8th Workshop on Current Trends in Cryptology (CTCrypt 2019)

June 4-7, 2019, Svetlogorsk, Kaliningrad region, Russia.

Pre-proceedings

In cooperation

— General partner —

infotecs

— Official partner —

КРИПТОПРО

— Partners —

КОД БЕЗОПАСНОСТИ

КОМПАНИЯ АКТИВ

— Support —

— Media partners —

CTCrypt 2019 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

Mikhail Glukhov	–	Academy of Cryptography of the Russian Federation, Russia
Andrey Zubkov	–	Steklov Mathematical Institute of RAS, Russia
Andrey Pichkur	–	Federal Educational and Methodical Association in System of Higher Education on Information Security, Russia
Dmitry Matyukhin	–	TC 026, 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
Tomer Ashur	–	Katholieke Universiteit Leuven, Belgium
Jean-Philippe Aumasson	–	Taurus Group SA, Switzerland
Sergey Checheta	–	Federal Educational and Methodical Association in System of Higher Education on Information Security, Russia
Ivan Chizhov	–	Lomonosov Moscow State University, Russia
Yury Kharin	–	Research Institute for Applied Problems of Mathematics and Informatics, Belarus
Igor Kruglov	–	Academy of Cryptography of the Russian Federation, Russia
Grigory Marshalko	–	TC 026, Russia
Dmitry Murin	–	P.G. Demidov Yaroslavl State University, Russia
Thomas Peyrin	–	Nanyang Technological University, Singapore
Eduard Primenko	–	Lomonosov Moscow State University, Russia
Boris Ryabko	–	Institute of Computational Technologies SB RAS; and Novosibirsk State University, Russia
Markku-Juhani Olavi Saarinen	–	Independent expert on information security, Finland
Igor Semaev	–	The University of Bergen, Norway
Vasily Shishkin	–	TC 026, Russia
Stanislav Smyshlyaev	–	TC 026, Russia
Alexey Tarasov	–	Federal Educational and Methodical Association in System of Higher Education on Information Security, Russia
Andrey Trishin	–	«Certification Research Center» Ltd., Russia
Amr Youssef	–	Concordia University, Canada
Andrey Zyazin	–	Russian Technological University (MIREA), Russia

External Reviewers

Aleksandr Semenov, Alexey Nesterenko, Alexey Tatuzov, Andrey Ivanov, Atul Luykx, Boltnev Yury, Denis Vasilyev, Dmitry Bagin, Ekaterina Malygina, Elena Kirshanova, Evgeny Alekseev, Grigory Karpunin, Igor Oshkin, Igor Sergeev, Liliya Akhmetzyanova, Liliya Kraleva, Luca De Feo, Michael

Koypish, Mohamed Tolba, Muhammad Elsheikh, Raluca Posteuca, Rinat Shakirov, Sergey Gashkov, Siemen Dhooghe, Vasily Nikolaev, Victor Markov, Vladimir Chubarikov, Vladimir Mironkin, Yu Long Chen

Dear colleagues!

This year the Workshop "Current Trends in Cryptology" will open its doors for information protection specialists and everyone interested in the subject for the 8th time.

This time the unbiased international program committee reviewed 37 papers submitted by representatives of 6 countries. Coming in second in the Workshop history, this number is almost twice as much as it was last year. The increase indicates the growth of the academic community understanding of information security importance, that happened including on the ground of Russian government course for digitalization in all aspects of life and activities of individuals, society and the state. 22 papers selected after the reviewing will be presented to you at the Workshop. There are different kind of topics to cover including analysis and design of classical block and stream cryptographical mechanisms as well as widely discussed post-quantum cryptographic protocols which can save their cryptographic characteristics even if an appropriate quantum computer is created. Practical applications of cryptography also will not be left out of attention. A wide range of subjects affirms that each of 128 participants from 8 countries will find something interesting and worth to be used in scientific and practical activity.

This year the program committee decided to carry out an experiment and added the scientific program with cryptography lectures for information security specialists and high-schoolers. In the former case the lectures will be delivered by leading experts of cryptographic devices developers and technical committee for standardization "Cryptography and security mechanisms" (TC 026), in the latter case – by the professors of a leading Russian university graduating cryptography specialists. Two panel discussions are to be held during the Workshop. The first one will be dedicated to the first one in Russian Federation brand new cryptography museum, particularly to the ideas and plans its founders as well as the issues will have to be solved by them. The second one will deal with the work of the Academy of Cryptography of the Russian Federation laboratory on standardization problems in cryptography and information security which was created as a part of national program "Digital Economy of the Russian Federation".

Covering the trends in cryptology and contributions in them by outstanding academia community representatives is one of distinguished characteristics of the Workshop that is also highlighted in its title. This is achieved including

by enlarging the program with invited talks by leading Russian and foreign specialists. This year there will be three such talks. The first one by Andrey Pichkur and Alexey Tarasov will be devoted to the Mikhail M. Glukhov passed away last year who was an academician of the Academy of Cryptography of the Russian Federation and is justifiably regarded as one of the founders of the Russian algebraic cryptographic school. We will also listen to Luca de Feo, one of the leading specialists in elliptical curves isogeny post-quantum cryptography, and Kenneth Paterson who will talk on a new version of TLS protocol. Dear colleagues, we are facing four days of effective work which in result, I hope, will let us expand the horizons of our knowledge and apply achieved information in scientific researches and projects.

Thereon I would like to declare the workshop "Current Trends in Cryptography" (CTCrypt 2019) open.

President of the Academy of Cryptography of the Russian Federation
Aleksandr Shoitov

Invited Talks

Introducing TLS 1.3

Kenneth Paterson

Applied Cryptography Group, Switzerland
kenny.paterson@inf.ethz.ch

Abstract

After a long gestation in the IETF TLS Working Group, work on TLS 1.3 was finally completed in 2018 with the publication of RFC 8446. In this talk, I'll explain how TLS 1.3 works, how it differs from earlier protocol versions, and why. I'll also reflect on the standardisation process which resulted in TLS 1.3.

Keywords: cryptographic protocol, standardisation, TLS 1.3.

How to prove a secret isogeny

Luca De Feo

Université de Versailles, France
luca.de-feo@uvsq.fr

Abstract

Isogenies of elliptic curves have proven to be a powerful tool to construct cryptographic protocols, in particular quantum-resistant ones.

The key encapsulation protocol SIKE is currently being considered for standardisation in the NIST post-quantum competition, while the younger primitive CSIDH is likely to find useful applications in more advanced protocols where a static-static key exchange is needed.

At present, the picture of isogeny-based signature protocols is much less bright. While it is known how to derive various identification schemes and signatures from both SIKE and CSIDH, they are all inefficient in some regard.

In this talk I will review the different protocols, both quantum-resistant and not, that have been devised to prove knowledge of a secret isogeny. I will explain their uses and limitations, report on ongoing work, and present some open questions.

Keywords: isogenies of elliptic curves, post-quantum cryptography, quantum-resistance, cryptographic protocol.

Contents

Algebraic and Probabilistic Aspects

On Isometric Mappings of the Set of All Boolean Functions into Itself Which Preserve Self-Duality and the Rayleigh Quotient

Aleksandr Kutsenko

Novosibirsk State University, Russia
alexandrkutsenko@bk.ru

Abstract

A bent functions is called self-dual if it equals to its dual. It is called anti-self-dual if it is equal to its complement. A mapping of the set of all Boolean functions in n variables into itself is said to be isometric if it preserves the Hamming distance. In this paper we study isometric mappings which preserve self-duality and anti-self-duality. The complete characterization of these mappings is obtained. Based on this result, the set of isometric mappings which preserve the Rayleigh quotient of a Boolean function is obtained. As a corollary all isometric mappings which preserve bentness and the Hamming distance between bent function and its dual are given.

Keywords: self-dual bent, Hamming distance, Isometric mapping, Rayleigh quotient.

1 Introduction

The term "bent function" was introduced by Oscar Rothaus in the 1960s in [13]. At the same time the maximally nonlinear Boolean functions were also under study in the Soviet Union. In 1962 the term *minimal function* which is in fact an analog of a bent function, was proposed by the Soviet scientists Eliseev and Stepchenkov, see [14].

Bent functions have applications in many domains, such as error correcting codes, spreading sequences for CDMA, and cryptology. In symmetric cryptography, due to maximal nonlinearity, these functions can be used as building blocks of stream and block ciphers in order to make them more resistant to main statistical methods of cryptanalysis among which are linear and differential cryptanalyses. Extensive information concerning bent functions can be found in monography of Tokareva [14].

A bent function that coincides with its dual is called self-dual. Open questions which are relevant to the class of bent functions are also relevant for the self-dual bent functions. A difficult problem is the complete characterization and description of the class of self-dual bent functions and estimation of its cardinality. There are a number of articles which are devoted to these and other problems. In particular, in the article [2] Carlet et al. explored self-dual bent functions: some symmetries, which preserve self-duality were given; it has been proved that the Hamming distance between a self-dual bent function and an anti-self-dual bent function in n variables is exactly 2^{n-1}. In [6] the classification of all quadratic self-dual bent functions is presented by Hou. Feulner et al. in [4] gave some new mappings which preserve self-duality. Some new constructions of bent functions both with their duals one can find in [12]. The upper bound for the cardinality of the set of self-dual bent functions which follows from the exact number of formally self-dual bent functions is presented by Hyun and Lee in [7]. The complete Hamming distance spectrum between self-dual Maiorana–McFarland bent functions was obtained in [8].

In current paper we study isometric mappings of the set of all Boolean functions in $n \geqslant 4$ variables into itself which preserve self-duality and anti-self-duality. The complete characterization of these mappings is obtained (Theorem 1). We also completely study isometric mappings which bijections between self-dual and anti-self-dual bent functions (Theorem 2). Based on this result, the set of isometric mappings which preserve the Rayleigh quotient of a Boolean

function is obtained (Corollary 1). All isometric mappings which preserve bentness and the Hamming distance between bent function and its dual are given.

2 Notations and definitions

Let $\mathbb{F}_2^n$ be a set of binary vectors of length n.

A *Boolean function* f in n variables is any map from $\mathbb{F}_2^n$ to $\mathbb{F}_2$. Its *sign function* is $F(x) = (-1)^{f(x)}$, $x \in \mathbb{F}_2^n$. Obviously we have $(-1)^{f(x)} = 1 - 2f(x)$ for any $x \in \mathbb{F}_2^n$. We will also refer to a sign function as to a vector from the set $\{\pm 1\}^{2^n}$: $F = (-1)^f = \left((-1)^{f_0}, (-1)^{f_1}, ..., (-1)^{f_{2^n-1}}\right) \in \{\pm 1\}^{2^n}$, where $(f_0, f_1, ..., f_{2^n-1}) \in \mathbb{F}_2^{2^n}$ is a truth-table representation of f with arguments given in the lexicographic order. The set of Boolean functions in n variables is denoted by $\mathcal{F}_n$.

The *Hamming weight* $\mathrm{wt}(x)$ of the vector $x \in \mathbb{F}_2^n$ is the number of nonzero coordinates of x. The *Hamming weight* $\mathrm{wt}(f)$ of the function $f \in \mathcal{F}_n$ is the Hamming weight of its vector of values. The sign $\oplus$ denotes a sum modulo 2. The *Hamming distance* $\mathrm{dist}(f, g)$ between Boolean functions f, g in n variables is a cardinality of the set $\{x \in \mathbb{F}_2^n : f(x) \oplus g(x) = 1\}$. For $x, y \in \mathbb{F}_2^n$ denote $\langle x, y \rangle = \bigoplus_{i=1}^{n} x_i y_i$. The *Walsh-Hadamard transform* (WHT) of the Boolean function f in n variables is an integer function $W_f : \mathbb{F}_2^n \to \mathbb{Z}$, defined as

$$W_f(y) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x) \oplus \langle x, y \rangle}, \quad y \in \mathbb{F}_2^n.$$

Let I_n be an 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

$$H_1 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

It is known the Hadamard property of this matrix

$$H_n H_n^T = 2^n I_{2^n}.$$

In [2] an orthogonal decomposition of $\mathbb{R}^{2^n}$ in eigenspaces of H_n was given:

$$\mathbb{R}^{2^n} = \mathrm{Ker}\left(H_n + 2^{n/2} I_{2^n}\right) \oplus \mathrm{Ker}\left(H_n - 2^{n/2} I_{2^n}\right),$$

where the symbol $\oplus$ denotes a direct sum of subspaces.

