeration method from 3.1.1 to

find the second preimage for r. Let i be the value such that $f(x_{e_1^{-1}e(k+i)P}) = r'$. Then, if we apply the procedure described in section 3.1.1 to modify the parameter s and obtain the value s_1, the signature $(r^* \| s_1)$ will be a valid signature for the message $\mathtt{Msg}_1$. That could be proven by the following equations:

$$e_1^{-1}s_1P - e_1^{-1}rQ = e_1^{-1}(e(k+i) + rd)P - e_1^{-1}dP = A,$$
$$f(x_A) = r'.$$

So if during the verification procedure l least significant bits of r' are not compared with the given constant the attack will succeed.

Thus, in the case of fixation the 18 least significant bits with zeros (i.e., $l = 18$), as proposed in [1], we will require no more than $2^{100-18} = 2^{82}$ calculations of the multiple points and the hash-functions to realise the attack.

3.1.3 An attack based on bit-cutting

Let’s consider Scheme 2 in which the bit fixation procedure (see sec. 2.2) is not implied. The signature is achieved only by the implementation of one-way function and bit-cutting.

Since the sign procedure based upon the equation $s = ke + rd$ where the least significant bits of $\bar{s}$ being further truncated, it is generally possible to influence the least significant bits in a way the signature remains valid. Let $\bar{j} \in \{0,1\}^t$, such that $\mathrm{LSB}_t(s) = \mathrm{LSB}_t(s + j)$. Let’s assume $s_2 = s + j$. Considering equations

$$e^{-1}s_2P - e^{-1}rQ = e^{-1}(ke + rd + j)P - e^{-1}rdP = kP + e^{-1}jP,$$
$$f(x_{(k+e^{-1}j)P}) = r_j,$$

we will get that for any value j that does not affect bits of s that are part of the signature, we can compute values r_j that are “equivalent” to the “correct” value r (that is part of the signature). In this case, for some j we may state that the verification procedure executes

$$f(x_R) = r_j.$$

Then by going through different values of the least significant bits of s for some j we will obtain the equality $f(x_R) = r_j$, indicating that the verified signature is valid.

The described feature may be used on practice as follows. When searching for the second preimage using the method described on 3.1.1, success will

be achieved not only when the tested equality is satisfied for the specified value r, but also for any value r_j. According to [1] there are 2^t possible values of j, and it is suggested to use $t = 18$. Thus, in order to find the second preimage for the one-way function, it is necessary to perform about $2^{100-18} = 2^{82}$ calculations of the multiple points and the hash-functions to realise the attack.

3.1.4 Attack against Scheme 2

If the attacks described in 3.1.2 and 3.1.3 will be realised simultaneously, security level of the «short» signature scheme may be estimated by a value of $2^{100-19} = 2^{81}$. The value is obtained because the cordiality of the set of favourable events at which one of the two given attacks could be mount is $2^l + 2^t$ that with $t = l = 18$ is 2^{19}.

4 Conclusions

Using the parameters recommended in [1], forging a signature would require about 2^{81} calculations of the multiple points and the hash-functions. In this case, we can say that for such a parameter size, the proposed signature scheme, has (by rough estimates) a security level equal to 81 bits. To achieve this security level in the scheme based on ElGamal framework, it is possible to use a private key of 162 bits. For example the non-modified signature scheme from GOST R 34.10-2012 with such size of private key will give a signature of 324 bits length. In [1] states that for a given set of parameters, the scheme will be secure (in particular model) if the signature is 320 bits and the private key is 256 bits, considering additional calculation should be made during the sign and verification.

Thus, for realisation of a “short” signature scheme require the private key of 256 bit length the bit security level of the scheme almost the same as in GOST scheme with truncated parameters, but the realisation of the scheme require additional computational evaluations. So we can conclude that the practical significance of the proposed in [1] modifications of sign and verification procedures defined in GOST R 34.10-2012 aimed to shorten the signature size needs further considerations.

References

[1] L. Akhmetzyanova, E. Alekseev, A. Babueva, and S. Smyshlyaev, “On Methods of Shortening ElGamal-type Signatures”, Pre-proceedings 9th Workshop on Current Trends in Cryptology

(CTCrypt 2020), **9** (2020), 251-286.

[2] M. Fersch, E. Kiltz, and B. Poettering, “On the provable security of (EC)DSA signatures”, Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016, 1651-1662.

[3] C. Schnorr, “Efficient Identification and Signatures for Smart Cards”, *LNCS*, CRYPTO’89, **435**, ed. G. Brassard, 1989, 239–252.

[4] D. Pointcheval and J. Stern, “Security arguments for digital signatures and blind signatures”, *Journal of Cryptology*, **13(3)**, 2000, 361-396.