Denote $\mathcal{H}_n = 2^{-n/2} H_n$.

A Boolean function f in an even number n of variables is said to be bent if

$$|W_f(y)| = 2^{n/2}$$

for all $y \in \mathbb{F}_2^n$. The set of bent functions in n variables is denoted by $\mathcal{B}_n$.

In other words, the function f is bent if and only if for its sign function F it holds $\mathcal{H}_n F \in \{\pm 1\}^{2^n}$. From the definition above it follows that for any $y \in \mathbb{F}_2^n$ we have

$$W_f(y) = (-1)^{\widetilde{f}(y)} 2^{n/2}$$

for some $\widetilde{f} \in \mathcal{F}_n$.

The Boolean function $\widetilde{f}$ defined above is called the dual function of the bent function f.

If bent function f coincides with its dual it is said to be self-dual bent. A bent function which coincides with the negation of its dual is called an anti-self-dual bent. In [9] it was proved that within the set of sign functions of self-dual bent functions in $n \geqslant 4$ variables there exist a basis of the eigenspace of the matrix H_n attached to the eigenvalue $2^{n/2}$. The set of (anti-)self-dual bent functions in n variables, according to [6], is denoted by $\mathrm{SB}^+(n)$ $\big(\mathrm{SB}^-(n)\big)$.

A mapping φ of the set of all Boolean functions in n variables into itself is called an *isometric* mapping if it preserves the Hamming distance between functions, that is

$$\mathrm{dist}(\varphi(f), \varphi(g)) = \mathrm{dist}(f, g),$$

for any $f, g \in \mathcal{F}_n$. The set of all isometric mappings of the set of all Boolean functions in n variables into itself is denoted by $\mathcal{I}_n$.

The general form of isometric mappings is

$$f(x) \longrightarrow f(\pi(x)) \oplus g(x),$$

where π is a permutation on the set $\mathbb{F}_2^n$ and $g \in \mathcal{F}_n$ [11].

It is known [15] that every isometric mapping of the set of all Boolean functions into itself that transforms bent functions into bent functions is a combination of an affine transform of coordinates and an affine shift. The mapping $f \longrightarrow \widetilde{f}$ defined on the set of bent functions, preserves the Hamming distance [1] that is it is an isometric mapping of the set $\mathcal{B}_n$.

There is a one-to-one correspondence between $\mathcal{I}_n$ and the set of matrices of

order $2^n \times 2^n$ with elements from the set $\{0, \pm 1\}$ such that in every row (column) there is exactly one nonzero element. Indeed, let $\varphi : f(x) \longrightarrow f(\pi(x)) \oplus g(x)$, where π is a permutation on the set $\mathbb{F}_2^n$ and $g \in \mathcal{F}_n$. Then for any $f \in \mathcal{F}_n$ and its sign function $F \in \{\pm 1\}^{2^n}$ the sign function $F' \in \{\pm 1\}^{2^n}$ of $\varphi(f)$ can be expressed as $F' = AF$, where A is a $2^n \times 2^n$ matrix

$$
\begin{array}{c}
\\
v_i
\end{array}
\begin{array}{c}
\begin{array}{ccccccc}
 & & & \pi(v_i) & & & \\
 & & & 0 & & & \\
 & & & \vdots & & & \\
 & & & 0 & & & \\
0 & \dots & 0 & (-1)^{g(v_i)} & 0 & \dots & 0 \\
 & & & 0 & & & \\
 & & & \vdots & & & \\
 & & & 0 & & &
\end{array}
\end{array},
$$

in which in the row with number $(i+1) \in \{1, 2, ..., 2^n\}$ a nonzero element is in the $(j+1)$-th column, where j is a number with binary representation $\pi(v_i)$. The vector $v_k \in \mathbb{F}_2^n$ is a binary representation of the number $k \in \{0, 1, ..., 2^n - 1\}$.

Denote, according to [5], the orthogonal group of index n over the field $\mathbb{F}_2$ as

$$\mathcal{O}_n = \left\{ L \in GL(n, 2) | LL^T = I_n \right\},$$

where L^T denotes the transpose of L and I_n is an identical matrix of order n over the field $\mathbb{F}_2$.

3 Isometric mappings preserving self-duality

In [4] (Theorem 1) it was shown that the mapping

$$f(x) \longrightarrow f\left(L\left(x \oplus c\right)\right) \oplus \langle c, x \rangle \oplus d,$$

where $L \in \mathcal{O}_n$, $c \in \mathbb{F}_2^n$, $\mathrm{wt}(c)$ is even, $d \in \mathbb{F}_2$, preserves self-duality of a bent function. It is obvious that this mapping is an element from $\mathcal{I}_n$.

In this section we generalize this result within isometric mappings.

Proposition 1. *Let $n \geqslant 4$. Isometric mapping $\varphi \in \mathcal{I}_n$ with matrix A:*

— *preserves self-duality if and only if it preserves anti-self-duality;*

— *preserves self-duality if and only if $A\mathcal{H}_n = \mathcal{H}_n A$.*

Proof. In [9] it has been proved that for $n \geqslant 4$ within the set $\mathrm{SB}^+(n)$ there exist a subset $\{f_i\}_{i=1}^{2^{n-1}} \subset \mathrm{SB}^+(n)$ with linearly independent sign functions $\{F_i\}_{i=1}^{2^{n-1}} \subset \mathrm{Ker}\,(\mathcal{H}_n - I_{2^n})$ and a subset $\{g_i\}_{i=1}^{2^{n-1}} \subset \mathrm{SB}^-(n)$ with linearly independent sign functions $\{G_i\}_{i=1}^{2^{n-1}} \subset \mathrm{Ker}\,(\mathcal{H}_n + I_{2^n})$.

Prove the first statement. Let A preserves self-duality. Since the matrix A is invertible one, the vectors $\{AF_i\}_{i=1}^{2^{n-1}}$ are also linearly independent sign functions of self-dual bent functions. Then for any sign functions G of $g \in \mathrm{SB}^-(n)$ we have:

$$\langle AG, AF_i \rangle = \left\langle A^T AG, F_i \right\rangle = \langle G, F_i \rangle = 0$$

for $i = 1, 2, ..., 2^{n-1}$. That is, for every anti-self-dual bent function g its image $\varphi(g)$ is also an anti-self-dual bent function. By the same arguments one can show that the statement is true in opposite direction as well.

Now prove the second assertion. If $A\mathcal{H}_n = \mathcal{H}_n A$, then for any sign functions F of $f \in \mathrm{SB}^+(n)$ it holds:

$$\mathcal{H}_n\,(AF) = A\,(\mathcal{H}_n F) = AF,$$

hence the mapping preserves self-duality.

Denote $B = \mathcal{H}_n A - A\mathcal{H}_n$ and assume that the mapping with matrix A preserves self-duality and, as mentioned in the first assertion, anti-self-duality. In particular, for $i = 1, 2, ..., 2^{n-1}$ it holds

$$\mathcal{H}_n\,(AF_i) = AF_i$$

and

$$\mathcal{H}_n\,(AG_i) = -AG_i.$$

For $i = 1, 2, ..., 2^{n-1}$ we have:

$$(\mathcal{H}_n A - A\mathcal{H}_n)\,F_i \;=\; \mathcal{H}_n\,(AF_i) \,-\, A\,(\mathcal{H}_n F_i) \;=\; \mathcal{H}_n\,(AF_i) \,-\, AF_i \;=\; BF_i.$$

Then $BF_i = \mathbf{0} \in \mathbb{R}^{2^n}$ for every $i = 1, 2, ..., 2^{n-1}$. Since the set $\{F_i\}_{i=1}^{2^{n-1}}$ forms a basis of the subspace $\mathrm{Ker}\,(\mathcal{H}_n - I_{2^n})$ it can be deduced that all rows of the matrix B are vectors from the subspace $(\mathrm{Ker}\,(\mathcal{H}_n - I_{2^n}))^{\perp} = \mathrm{Ker}\,(\mathcal{H}_n + I_{2^n})$.

For $i = 1, 2, ..., 2^{n-1}$ we also have:

$$(\mathcal{H}_n A - A\mathcal{H}_n)\,G_i \;=\; \mathcal{H}_n\,(AG_i) \,-\, A\,(\mathcal{H}_n G_i) \;=\; \mathcal{H}_n\,(AG_i) \,+\, AG_i \;=\; BG_i.$$

In this case $BG_i = \mathbf{0} \in \mathbb{R}^{2^n}$ for every $i = 1, 2, ..., 2^{n-1}$. Since the set $\{G_i\}_{i=1}^{2^{n-1}}$ forms a basis of the subspace $\mathrm{Ker}\,(\mathcal{H}_n + I_{2^n})$ we can conclude that all rows of the matrix B are vectors from the subspace $(\mathrm{Ker}\,(\mathcal{H}_n + I_{2^n}))^{\perp} = \mathrm{Ker}\,(\mathcal{H}_n - I_{2^n})$.

Thus we have proved that all rows of the matrix B lie in $\mathrm{Ker}\,(\mathcal{H}_n + I_{2^n}) \cap \mathrm{Ker}\,(\mathcal{H}_n - I_{2^n})$ but the intersection of orthogonal subspaces consists only of the zero element of the space $\mathbb{R}^n$. Therefore the matrix B is zero matrix. □

Theorem 1. *An isometric mapping $f(x) \longrightarrow f(\pi(x)) \oplus g(x)$ of the set of all Boolean functions in $n \geqslant 4$ variables into itself preserves (anti-)self-duality if and only if*

$$\pi(x) = L(x \oplus c)$$

and

$$g(x) = \langle c, x \rangle \oplus d,$$

where $L \in \mathcal{O}_n$, $c \in \mathbb{F}_2^n$, $\mathrm{wt}(c)$ is even, $d \in \mathbb{F}_2$.

Proof. The opposite direction immediately comes from [4] (Theorem 1).

Assume that A is a matrix of the mapping $f(x) \longrightarrow f(\pi(x)) \oplus g(x)$ of the set of all Boolean functions in n variables into itself and this mapping preserves (anti-)self-duality. Let $T_{a,r}$ be a sign function of an affine function $\langle a, x \rangle \oplus r$, where $a \in \mathbb{F}_2^n, r \in \mathbb{F}_2$. In other words $T_{a,r}$ is equal to some row (column) of the matrix H_n or $-H_n$. From Proposition 1 it follows that $A\mathcal{H}_n = \mathcal{H}_n A$ hence

$$\mathcal{H}_n(AT_{a,r}) = A(\mathcal{H}_n T_{a,r}) = 2^{n/2}\sigma \cdot Ae_k = 2^{n/2}\sigma' \cdot e_{k'},$$

where $k, k' \in \{1, 2, ..., 2^n\}, \sigma, \sigma' \in \{\pm 1\}$. Then

$$AT_{a,r} = 2^{n/2}\sigma' \cdot \mathcal{H}_n e_{k'} = T_{a',r'}$$

for some $a' \in \mathbb{F}_2^n, r' \in \mathbb{F}_2$.

Thus the considered mapping transforms the set of all affine functions in n variables into itself hence it has form

$$f(x) \longrightarrow f(Lx \oplus b) \oplus \langle c, x \rangle \oplus d,$$

where L is a $n \times n$ invertible binary matrix, $b, c \in \mathbb{F}_2^n, d \in \mathbb{F}_2$, see [10], for example.

Now consider the relation $AH_n = H_n A$ in details. Denote, $N = 2^n$ and let, as before, $v_k \in \mathbb{F}_2^n$ be a binary representation of the number $k \in \{0, 1, ..., 2^n - 1\}$.

Then

$$H_n = \begin{pmatrix} (-1)^{\langle v_0,v_0\rangle} & (-1)^{\langle v_0,v_1\rangle} & \dots & (-1)^{\langle v_0,v_{N-1}\rangle} \\ (-1)^{\langle v_1,v_0\rangle} & (-1)^{\langle v_1,v_1\rangle} & \dots & (-1)^{\langle v_1,v_{N-1}\rangle} \\ \vdots & \vdots & \ddots & \vdots \\ (-1)^{\langle v_{N-1},v_0\rangle} & (-1)^{\langle v_{N-1},v_1\rangle} & \dots & (-1)^{\langle v_{N-1},v_{N-1}\rangle} \end{pmatrix}$$

and A is the matrix

$$\begin{array}{c} Lv_i \oplus b \\ v_i \end{array} \begin{pmatrix} & & 0 & & \\ & & \vdots & & \\ & & 0 & & \\ 0 & \dots & (-1)^{\langle c,v_i\rangle \oplus d} & \dots & 0 \\ & & 0 & & \\ & & \vdots & & \\ & & 0 & & \end{pmatrix},$$

in which in the row with number $(i+1) \in \{1, 2, ..., N\}$ a nonzero element is in the $(j+1)$-th column, where j is a number with binary representation $Lv_i \oplus b$.

Fix arbitrary $i, j \in \{0, 1, ..., N-1\}$. Write explicitly

$$(AH_n)_{i+1,j+1} = (-1)^{\langle c,v_i\rangle \oplus \langle Lv_i \oplus b,v_j\rangle \oplus d}.$$

In order to obtain $(H_nA)_{i+1,j+1}$ rewrite matrix A in the following form

$$\begin{array}{c} L^{-1}(v_j \oplus b) \\ \end{array} \begin{array}{c} v_j \\ \end{array} \begin{pmatrix} & & 0 & & \\ & & \vdots & & \\ & & 0 & & \\ 0 & \dots & (-1)^{\left\langle c,L^{-1}(v_j\oplus b)\right\rangle \oplus d} & \dots & 0 \\ & & 0 & & \\ & & \vdots & & \\ & & 0 & & \end{pmatrix}.$$

Then it clear that

$$(H_nA)_{i+1,j+1} = (-1)^{\left\langle v_i,L^{-1}(v_j\oplus b)\right\rangle \oplus \left\langle c,L^{-1}(v_j\oplus b)\right\rangle \oplus d}.$$

Since $AH_n = H_nA$ implies $(AH_n)_{i+1,j+1} = (H_nA)_{i+1,j+1}$ for any $i, j \in$

$\{0, 1, ..., N-1\}$, the following relation must hold

$$(-1)^{\langle c, v_i\rangle \oplus \langle Lv_i \oplus b, v_j\rangle \oplus d} = (-1)^{\left\langle v_i, L^{-1}(v_j \oplus b)\right\rangle \oplus \left\langle c, L^{-1}(v_j \oplus b)\right\rangle \oplus d},$$

or equivalently

$$\langle c, x\rangle \oplus \langle Lx \oplus b, y\rangle \oplus d = \left\langle x, L^{-1}(y \oplus b)\right\rangle \oplus \\ \oplus \left\langle c, L^{-1}(y \oplus b)\right\rangle \oplus d. \quad (1)$$

for any $x, y \in \mathbb{F}_2^n$.

Put $y \in \mathbb{F}_2^n$ with $\mathrm{wt}(y) = 0$ in (1). Then

$$\langle c, x\rangle = \left\langle x, L^{-1}b\right\rangle \oplus \left\langle c, L^{-1}b\right\rangle,$$

$$\left\langle x, L^{-1}b \oplus c\right\rangle = \left\langle c, L^{-1}b\right\rangle$$

for any $x \in \mathbb{F}_2^n$. Then

$$\begin{cases} L^{-1}b \oplus c = 0, \\ \left\langle c, L^{-1}b\right\rangle = 0, \end{cases}$$

$$\begin{cases} b = Lc, \\ \mathrm{wt}\,(c) \text{ is even.} \end{cases} \quad (2)$$

Return to (1) and take (2) into account:

$$\langle c, x\rangle \oplus \langle Lx \oplus Lc, y\rangle = \left\langle x, L^{-1}(y \oplus Lc)\right\rangle \oplus \left\langle c, L^{-1}(y \oplus Lc)\right\rangle,$$

$$\langle c, x\rangle \oplus \langle Lx, y\rangle \oplus \langle Lc, y\rangle = \left\langle x, L^{-1}y\right\rangle \oplus \langle x, c\rangle \oplus \left\langle c, L^{-1}y\right\rangle \oplus \langle c, c\rangle,$$

$$\langle Lx, y\rangle \oplus \langle Lc, y\rangle = \left\langle x, L^{-1}y\right\rangle \oplus \left\langle c, L^{-1}y\right\rangle,$$

$$\langle L(x \oplus c), y\rangle = \left\langle \left(L^{-1}\right)^T (x \oplus c), y\right\rangle.$$

for any $x, y \in \mathbb{F}_2^n$. In this case

$$L(x \oplus c) = \left(L^{-1}\right)^T (x \oplus c)$$

for any $x \in \mathbb{F}_2^n$ that is

$$L(z) = \left(L^{-1}\right)^T (z)$$

for any $z \in \mathbb{F}_2^n$. It holds if and only if

$$L = \left(L^{-1}\right)^T. \tag{3}$$

Thus, combining (2) and (3) we obtain

$$\begin{cases} L^{-1} = L^T, \\ b = Lc, \\ \mathrm{wt}\,(c) \text{ is even.} \end{cases}$$

□

4 Isometric bijections between self-dual and anti-self-dual bent functions

It is known [2] (Theorems 5.1, 5.3) that there exists a bijection between $\mathrm{SB}^+(n)$ and $\mathrm{SB}^-(n)$, based on the decomposition of sign functions of (anti-)self-dual bent functions. Namely, let $(Y, Z) \in \{\pm 1\}^{2^n}$, where $Y, Z \in \{\pm 1\}^{2^{n-1}}$, be a sign function for some $f \in \mathrm{SB}^+(n)$. Then a vector $(Z, -Y) \in \{\pm 1\}^{2^n}$ is a sign function for some function from $\mathrm{SB}^-(n)$. In terms of isometric mappings the mentioned transform can be represented as

$$f(x) \longrightarrow f\left(x \oplus c\right) \oplus \langle c, x \rangle,$$

where $c = (1, 0, 0, ..., 0) \in \mathbb{F}_2^n$.

In paper [6] it was mentioned that the more general form of this mapping

$$f(x) \longrightarrow f\left(x \oplus c\right) \oplus \langle c, x \rangle,$$

where $c \in \mathbb{F}_2^n$, $\mathrm{wt}(c)$ is odd, is a bijection between $\mathrm{SB}^+(n)$ and $\mathrm{SB}^-(n)$. It is obvious that this mapping is an element from $\mathcal{I}_n$.

In this section we generalize these results within isometric mappings.

Proposition 2. *Let $n \geqslant 4$. Isometric mapping $\varphi \in \mathcal{I}_n$ with matrix A is a bijection between $\mathrm{SB}^+(n)$ and $\mathrm{SB}^-(n)$ if and only if $A\mathcal{H}_n = -\mathcal{H}_n A$.*

Proof. If $\mathcal{H}_n A = -A\mathcal{H}_n$, then for any sign functions F, G of $f \in \mathrm{SB}^+(n)$ and $g \in \mathrm{SB}^-(n)$ respectively it holds:

$$\mathcal{H}_n\left(AF\right) = -A\left(\mathcal{H}_n F\right) = -AF,$$

$$\mathcal{H}_n\left(AG\right) = -A\left(\mathcal{H}_n G\right) = AG,$$

hence the mapping is a bijection between $\mathrm{SB}^+(n)$ and $\mathrm{SB}^-(n)$.

Take $\{f_i\}_{i=1}^{2^{n-1}} \subset \mathrm{SB}^+(n)$ with linearly independent sign functions $\{F_i\}_{i=1}^{2^{n-1}} \subset \mathrm{Ker}\left(\mathcal{H}_n - I_{2^n}\right)$ and $\{g_i\}_{i=1}^{2^{n-1}} \subset \mathrm{SB}^-(n)$ with linearly independent sign functions $\{G_i\}_{i=1}^{2^{n-1}} \subset \mathrm{Ker}\left(H_n + I_{2^n}\right)$ from the proof of the Proposition 1. Denote $B = \mathcal{H}_n A + A\mathcal{H}_n$ and assume that the mapping with matrix A is a bijection between $\mathrm{SB}^+(n)$ and $\mathrm{SB}^-(n)$. In particular, for $i = 1, 2, ..., 2^{n-1}$ it holds

$$\mathcal{H}_n\left(AF_i\right) = -AF_i$$

and

$$\mathcal{H}_n\left(AG_i\right) = AG_i.$$

For $i = 1, 2, ..., 2^{n-1}$ we have:

$$\left(\mathcal{H}_n A + A\mathcal{H}_n\right) F_i \;=\; \mathcal{H}_n\left(AF_i\right) + A\left(\mathcal{H}_n F_i\right) \;=\; \mathcal{H}_n\left(AF_i\right) + AF_i \;=\; BF_i.$$

Then $BF_i = \mathbf{0} \in \mathbb{R}^{2^n}$ for every $i = 1, 2, ..., 2^{n-1}$. Since the set $\{F_i\}_{i=1}^{2^{n-1}}$ forms a basis of the subspace $\mathrm{Ker}\left(\mathcal{H}_n - I_{2^n}\right)$ it can be deduced that all rows of the matrix B are vectors from the subspace $\left(\mathrm{Ker}\left(\mathcal{H}_n - I_{2^n}\right)\right)^{\perp} = \mathrm{Ker}\left(\mathcal{H}_n + I_{2^n}\right)$.

For $i = 1, 2, ..., 2^{n-1}$ we also have:

$$\left(\mathcal{H}_n A + A\mathcal{H}_n\right) G_i \;=\; \mathcal{H}_n\left(AF_i\right) + A\left(\mathcal{H}_n G_i\right) \;=\; \mathcal{H}_n\left(AG_i\right) - AG_i \;=\; BG_i.$$

In this case $BG_i = \mathbf{0} \in \mathbb{R}^{2^n}$ for every $i = 1, 2, ..., 2^{n-1}$. Since the set $\{G_i\}_{i=1}^{2^{n-1}}$ forms a basis of the subspace $\mathrm{Ker}\left(\mathcal{H}_n + I_{2^n}\right)$ we can conclude that all rows of the matrix B are vectors from the subspace $\left(\mathrm{Ker}\left(\mathcal{H}_n + I_{2^n}\right)\right)^{\perp} = \mathrm{Ker}\left(\mathcal{H}_n - I_{2^n}\right)$.

Thus we have proved that all rows of the matrix B lie in $\mathrm{Ker}\left(\mathcal{H}_n + I_{2^n}\right) \cap \mathrm{Ker}\left(\mathcal{H}_n - I_{2^n}\right)$ but the intersection of orthogonal subspaces consists only of the zero element of the space $\mathbb{R}^n$. Therefore the matrix B is zero matrix. □

Theorem 2. *An isometric mapping $f(x) \longrightarrow f\left(\pi(x)\right) \oplus g(x)$ of the set of all Boolean functions in $n \geqslant 4$ variables into itself is a bijection between $\mathrm{SB}^+(n)$ and $\mathrm{SB}^-(n)$ if and only if*

$$\pi(x) = L\left(x \oplus c\right)$$

and

$$g(x) = \langle c, x \rangle \oplus d,$$

where $L \in \mathcal{O}_n$, $c \in \mathbb{F}_2^n$, $\mathrm{wt}(c)$ *is odd,* $d \in \mathbb{F}_2$.

Proof. Let $f \in \mathcal{B}_n$ and $\widetilde{f} = f \oplus \varepsilon$ for some $\varepsilon \in \mathbb{F}_2$. Consider a function $g(x) = f\left(L\left(x \oplus c\right)\right) \oplus \langle c, x\rangle \oplus d$, where $L \in \mathcal{O}_n$, $c \in \mathbb{F}_2^n$, $\mathrm{wt}(c)$ is odd, $d \in \mathbb{F}_2$:

$$W_g(y) = \sum_{x \in \mathbb{F}_2^n} (-1)^{\langle x,y\rangle \oplus g(x)} = \sum_{x \in \mathbb{F}_2^n} (-1)^{\langle x,y\rangle \oplus f(L(x \oplus c)) \oplus \langle c,x\rangle \oplus d} =$$

$$= (-1)^d \sum_{x \in \mathbb{F}_2^n} (-1)^{\langle x,y \oplus c\rangle \oplus f(L(x \oplus c))} =$$

$$= (-1)^d \sum_{z \in \mathbb{F}_2^n} (-1)^{\left\langle L^{-1}z \oplus c, y \oplus c\right\rangle \oplus f(z)} =$$

$$= (-1)^{d \oplus \langle c,y\rangle \oplus \langle c,c\rangle} \sum_{z \in \mathbb{F}_2^n} (-1)^{\langle z, L(y \oplus c)\rangle \oplus f(z)} =$$

$$= (-1)^{d \oplus \langle c,y\rangle \oplus 1} 2^{n/2} (-1)^{\widetilde{f}(L(y \oplus c))} = 2^{n/2} (-1)^{f(L(y \oplus c)) \oplus \langle c,y\rangle \oplus d \oplus \varepsilon \oplus 1} =$$

$$= 2^{n/2} (-1)^{g(y) \oplus \varepsilon \oplus 1} = 2^{n/2} (-1)^{\tilde{g}(y)}.$$

The opposite direction has been proved.

By using the same considerations as in the proof of the Theorem 1 it has form

$$f(x) \longrightarrow f\left(Lx \oplus b\right) \oplus \langle c, x\rangle \oplus d,$$

where L is a $n \times n$ invertible binary matrix, $, b, c \in \mathbb{F}_2^n, d \in \mathbb{F}_2$.

From Proposition 2 it follows that $AH_n = -H_nA$. Let, as before, $v_k \in \mathbb{F}_2^n$ be a binary representation of the number $k \in \{0, 1, ..., 2^n - 1\}$.

Recall from the proof of the Theorem 1 that

$$\left(AH_n\right)_{i+1,j+1} = (-1)^{\langle c,v_i\rangle \oplus \langle Lv_i \oplus b, v_j\rangle \oplus d},$$

$$\left(H_nA\right)_{i+1,j+1} = (-1)^{\left\langle v_i, L^{-1}(v_j \oplus b)\right\rangle \oplus \left\langle c, L^{-1}(v_j \oplus b)\right\rangle \oplus d}$$

for any $i, j \in \{0, 1, ..., 2^n - 1\}$.

Since $AH_n = -H_nA$ implies $\left(AH_n\right)_{i+1,j+1} = -\left(H_nA\right)_{i+1,j+1}$ for any $i, j \in \{0, 1, ..., 2^n - 1\}$, the following relation must hold

$$(-1)^{\langle c,v_i\rangle \oplus \langle Lv_i \oplus b, v_j\rangle \oplus d} = (-1)^{\left\langle v_i, L^{-1}(v_j \oplus b)\right\rangle \oplus \left\langle c, L^{-1}(v_j \oplus b)\right\rangle \oplus d \oplus 1},$$

or equivalently

$$\langle c, x\rangle \oplus \langle Lx \oplus b, y\rangle \oplus d = \left\langle x, L^{-1}(y \oplus b)\right\rangle \oplus \\ \oplus \left\langle c, L^{-1}(y \oplus b)\right\rangle \oplus d \oplus 1 \quad (4)$$

for any $x, y \in \mathbb{F}_2^n$.

Put $y \in \mathbb{F}_2^n$ with $\mathrm{wt}(y) = 0$ in (4). Then

$$\langle c, x\rangle = \left\langle x, L^{-1}b\right\rangle \oplus \left\langle c, L^{-1}b\right\rangle \oplus 1,$$

$$\left\langle x, L^{-1}b \oplus c\right\rangle = \left\langle c, L^{-1}b\right\rangle \oplus 1$$

for any $x \in \mathbb{F}_2^n$. Then

$$\begin{cases} L^{-1}b \oplus c = 0, \\ \left\langle c, L^{-1}b\right\rangle = 1, \end{cases}$$

$$\begin{cases} b = Lc, \\ \mathrm{wt}\,(c) \text{ is odd.} \end{cases} \quad (5)$$

Return to (4) and take (5) into account:

$$\langle c, x\rangle \oplus \langle Lx \oplus Lc, y\rangle = \left\langle x, L^{-1}(y \oplus Lc)\right\rangle \oplus \left\langle c, L^{-1}(y \oplus Lc)\right\rangle \oplus 1,$$

$$\langle c, x\rangle \oplus \langle Lx, y\rangle \oplus \langle Lc, y\rangle = \left\langle x, L^{-1}y\right\rangle \oplus \langle x, c\rangle \oplus \left\langle c, L^{-1}y\right\rangle \oplus \langle c, c\rangle \oplus 1,$$

$$\langle Lx, y\rangle \oplus \langle Lc, y\rangle = \left\langle x, L^{-1}y\right\rangle \oplus \left\langle c, L^{-1}y\right\rangle,$$

$$\langle L(x \oplus c), y\rangle = \left\langle \left(L^{-1}\right)^T (x \oplus c), y\right\rangle.$$

for any $x, y \in \mathbb{F}_2^n$. It holds if and only if

$$L = \left(L^{-1}\right)^T. \quad (6)$$

Thus, combining (5) and (6) we obtain

$$\begin{cases} L^{-1} = L^T, \\ b = Lc, \\ \mathrm{wt}\,(c) \text{ is odd.} \end{cases}$$

□

5 Isometric mappings preserving the Rayleigh quotient

In [2] the *Rayleigh quotient* S_f of a Boolean function $f \in \mathcal{F}_n$ was defined as

$$S_f = \sum_{x,y \in \mathbb{F}_2^n} (-1)^{f(x) \oplus f(y) \oplus \langle x,y \rangle} = \sum_{y \in \mathbb{F}_2^n} (-1)^{f(y)} W_f(y).$$

For any $f \in \mathcal{B}_n$ the *normalized Rayleigh quotient* N_f is a number

$$N_f = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x) \oplus \widetilde{f}(x)} = 2^{-n/2} S_f.$$

In [2] (Theorem 3.1) it was proved that for any $f \in \mathcal{F}_n$ the absolute value of S_f is at most $2^{3n/2}$ with equality if and only if f is self-dual $\left(+2^{3n/2}\right)$ and anti-self-dual $\left(-2^{3n/2}\right)$ bent function. In the article [3] the operations on Boolean functions that preserve bentness and the Rayleigh quotient were given. Namely, it was proven that for any $f \in \mathcal{B}_n, L \in \mathcal{O}_n, c \in \mathbb{F}_2^n, d \in \mathbb{F}_2$ the functions $g, h \in \mathcal{B}_n$ defined as $g(x) = f(Lx) \oplus d$ and $h(x) = f(x \oplus c) \oplus \langle c, x \rangle$ provide $N_g = N_f$ and $N_h = (-1)^{\langle c,c \rangle} N_f$.

One can notice that the mentioned operations are isometric mappings from $\mathcal{I}_n$. In this section we generalize these results within isometric mappings.

Theorem 3. *An isometric mapping $\varphi \in \mathcal{I}_n$ of the set of all Boolean functions in $n \geqslant 4$ variables into itself preserves the Rayleigh quotient if and only if it preserves self-duality.*

Proof. For straight direction it is enough to mention that $S_f = +2^{3n/2}$ if and only if $f \in \mathrm{SB}^+(n)$ ([2], Theorem 3.1).

Assume that the mapping φ preserves self-duality. Let A be its matrix. Then by Proposition 1 we have $AH_n = H_n A$. Rewrite the Rayleigh quotient in the following form:

$$S_f = \sum_{x,y \in \mathbb{F}_2^n} (-1)^{f(x) \oplus f(y) \oplus \langle x,y \rangle} = \langle F, H_n F \rangle,$$

where F is a sign function. The mapping preserves the Rayleigh quotient if

$$S_{\varphi(f)} = \langle AF, H_n (AF) \rangle = \langle F, H_n F \rangle = S_f.$$

for any sign function F. Consider

$$\langle AF, H_n(AF)\rangle = \langle AF, A(H_nF)\rangle = \left\langle A^TAF, H_nF\right\rangle = \langle F, H_nF\rangle,$$

therefore it preserves the Rayleigh quotient. □

Corollary 1. *An isometric mapping $f(x) \longrightarrow f(\pi(x)) \oplus g(x)$ of the set of all Boolean functions in $n \geqslant 4$ variables into itself preserves the Rayleigh quotient if and only if*

$$\pi(x) = L(x \oplus c),$$

and

$$g(x) = \langle c, x\rangle \oplus d,$$

where $L \in \mathcal{O}_n$, $c \in \mathbb{F}_2^n$, $\mathrm{wt}(c)$ is even, $d \in \mathbb{F}_2$.

Theorem 4. *An isometric mapping $\varphi \in \mathcal{I}_n$ of the set of all Boolean functions in $n \geqslant 4$ variables into itself changes the sign of the Rayleigh quotient if and only if it is a bijection between $\mathrm{SB}^+(n)$ and $\mathrm{SB}^-(n)$.*

Proof. For straight direction it is enough to mention that $S_f = +2^{3n/2}$ if and only if $f \in \mathrm{SB}^+(n)$ and $S_f = -2^{3n/2}$ if and only if $f \in \mathrm{SB}^-(n)$ ([2], Theorem 3.1).

Assume that the mapping φ is a bijection between $\mathrm{SB}^+(n)$ and $\mathrm{SB}^-(n)$. Let A be its matrix. Then by Proposition 2 we have $AH_n + H_nA = 0$. Rewrite the Rayleigh quotient in the following form:

$$S_f = \sum_{x,y \in \mathbb{F}_2^n} (-1)^{f(x) \oplus f(y) \oplus \langle x, y\rangle} = \langle F, H_nF\rangle,$$

where F is a sign function. The mapping changes the sign of the Rayleight quotient if

$$S_{\varphi(f)} = \langle AF, H_n(AF)\rangle = -\langle F, H_nF\rangle = -S_f.$$

for any sign function F. Consider

$$\langle AF, H_n(AF)\rangle = \langle AF, -A(H_nF)\rangle = \\ = -\left\langle A^TAF, H_nF\right\rangle = -\langle F, H_nF\rangle,$$

therefore it changes the sign of the Rayleigh quotient. □

Corollary 2. *An isometric mapping $f(x) \longrightarrow f(\pi(x)) \oplus g(x)$ of the set of all Boolean functions in $n \geqslant 4$ variables into itself changes the sign of the Rayleigh*

quotient if and only if

$$\pi(x) = L\left(x \oplus c\right),$$

and

$$g(x) = \langle c, x \rangle \oplus d,$$

where $L \in \mathcal{O}_n$, $c \in \mathbb{F}_2^n$, $\mathrm{wt}(c)$ *is odd,* $d \in \mathbb{F}_2$.

The following corollary can be deduced:

Corollary 3. *Any isometric mapping of the set of all Boolean functions in* $n \geqslant 4$ *variables into itself which preserves the Rayleigh quotient or changes the sign of the Rayleigh quotient also preserves bentness.*

The Rayleigh quotient characterizes the Hamming distance between a bent-function and its dual. Indeed, let $f \in \mathcal{B}_n$, then

$$\mathrm{dist}\left(f, \widetilde{f}\right) = 2^{n-1} - \frac{1}{2^{n/2+1}} S_f = 2^{n-1} - \frac{1}{2} N_f.$$

Thus from Proposition 1, Theorem 1 it follows that the isometric mapping preserves bentness and the Hamming distance between any bent function in $n \geqslant 4$ variables and its dual if and only if it preserves self-duality and its form is described by the Theorem 1.

Let us summarize the main results from this paper. Let φ be an isometric mapping of the set of all Boolean functions in $n \geqslant 4$ variables into itself with matrix A, namely

$$\varphi : f(x) \longrightarrow f\left(\pi(x)\right) \oplus g(x),$$

where π is a permutation in $\mathbb{F}_2^n$ and $g \in \mathcal{F}_n$.

Theorem 5. *The following conditions are equivalent:*

- φ *preserves self-duality;*
- φ *preserves anti-self-duality;*
- φ *preserves the Rayleigh quotient;*
- φ *preserves bentness and the Hamming distance between any bent function and its dual;*
- $\pi(x) = L\left(x \oplus c\right)$ *and* $g(x) = \langle c, x \rangle \oplus d$, *where* $L \in \mathcal{O}_n$, $c \in \mathbb{F}_2^n$, $\mathrm{wt}(c)$ *is even,* $d \in \mathbb{F}_2$;

– $A\mathcal{H}_n = \mathcal{H}_n A$.

Theorem 6. *The following conditions are equivalent:*

– φ *is a bijection between* $\mathrm{SB}^+(n)$ *and* $\mathrm{SB}^-(n)$*;*

– φ *changes sign of the Rayleigh quotient;*

– $\pi(x) = L\,(x \oplus c)$ *and* $g(x) = \langle c, x\rangle \oplus d$*, where* $L \in \mathcal{O}_n$*,* $c \in \mathbb{F}_2^n$*,* $\mathrm{wt}(c)$ *is odd,* $d \in \mathbb{F}_2$*;*

– $A\mathcal{H}_n = -\mathcal{H}_n A$.

It follows that the way of classifying self-dual bent functions given in [2, 4] is the most general within isometric mappings.

References

[1] Carlet C., "Boolean functions for cryptography and error correcting code", *In: Crama Y., Hammer P.L. (eds.) Boolean Models and Methods in Mathematics, Computer Science, and Engineering. Cambridge University Press, Cambridge*, 2010, 257 – 397.

[2] Carlet C., Danielson L.E., Parker M.G., Solé. P., "Self-dual bent functions", *Int. J. Inform. Coding Theory*, **1** (2010), 384 – 399.

[3] Danielsen L.E., Parker M.G., Solé P., "The Rayleigh quotient of bent functions", *Springer Lect. Notes in Comp. Sci.*, **5921** (2009), 418 – 432.

[4] Feulner T., Sok L., Solé P., "Towards the Classification of Self-Dual Bent Functions in Eight Variables", *Des. Codes Cryptogr.*, **68**:1 (2013), 395 – 406.

[5] Janusz G.J., "Parametrization of self-dual codes by orthogonal matrices", *Finite Fields Appl.*, **13**:3 (2007), 450 – 491.

[6] Hou X.-D., "Classification of self dual quadratic bent functions", *Des. Codes Cryptogr.*, **63**:2 (2012), 183 – 198.

[7] 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.

[8] Kutsenko A.V., "The Hamming Distance Spectrum Between Self-Dual Maiorana-McFarland Bent Functions", *Journal of Applied and Industrial Mathematics*, **12**:1 (2018), 112 – 125.

[9] Kutsenko A.V., "On metrical properties of self-dual bent functions", 2019, submitted.

[10] MacWilliams F.J., Sloane N.J.A., "The Theory of Error-Correcting Codes", *Amsterdam, New York, Oxford: North-Holland*, 1983, 782.

[11] 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.

[12] Mesnager S., "Several New Infinite Families of Bent Functions and Their Duals", *IEEE Trans. Inf. Theory*, **60**:7 (2014), 4397 – 4407.

[13] Rothaus O.S., "On bent functions", *J. Combin. Theory. Ser. A*, **20**:3 (1976), 300 – 305.

[14] Tokareva N., "Bent Functions, Results and Applications to Cryptography", *Acad. Press. Elsevier*, 2015, 230.

[15] Tokareva N.N., "The group of automorphisms of the set of bent functions", *Discrete Mathematics and Applications*, **20**:5 (2010), 655 – 664.

On the Properties of Boolean Functions Related to Planar Approximation of the Filter Generator

Evgeny Alekseev and Lyudmila Kushchinskaya

Crypto-Pro LLC, Russia
{alekseev, ess}@cryptopro.ru

Abstract

This study examines the properties of Boolean functions related to ensuring the resistance of the filter generator to the key recovery method, based on the so-called planar approximations. The problem of the existence of an «ideal» function, balanced on all possible planes, is considered. The weight distribution of Boolean functions on planes of different dimensions is studied: the number of planes on which the function is not balanced and the weight of the function on planes of a given dimension are estimated. In particular, the study presents all possible sets of values of the specified parameters for functions of 5 variables.

Keywords: filter generator, Boolean function, stream cipher.

1 Introduction

The filter generator is one of the elements often used in the construction of stream ciphers ([8, 14, 15]). The internal state of such a generator, which is a binary string of a fixed length, initially filled with the key bits, is transformed from cycle to cycle using a certain fixed linear transformation. The output of each clock cycle is the value of the fixed Boolean function (which is usually called the filter function) of the current state of the generator. Several key recovery methods are proposed for this scheme (see, for example, [10, 9, 11]). Such methods are effective if the elements of the generator satisfy some negative properties (for example, the filter function is close to the function of a small number of variables or the shift register that specifies the linear transformation has a small number of feedbacks). Accordingly, to ensure resistance to these methods, generator elements must satisfy certain properties. Examples of such properties for the filter function are balancedness, nonlinearity, correlation, and algebraic immunity. Typically, these properties are called cryptographic, and a deeper

understanding of their nature and interrelationships is of particular interest (a review of the results on this subject is contained in the monograph [6]).

In [1], a key recovery method for a filter generator is proposed. It uses a planar approximation, a concept introduced in the same reference. This is the name of a set of sequences of planes (cosets in some linear subspaces). In this case, the planes included in the same sequence should be the images of a certain single plane with respect to different degrees of the linear transformation used in the generator. The key can be recovered the more effectively, the closer the filter function to the constant on those planes that are included in the approximation. The problem of constructing such approximations is generally nontrivial (some special cases are considered in [1] and [5]).

As far as the authors know, the properties that generator elements must possess to achieve the generator resistance to the method described in [1] have not been previously studied. In this study, such properties are examined exclusively in the context of the filter function. That is, the study examines whether it is possible to ensure generator resistance only by choosing such a function. This problem statement may seem unnatural. Indeed, the filter generator will be resistant to the specified method if it is unable to build a sufficiently accurate planar approximation. The possibility of its construction in the general case essentially depends on the properties of not only the filter function, but also the linear transformation. In this case, resistance can still be ensured regardless of the linear transformation, for example, with the help of a hypothetical «ideal function», which is balanced on all possible planes of all dimensions. However, this study shows (see Section 4) that there is not only such a function, but also a function balanced on all planes of at least one arbitrary dimension.

Because of the lack of a specified «ideal», this study examines how Boolean functions can be close to it. The number of planes of various dimensions on which the Boolean function is not balanced is estimated (see Section 5.2). We also obtained weight estimates of a Boolean function on planes of a fixed dimension for fixed values of its nonlinearity and weight (see Section 5.3). Therefore, an inequality relating the degree of algebraic degeneracy of a Boolean function and its nonlinearity was obtained. It is also proved that the number of planes on which the weight of the function differs from half the power of the plane by an arbitrary fixed value does not change when the function is affected by the elements of the group $\mathfrak{GU}(V_n)\mathfrak{H}_0$ — a certain generalization of the full affine group (inversion of the function values is additionally allowed). Regarding this

group, a classification of Boolean functions of 5 variables has been compiled, and for each class, the above values are given, which are related to the weight of the function on the planes (see Section 6). The study also presents (see Section 7) the results of estimation of these parameters for some specific Boolean functions (for example, for the filter function used in the LILI-128 cipher [8] and bent functions of 6 variables).

The results obtained in this work can be used to refine the estimates of the complexity of the preliminary stage of the key recovery method, during which planar approximations are constructed. In [4], the problem of constructing «good» planar approximations is studied under the conditions of fairly strong model assumptions: only the weight of the Boolean function is considered. Within the framework of this model, it is impossible to compare the resistance of generators built based on different filter functions. If the number of planes with different weights of the function on them is known for the studied Boolean function, then the estimate of the probability of adding a new plane to the trajectory can be refined (for example, using the sample model without return).

The following Section 2 outlines the basic concepts and definitions that are necessary for further discussion, and Section 3 briefly describes the method from [1] and related concepts. The following sections describe the main results of this work and the issues that remain open.

The proofs of all theorems and propositions are presented in Appendix.

2 Basic concepts and definitions

Let $\mathbb{F}_2$ be a field of 2 elements. Let $V_n = \mathbb{F}_2^n$ be a linear space of dimension n on the field $\mathbb{F}_2$. The set $supp(x) = \{i \in \{0, \ldots, n-1\} \,|\, x_i = 1\}$ is called the carrier of vector $x = (x_0, \ldots, x_{n-1}) \in V_n$. The fact that $L \subseteq V_n$ is a subspace of space V_n shall be denoted as follows: $L < V_n$. Let us denote the linear shell of vectors $v^{(1)}, \ldots, v^{(k)}$ of V_n by $L(v^{(1)}, \ldots, v^{(k)})$ [12]. A coset in the subspace of this space shall be called the *plane* plane in the space V_n, and its dimension shall be the dimension of this subspace. Planes in the space V_n, the dimension of which is equal to $n-1$, are called hyperplanes.

The Boolean function f of n variables is the mapping $f : V_n \to \mathbb{F}_2$. The set of all Boolean functions of n variables shall be denoted by $\mathcal{F}_n$. The carrier of the function $f \in \mathcal{F}_n$ is the set $1_f = \{x \in V_n \,|\, f(x) = 1\}$. *The weight* $\mathrm{wt}\,(f)$ of a Boolean function $f \in \mathcal{F}_n$ is the power of its carrier. The function $f \in \mathcal{F}_n$

is *balanced* if $\mathrm{wt}(f) = 2^{n-1}$ [6]. For $u \in V_n$ and $a \in \mathbb{F}_2$ let $l_{u,a}$ be an affine function $l_{u,a}(x) = \langle x, u\rangle \oplus a$ of n variables, where $\langle x, u\rangle$ is the scalar product of vectors x and u. Let l_u be the linear function $l_{u,0^n}$. For $S \subset V_n$ and $f \in \mathcal{F}_n$ $f|_S$ shall denote the restriction of the function f on the set S, that is, a function $f|_S : S \to \mathbb{F}_2$, such that $f|_S(x) = f(x)$.

Let $\mathbb{N}_0$ be the set $\mathbb{N} \cup \{0\}$. A *filtering generator* shall mean a mapping from $\mathbb{N}_0 \times V_n$ to $\mathbb{F}_2$, which is determined by a non-degenerate linear mapping $A : V_n \to V_n$ and a balanced Boolean function $f \in \mathcal{F}_n$ which assigns the number i and the vector $u^* \in V_n$ to the bit $z_i = f(A^i(u^*))$. The vector u^* shall be called the *key* or the *initial content* of the filter generator, and the sequence of bits $z_0, z_1, \ldots$ — *the output sequence* of the filter generator. The result of encrypting plaintext $x \in V_N$ based on the key $u^* \in V_n$ using a stream cipher based on the filtering generator is the vector $y \in V_N$, such that $y_i = x_i \oplus z_i$ for any $i \in \{0, \ldots, N-1\}$. In other words, $y = x \oplus z$, where $z = (z_0, z_1, \ldots, z_{N-1}) \in V_N$ is the initial segment of length N of the output sequence of the filter generator.

The Walsh-Hadamard transform is often used to analyze the cryptographic properties of Boolean functions. The Walsh-Hadamard transform of the Boolean function $f \in \mathcal{F}_n$ Fn is the function $W_f : V_n \to \mathbb{Z}$, such that $W_f(u) = \sum_{x \in V_n} (-1)^{f(x) \oplus \langle u, x\rangle}$. The values $W_f(u)$ are called Walsh-Hadamard coefficients (or, in short, Walsh coefficients). The following relations are valid (see, for example, [6]):

$$W_f(u) = 2^n - 2 \cdot \mathrm{dist}(f, l_u), \tag{1}$$

$$\sum_{u \in V_n} W_f^2(u) = 2^{2n} \text{ (Parseval equality)}. \tag{2}$$

3 Key recovery method and planar approximations

In [1], a method for recovering the filter generator key was proposed, the main idea of which is as follows. The membership of a key in a certain plane is determined as a result of only one check of the equality of some bits of the output sequence by previously calculated fixed values. To improve the efficiency of the method, it is necessary to be able to perform such a check for a set of planes that almost completely cover the entire space of keys V_n. A detailed description of the method, evaluation of its characteristics and application examples are given in [1].

For the present study, the necessary condition for the efficiency of this

method is important. It consists in the existence for the filter generator of an approximation of a special type, which was called *planar* in [1]. The following are the relevant definitions from this reference.

Here and below, A is a linear mapping from V_n to V_n, and f is a function from $\mathcal{F}_n$.

Let $m \in \mathbb{N}$, $\mathbb{L} = (L_0, \ldots, L_m)$, where all L_i are planes in V_n, and $\mathbb{T} = (t_0, t_1, \ldots, t_m)$, where $t_0 = 0, t_0 \leqslant t_1 < \ldots < t_m$ are positive integers. A triple $\mathrm{Traj} = (m, \mathbb{L}, \mathbb{T})$ is called a *trajectory* for A if the relations $L_i = A^{t_i - t_{i-1}}(L_{i-1}),\ i = 1, \ldots, m$ are valid. In this case, the value m is called *trajectory length*, and L_0 is an *initial plane*.

Let $\mathrm{Traj} = (m, \mathbb{L}, \mathbb{T})$ be the *trajectory* for some linear transformation. Let also $\mathbb{B} = (b_1, \ldots, b_m)$, where all $b_i \in \mathbb{F}_2$, and $\mathbb{P} = (p_1, \ldots, p_m)$, where all $p_i \in [0; 1]$. A couple $(\mathbb{B}, \mathbb{P})$ is called a *characteristic* of the Traj trajectory with respect to function $f \in \mathcal{F}_n$, if p_i is the probability that with a random equiprobable choice of a vector v from L_i the value of $f(v)$ coincides with the constant b_i.

Each trajectory of length m corresponds to 2^m characteristics, among which there is at most one characteristic that has all $p_i > \frac{1}{2}$. A characteristic with such a property is called *positive* with respect to f. The trajectory for which there is a positive characteristic is called a *suitable trajectory*. In this case, for any trajectory there is a characteristic in which $p_i \geqslant \frac{1}{2}$ for all i.

The finite set of trajectories $\mathrm{Traj}^{(1)}, \ldots, \mathrm{Traj}^{(s)}$ for A, which are suitable with respect to the function f, with pairwise different starting planes shall be called the *planar approximation* of the function f with respect to the mapping A.

The following section deals with the existence of a function for which no suitable trajectory exists, and therefore, there is no planar approximation for any A.

4 Non-existence of an «ideal» function

The method described in the previous section is not applicable for a function balanced on all planes of all dimensions. Indeed, there is no suitable trajectory for such a function. However, the following theorem says about the non-existence of even such a function, which is balanced on all planes of at least one dimension.

Further we will say that a certain plane of the space V_n is f-balanced (f-

unbalanced)) if the function f is balanced (unbalanced) on this plane. If specifying a particular function f is not important or it is clear from the context which function f is in question, we will simply say about a balanced (unbalanced) plane.

Theorem 1. *For any function $f \in \mathcal{F}_n$ and for any k, $1 \leqslant k \leqslant n-1$, there exists a plane of space V_n of dimension k such that the weight of the function f on it is different from 2^{k-1}.*

Even though the requirement of balancedness on all planes of at least one dimension, the impossibility of fulfilling which has been proved above, guarantees the absence of suitable trajectories, it can still be weakened. This is explained by the fact that the efficiency of the method depends not on the presence of unbalanced planes, but on their number and on how close the function f on such planes is to a constant. Further, we study these two parameters of Boolean functions: number of unbalanced planes and weight of functions on planes of different dimensions.

5 The weight of a Boolean function on planes

5.1 Planes of the space V_n

For convenience of presentation and perception of further material, let us define the following directed graph. Let $G = (V, E)$, where V is the set of all planes of the space V_n of dimension $k, 1 \leqslant k \leqslant n$, and $E \subset \{(u, v) | u, v \in V\}$ is the set of ordered couples of vertices such that $\dim v = \dim u - 1$ and $v \subset u$. Next, let us identify the vertex of the graph with the plane that corresponds to it.

Note that vertex V_n has only outgoing arcs and planes of dimension 1have only incoming arcs. Let us call the set of planes of one dimension equal to k, the k-th tier, $1 \leqslant k \leqslant n$. Let us also note that if two planes corresponding to vertices v_1, v_2 of one tier do not intersect, then the subgraphs formed by the vertices to which there is a directed path from v_1 and v_2, respectively, do not have common vertices.

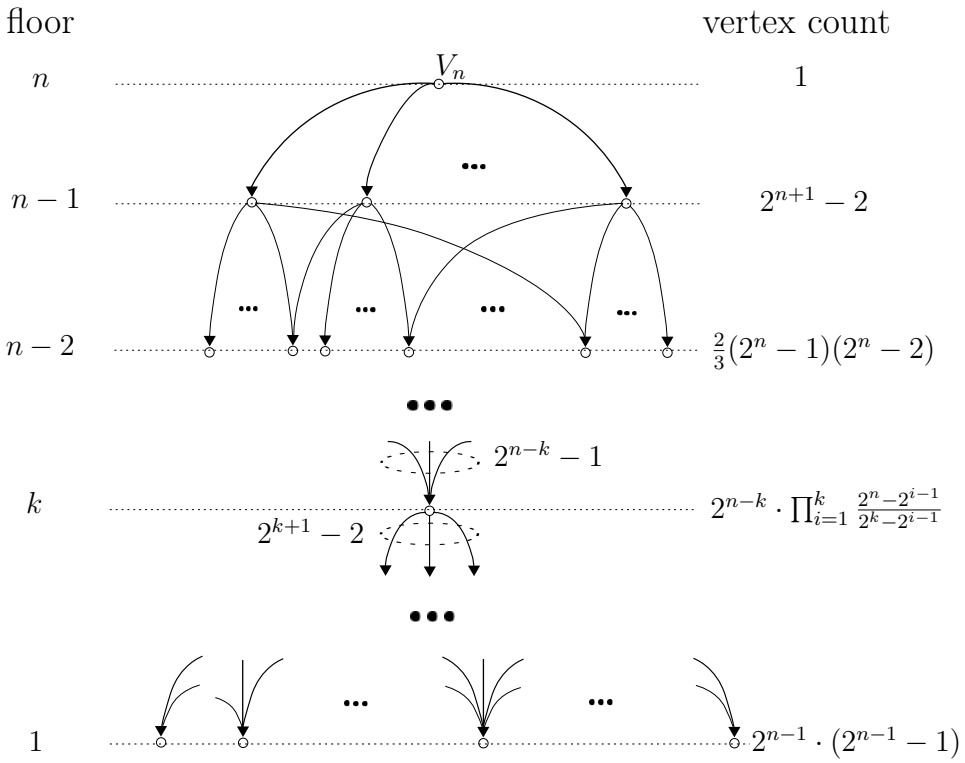


Figure 1: The properties of G.

Let us note some properties of the specified graph following from the standard linear algebra statements.

Statement 1. *The number of vertices on the k-th ($k = 1, \ldots, n$) tier of the graph is equal to*

$$2^{n-k} \cdot \prod_{i=1}^{k} \frac{2^n - 2^{i-1}}{2^k - 2^{i-1}}.$$

Statement 2. *The number of outgoing arcs for any vertex on the k-th ($k = 2, \ldots, n$) tier of the graph is $2^{k+1} - 2$.*

Statement 3. *The number of incoming arcs for any vertex on the k-th ($k = 1, \ldots, n-1$) tier of the graph is $2^{n-k} - 1$.*

Statement 4. *. For any two vertices u_1, u_2 on the k-th ($k = 2, \ldots, n-1$) tier, there is at most one vertex v on the $k-1$ tier such that there exist arcs $(u_1, v) \in E$ and $(u_2, v) \in E$.*

For convenience, these properties are summarized in Figure 1.

5.2 The number of unbalanced planes

Throughout this section, for the Boolean function $f \in \mathcal{F}_n$ let $S_f(k)$ be the number of planes of dimension $1 \leqslant k \leqslant n$ on which the weight of the function is different from 2^{k-1}, that is, on which the function is unbalanced.

Theorem 2. *For a balanced Boolean function $f \in \mathcal{F}_n$, the number of unbalanced hyperplanes is equal to twice a number of non-zero Walsh-Hadamard coefficients. In other words,*

$$S_f(n-1) = 2 \cdot |\{u \in V_n | W_f(u) \neq 0\}|.$$

Theorem 3. *For a Boolean function $f \in \mathcal{F}_n$ of the weight w, the number of unbalanced planes of dimension 1 is equal to*

$$S_f(1) = \frac{w(w-1)}{2} + \frac{(2^n - w)(2^n - w - 1)}{2}.$$

Theorem 4. *Let $f \in \mathcal{F}_n$. If for some k, $2 \leqslant k \leqslant n-1$, $S_f(k) = N > 1$, then the number of unbalanced planes of dimension $k-1$ satisfies the inequality*

$$S_f(k-1) \geqslant t \cdot (2^k - 1) - \frac{t \cdot (t+1)}{2},$$

where $t = \min(2^k - 1, N - 1)$.

Knowing the value of the number of unbalanced planes of dimension $n-1$ for a balanced function (see Theorem 2), applying Theorem 4 recursively, we can obtain non-degenerate estimates of the number of unbalanced planes on the k-th tier, $1 \leqslant k \leqslant n-2$.

Thus, according to Theorem 2, for the balanced function f = 00017FFF, the number of unbalanced planes of dimension $n-1$ is 8. Applying Theorem 4 recursively, we obtain the following estimates (values in parentheses are the real values that can be found in Appendix 1): there are at least 105 (270) unbalanced planes of dimension 3, 21 (490) unbalanced planes of dimension 2 and 3 (240) planes of dimension 1.

5.3 Some estimates of the function weight on planes

Definition 1. *The plane weight characteristic* $\mathrm{pwc}_d(f)$ *of a function f of order d, $1 \leqslant d \leqslant n$, is a vector of length $2^{d-1} + 1$, the w-th component of which is*

equal to the number of planes of dimension d on which the weight of function f is equal to either $2^{d-1} - w$ or $2^{d-1} + w$ ($0 \leqslant w \leqslant 2^{d-1}$).

For example, for functions of 5 variables identically equal to 0 and 1, planar weight characteristics of order 3 are the same and equal $(0, 0, 0, 0, 620)$. For the function $f(x_1, x_2, x_3, x_4) = x_2x_3 + x_1x_3 + x_1x_2$, the planar weight characteristic of order 2 is $(70, 64, 6)$, and of order 3 is $(22, 0, 8, 0, 0)$.

The following statement holds.

Theorem 5. *[6] Let $f \in \mathcal{F}_n$, $L \subseteq V_n$ be an arbitrary subspace and $a, b \in V_n$ be arbitrary vectors. Then,*

$$\sum_{x \in a \oplus L} (-1)^{f(x) \oplus \langle b, x \rangle} = 2^{\dim L - n} \cdot (-1)^{\langle a, b \rangle} \cdot \sum_{u \in b \oplus L^{\perp}} W_f(u)(-1)^{\langle u, a \rangle}. \qquad (3)$$

Given that $(-1)^{f(x)} = 1 - 2f(x)$, from Theorem 5 we obtain a relation for the weight of the function f on the plane $a \oplus L$ (in relation (3) we assume $b = 0^n$):

$$\mathrm{wt}\,(f|_{a \oplus L}) = 2^{\dim L - 1} - \frac{1}{2^{n - \dim L + 1}} \cdot \sum_{u \in L^{\perp}} W_f(u)(-1)^{\langle u, a \rangle}. \qquad (4)$$

Theorem 6. *Let natural n and k be such that $n \geqslant 2$ and $1 \leqslant k \leqslant n - 1$. Then for any Boolean function f of n variables, any subspace L of the space V_n of dimension k and any vector $a \in V_n$, the following inequality is valid*

$$\left| \mathrm{wt}\,(f|_{a \oplus L}) - \frac{\mathrm{wt}\,(f)}{2^{n-k}} \right| \leqslant \left(1 - \frac{1}{2^{n-k}}\right) \cdot (2^{n-1} - \mathrm{nl}\,(f)). \qquad (5)$$

Let us note that for a balanced Boolean function f, inequality (5) allows to estimate the deviation of the weight of the function f on an arbitrary plane $a \oplus L$ from half of its power $|a \oplus L|/2 = 2^{k-1}$.

A corollary of the Theorem 6 is the following statement on the relation between nonlinearity and the order of algebraic degeneracy (see [7, 3]) of the Boolean functions.

For a Boolean function $f \in \mathcal{F}_n$ the order of algebraic degeneracy $\mathrm{AD}\,(f)$ is the maximum value of k, for which there exist a subspace $L < V_n$ of dimension k, such that the function f is constant on each coset of L.

Corollary 1. *For Boolean function* $f \in \mathcal{F}_n$, *such that* $\mathrm{wt}(f) \leqslant 2^{n-1}$, *the following inequality holds:*

$$\mathrm{AD}(f) \leqslant n - \log_2 \left(\frac{\mathrm{wt}(f)}{2^{n-1} - \mathrm{nl}(f)} + 1 \right).$$

This estimation is achieved, for example, for non constant linear functions (then $\mathrm{wt}(f) = 2^{n-1}$, $\mathrm{nl}(f) = 0$ and $\mathrm{AD}(f) = n - 1$).

Now consider properties of the affine functions related to planar approximations. For natural $n \geqslant 2$ and k, $1 \leqslant k \leqslant n-1$, let $\mathcal{P}_{n,k}$ be the number of planes of dimension k of the space V_n. That is

$$\mathcal{P}_{n,k} = 2^{n-k} \cdot \prod_{i=1}^{k} \frac{2^n - 2^{i-1}}{2^k - 2^{i-1}}.$$

For affine functions, the properties of planar weight characteristics are described by the following statement.

Theorem 7. *Any non-constant affine function* $f \in \mathcal{F}_n$ *Fn on any plane is either balanced or constant. Also*

$$pwc_{n-1}(f) = (2^{n+1} - 4, 0, \ldots, 0, 2),$$

for any k, $2 \leqslant k \leqslant n - 2$, *the following ratio is true*

$$pwc_k(f) = (\mathcal{P}_{n,k} - 2 \cdot \mathcal{P}_{n-1,k}, 0, 0, \ldots, 0, 2 \cdot \mathcal{P}_{n-1,k}).$$

6 Numerical characteristics of Boolean functions of 5 variables

This section presents the numerical results of an analysis of the weight distribution of Boolean functions of 5 variables over planes of various dimensions.

This problem has been resolved exhaustively in the sense that the Appendix contains all possible values of the number of unbalanced planes for functions of 5 variables. And for balanced functions, the most interesting in terms of cryptography, all possible values of planar weight characteristics are given. It was possible to obtain and present the indicated results in a form convenient for analysis because the planar weight characteristic is invariant relative to

some generalization of the complete affine group, namely, the group $\mathfrak{GU}(V_n)\mathfrak{H}_0$. Problems related to transformation groups of a set of Boolean functions and classifications, are discussed in detail in [6].

Let $\mathfrak{GU}(V_n)\mathfrak{H}_d$ be a set of triples (A, b, h), where A is a nondegenerate $n \times n$-matrix over the field $\mathbb{F}_2$, $b \in V_n$, and h is a function from $\mathcal{F}_n$ such that $\deg h \leqslant d$. If $\alpha = (A, b, h) \in \mathfrak{GU}(V_n)\mathfrak{H}_d$, and $f \in \mathcal{F}_n$, then let f^α be a function of $\mathcal{F}_n$, such that $f^\alpha(x) = f(Ax \oplus b) \oplus h(x)$. Thus, each element of $\mathfrak{GU}(V_n)\mathfrak{H}_d$ corresponds to some transformation of the set $\mathcal{F}_n$. The set of such transformations is a group with respect to the superposition operation.

Theorem 8. *For any function $f \in \mathcal{F}_n$, any natural $d, 1 \leqslant d \leqslant n$, , and any element $\alpha \in \mathfrak{GU}(V_n)\mathfrak{H}_0$ the planar weight characteristic* $\mathrm{pwc}_d(f)$ *and* $\mathrm{pwc}_d(f^\alpha)$ *coincide.*

It is easy to see that the set $\mathcal{F}_n$ is split into non-overlapping sets $\{f^\alpha \mid \alpha \in \mathfrak{GU}(V_n)\mathfrak{H}_d\}$ called equivalence classes with respect to $\mathfrak{GU}(V_n)\mathfrak{H}_d$ and denoted by $\{f\}_{\mathfrak{GU}(V_n)\mathfrak{H}_d}$. Any function from such a set is called a representative of this equivalence class (the entire equivalence class can be obtained using the action of the elements of the group $\mathfrak{GU}(V_n)\mathfrak{H}_d$ on this function). The compilation of the classification of the set $\mathcal{F}_n$ with respect to a group $\mathfrak{GU}(V_n)\mathfrak{H}_d$ is understood as the compilation of a list that includes one representative of each of the existing equivalence classes. An example of such classification can be found in [13, 2].

The Appendix provides a classification of Boolean functions of 5 variables with respect to the group $\mathfrak{GU}(V_5)\mathfrak{H}_0$. For each of these functions, the values of parameters are given, which coincide for all functions from the corresponding equivalence class. Namely, the power of the equivalence class, the algebraic degree, the nonlinearity and the number of unbalanced planes of dimensions $4, 3, 2, 1$. . From the definition of a group $\mathfrak{GU}(V_n)\mathfrak{H}_0$ it follows that the same equivalence class contains the same number of functions of weight w and $2^n - w$. Therefore, the entire classification is divided into 17 tables, each of which includes equivalence classes containing weight functions w and $2^5 - w$ for $w = 0, 1, \ldots, 16$. The tables show global and local numbering. The minimum and maximum values in the columns containing the numbers of unbalanced planes of various dimensions are in bold. The representative function itself is specified in the form of a truth table written in hexadecimal notation, and the function values written in the lexicographical order of its input arguments from left to

right:

$$f(00000)f(00001)f(00010)\dots f(11101)f(11110)f(11111).$$

For example, the function $f = 80018003$ takes on a value 1 only on vectors $(00000), (01111), (10000), (11110), (11111)$.

Let us note some features of the obtained classification. It contains 210 functions, 38 of which are balanced. Among the balanced functions for the function 0000FFFF, which is a representative of the class of affine functions, the minimum number of unbalanced planes is achieved at the same time for all dimensions under consideration. In this case, for any affine function, if the plane is unbalanced, then the function is a constant on it. The function for which the number of unbalanced planes is maximal for each dimension does not exist (for the function 011F37BC, the maximum is reached for dimensions $4, 2$ and 1).

7 Numerical Characteristics of Some Boolean Functions

7.1 Numerical characteristics of Bent functions of 6 variables

In [16] the classification of Bent functions of 6 variables under the group $\mathfrak{GU}(V_6)\mathfrak{H}_1$ was proposed. The classification consists of 4 equivalence classes, for each of which below are the quantity of unbalanced planes of dimensions $5, 4, 3, 2, 1$.

№	f	$\deg f$	$\mathrm{nl}(f)$	5	4	3	2	1
1	111E111E111EEEE1	2	28	63	1659	5175	6636	1008
2	005533660F5A3C96	3	28	63	1659	7415	6636	1008
3	033055660C3FA569	3	28	63	1659	7975	6636	1008
4	066A503C09655FCC	3	28	63	1659	8255	6636	1008

Note that for these Bent functions, the addition of arbitrary linear functions does not change the number of unbalanced planes of any dimension. However, an arbitrary Boolean function does not have this property.

7.2 Numerical Characteristics of filter function of LILI-128

Let $n = 10$, $f_d \in \mathcal{F}_n$ be the filter function of LILI-128 cipher [8]. This function is balanced ($\mathrm{wt}(f_d) = 512$) and the nonlinearity $\mathrm{nl}(f_d)$ is equal to

480.

The series of practical experiments of random planes of various dimensions generation and calculation the weight of the function on them for the function f_d were carried out. The results are shown in the following table. The first column contains the dimension of the plane, the second column contains the theoretical boundaries of the function weight on the planes of the corresponding dimension, according to Theorem 6. The third column contains the boundaries of the function weight on the planes obtained as a result of direct counting for specific planes generated during the experiment, the fourth column shows the number of experiments.

dim	theoretical wt $(f_d\|_{L\oplus a})$	experimental wt $(f_d\|_{L\oplus a})$	number of tests
9	256 ± 16	256 ± 16	2^{10}
8	128 ± 24	128 ± 24	2^{18}
7	64 ± 28	64 ± 20	2^{18}
6	32 ± 30	32 ± 20	2^{22}
5	16 ± 16	16 ± 16	2^{22}

The results given in the table show that the inequality from the Theorem 6 is best possible in the general case (the boundaries are reached for the dimensions 9 and 8).

8 Conclusion

The results obtained in this study mainly relate to the properties of Boolean functions, and not the cryptographic properties of filter generators. The clear need for the results obtained from cryptanalysis was explained in detail in Section 1. At the same time, this study does not demonstrate the application of these results to obtain applied cryptographic conclusions about the resistance of the filter generator. This is the main unresolved issue that the authors intend to make the main topic of further research.

However, there are problems concerning the specific features of the structure of Boolean functions in terms of the location of their units on the planes of the space V_n. Here are some of them.

– The issue of whether affine functions attain the minimum number of unbalanced planes for all dimensions k, $2 \leqslant k \leqslant n-1$ remains open.

– Refinement of the inequality from Theorem 4, which is required to obtain more accurate estimates of the number of unbalanced planes for dimensions that are not available for complete enumeration, is of considerable interest.

– In addition, there are interesting relations connecting the weight of a function on planes or the number of unbalanced planes with various cryptographic parameters of Boolean functions, for example, with an algebraic degree.

References

[1] Alekseev E.K., Kuschinskaya L.A., "Generalization of one method of key recovery of a filter generator", *Diskr. Mat.*, **29**:4 (2017), 3 – 27.

[2] Alekseev E.K., Karelina E.K., "Classification of correlation-immune and minimal correlation-immune Boolean functions of 4 and 5 variables", *Diskr. Mat.*, **27**:1 (2015), 22 – 33.

[3] Alekseev E.K., "On some measures of nonlinearity for Boolean functions", *Prikl. Diskr. Mat.*, **12**:2 (2011), 5 – 16.

[4] Alekseev E.K., Kushchinskaya L.A., "On the construction of generalized approximations for one filter generator key recovery method", *Pre-proceedings*, 6rd Workshop on Current Trends in Cryptology (CTCrypt 2017), 2017, 247 – 259.

[5] Alekseev E.K., "On an attack on the generator with a filter function close to algebraic degenerated function", *Sbornik statey molodykh uchenykh*, **8** (2011), 114 – 123.

[6] Logachev O.A., Sal'nikov A.A., Smyshlyaev S.V., Yashchenko V.V., *Boolean functions in coding theory and cryptography*, LENAND, Moscow, 2015, 584 pp.

[7] Dawson E., Wu C.K., "Construction of correlation immune boolean functions", *LNCS*, **1334**, 1997, 170 – 180

[8] Dawson E., Clark A., Golic J., Millan W., Penna L., Simpson L., "The LILI-128 keystream generator", Proc. of first NESSIE workshop, 2000, http://www.cryponessie.org.

[9] Meier W., Staffelbach O., "Fast correlation attacks on certain stream cipher", J. Cryptology, **1**:3 (1989), 159 – 176.

[10] Siegenthaler T., "Decrypting a class of stream cipher using ciphertext only", IEEE Trans. Computers, **C-34**:1 (1985), 81 – 85.

[11] Courtois N., Meier W., "Algebraic attacks on stream ciphers with linear feedback", *LNCS*, EUROCRYPT'03, **2656**, 2003, 346 – 359.

[12] Glukhov M.M., Elizarov V.P., Nechaev A.A., *Algebra: Textbook*, M.: Gelious ARV, 2003, 416 pp.

[13] Strazdin I.E., *Affine classification of Boolean functions of five variables Automat Control Comput. Sci*, **9**, 1975

[14] Hell M., Johansson T., Meier W., "Grain: a stream cipher for constrained environments", Int. J. Wireless and Mobile Computing 2(1), 2007, 86 – 93

[15] Hawkes P., Rose G., "Primitive specification and supporting documentation for SOBER-t16 submission to NESSIE", In Proceedings of the First Open NESSIE Workshop, 2000

[16] Rothaus O.S., "On bent functions", J. Comb. Theory, **20** (1979), 300 – 305

A Appendix 1. Quantities of unbalanced planes of functions of 5 variables

Function of the weight of 0 and 32

№	$№_{\text{local}}$	f	$\lvert\{f\}_{\mathfrak{GL}(V_5)\mathfrak{H}_0}\rvert$	$\deg f$	$\mathrm{nl}(f)$	4	3	2	1
1	1	00000000	2	0	0	62	620	1240	496

Function of the weight of 1 and 31

№	$№_{\text{local}}$	f	$\lvert\{f\}_{\mathfrak{GL}(V_5)\mathfrak{H}_0}\rvert$	$\deg f$	$\mathrm{nl}(f)$	4	3	2	1
2	1	00000001	64	5	1	62	620	1240	465

Function of the weight of 2 and 30

№	$№_{\text{local}}$	f	$\lvert\{f\}_{\mathfrak{GL}(V_5)\mathfrak{H}_0}\rvert$	$\deg f$	$\mathrm{nl}(f)$	4	3	2	1
3	1	00000003	992	4	2	62	620	1225	436

Function of the weight of 3 and 29

№	$№_{\text{local}}$	f	$\lvert\{f\}_{\mathfrak{GL}(V_5)\mathfrak{H}_0}\rvert$	$\deg f$	$\mathrm{nl}(f)$	4	3	2	1
4	1	00000007	9920	5	3	62	620	1198	409

Function of the weight of 4 and 28

№	$№_{\text{local}}$	f	$\lvert\{f\}_{\mathfrak{GL}(V_5)\mathfrak{H}_0}\rvert$	$\deg f$	$\mathrm{nl}(f)$	4	3	2	1
5	1	0000000F	2480	3	4	62	613	1156	384
6	2	00000017	69440	4	4	62	619	1162	384

Function of the weight of 5 and 27

№	$№_{\text{local}}$	f	$\lvert\{f\}_{\mathfrak{GL}(V_5)\mathfrak{H}_0}\rvert$	$\deg f$	$\mathrm{nl}(f)$	4	3	2	1
7	1	0000001F	69440	5	5	62	614	1114	361
8	2	00000117	333312	5	5	62	615	1120	361

Function of the weight of 6 and 26

№	№$_{\text{local}}$	f	$\lvert\{f\}_{\mathfrak{GU}(V_5)\mathfrak{H}_0}\rvert$	$\deg f$	$\mathrm{nl}\,(f)$	4	3	2	1
9	1	0000003F	34720	4	6	**62**	**602**	**1057**	340
10	2	0000011F	833280	4	6	**62**	**609**	1069	340
11	3	00000356	55552	3	6	**62**	605	**1075**	340
12	4	00010117	888832	4	6	**62**	605	**1075**	340

Function of the weight of 7 and 25

№	№$_{\text{local}}$	f	$\lvert\{f\}_{\mathfrak{GU}(V_5)\mathfrak{H}_0}\rvert$	$\deg f$	$\mathrm{nl}\,(f)$	4	3	2	1
13	1	0000007F	9920	5	7	**62**	**578**	**988**	321
14	2	0000013F	833280	5	7	**62**	597	1012	321
15	3	00000357	555520	5	7	**62**	**603**	1018	321
16	4	0001011F	4444160	5	7	**62**	594	1024	321
17	5	00010356	888832	5	7	**62**	585	**1030**	321

Function of the weight of 8 and 24

№	№$_{\text{local}}$	f	$\lvert\{f\}_{\mathfrak{GU}(V_5)\mathfrak{H}_0}\rvert$	$\deg f$	$\mathrm{nl}\,(f)$	4	3	2	1
18	1	000000FF	1240	2	8	**59**	**536**	**904**	304
19	2	0000017F	238080	4	8	61	578	946	304
20	3	0000033F	104160	3	8	61	574	952	304
21	4	0000035F	1249920	4	8	61	**590**	958	304
22	5	0001013F	6666240	4	8	**62**	577	970	304
23	6	00010357	8888320	4	8	**62**	578	976	304
24	7	00030355	555520	3	8	**62**	578	976	304
25	8	00030356	3333120	4	8	**62**	564	**982**	304

Function of the weight of 9 and 23

№	№$_{\text{local}}$	f	$\lvert\{f\}_{\mathfrak{GU}(V_5)\mathfrak{H}_0}\rvert$	$\deg f$	$\mathrm{nl}\,(f)$	4	3	2	1
26	1	000001FF	29760	5	7	**60**	550	**868**	289
27	2	0000037F	833280	5	7	**62**	569	892	289
28	3	00000777	555520	5	7	**62**	**575**	898	289
29	4	0001017F	1904640	5	9	**60**	557	910	289
30	5	0001033F	1666560	5	9	61	548	916	289
31	6	0001035F	19998720	5	9	61	559	922	289

Continued on the next page

Function of the weight of 9 and 23

№	$№_{\text{local}}$	f	$\lvert\{f\}_{\mathfrak{GU}(V_5)\mathfrak{H}_0}\rvert$	$\deg f$	$\mathrm{nl}(f)$	4	3	2	1
32	7	00030357	13332480	5	9	**62**	555	928	289
33	8	00030567	13332480	5	9	**62**	551	**934**	289
34	9	00031556	4444160	5	9	**62**	**536**	**934**	289

Function of the weight of 10 and 22

№	$№_{\text{local}}$	f	$\lvert\{f\}_{\mathfrak{GU}(V_5)\mathfrak{H}_0}\rvert$	$\deg f$	$\mathrm{nl}\,(f)$	4	3	2	1
35	1	000003FF	104160	4	6	60	542	**817**	276
36	2	0000077F	833280	4	6	**62**	**549**	829	276
37	3	0000177E	55552	3	6	**62**	545	835	276
38	4	000101FF	238080	4	8	61	536	841	276
39	5	0001037F	13332480	4	8	61	535	865	276
40	6	00010777	8888320	4	8	**62**	536	871	276
41	7	0003033F	166656	4	10	**57**	500	865	276
42	8	0003035F	9999360	4	10	59	527	877	276
43	9	0003056F	3333120	3	10	59	533	883	276
44	10	00030577	39997440	4	10	60	533	883	276
45	11	00031557	4444160	4	10	59	532	877	276
46	12	0003155B	39997440	4	10	61	524	889	276
47	13	00035556	634880	3	10	59	508	883	276
48	14	0003555A	1666560	4	10	61	**494**	889	276
49	15	00071356	5332992	4	10	**62**	530	**895**	276

Function of the weight of 11 and 21

№	$№_{\text{local}}$	f	$\lvert\{f\}_{\mathfrak{GU}(V_5)\mathfrak{H}_0}\rvert$	$\deg f$	$\mathrm{nl}\,(f)$	4	3	2	1
50	1	000007FF	208320	5	5	60	514	**754**	265
51	2	0000177F	333312	5	5	**62**	**515**	760	265
52	3	000103FF	1666560	5	7	61	512	802	265
53	4	0001077F	13332480	5	7	60	509	814	265
54	5	0001177E	888832	5	7	**62**	500	820	265
55	6	0003037F	3333120	5	9	59	491	826	265
56	7	0003057F	19998720	5	9	61	507	832	265
57	8	00030777	26664960	5	9	58	503	838	265
58	9	0003155F	39997440	5	9	58	508	838	265
59	10	0003156F	39997440	5	9	60	504	844	265
60	11	00035557	634880	5	9	**62**	**515**	820	265
61	12	0003555B	13332480	5	9	60	494	844	265
62	13	0007133D	19998720	5	11	**57**	490	850	265
63	14	00071357	63995904	5	11	**57**	505	850	265

Continued on the next page

Function of the weight of 11 and 21

№	$№_{\text{local}}$	f	$\lvert\{f\}_{\mathfrak{GU}(V_5)\mathfrak{H}_0}\rvert$	$\deg f$	$\mathrm{nl}\,(f)$	4	3	2	1
64	15	0007333C	333312	5	11	**57**	**430**	850	265
65	16	00073356	13332480	5	11	59	506	**856**	265

Function of the weight of 12 and 20

№	$№_{\text{local}}$	f	$\lvert\{f\}_{\mathfrak{GU}(V_5)\mathfrak{H}_0}\rvert$	$\deg f$	$\mathrm{nl}(f)$	4	3	2	1
66	1	00000FFF	17360	3	4	56	461	**676**	256
67	2	000017FF	208320	4	4	**60**	467	682	256
68	3	000107FF	3333120	4	6	59	479	754	256
69	4	0001177F	5332992	4	6	57	475	760	256
70	5	000303FF	416640	3	8	55	457	772	256
71	6	000305FF	2499840	4	8	59	483	778	256
72	7	0003077F	19998720	4	8	59	470	790	256
73	8	0003157F	53329920	4	8	57	481	796	256
74	9	0003177D	6666240	3	8	51	481	796	256
75	10	0003177E	6666240	4	8	55	467	802	256
76	11	0003555F	6666240	4	8	59	**490**	790	256
77	12	0003556F	19998720	4	8	55	477	802	256
78	13	00070777	4444160	4	10	59	467	802	256
79	14	0007133F	9999360	4	10	59	477	802	256
80	15	0007135F	79994880	4	10	57	483	808	256
81	16	0007137D	79994880	4	10	55	474	814	256
82	17	0007333D	6666240	4	10	55	449	814	256
83	18	00073357	79994880	4	10	55	479	814	256
84	19	00073567	53329920	4	10	53	485	820	256
85	20	000F333C	27776	2	12	**47**	**335**	820	256
86	21	000F3355	1666560	3	12	**47**	455	820	256
87	22	000F3356	4999680	4	12	51	481	**826**	256
88	23	00171B56	5332992	3	12	**47**	485	820	256

Function of the weight of 13 and 19

№	$№_{\text{local}}$	f	$\lvert\{f\}_{\mathfrak{GU}(V_5)\mathfrak{H}_0}\rvert$	$\deg f$	$\mathrm{nl}\,(f)$	4	3	2	1
89	1	00001FFF	69440	5	3	56	400	**598**	249
90	2	00010FFF	277760	5	5	**59**	436	694	249
91	3	000117FF	3333120	5	5	53	437	700	249
92	4	000307FF	4999680	5	7	57	434	742	249
93	5	000315FF	6666240	5	7	51	455	748	249
94	6	0003177F	26664960	5	7	54	446	754	249
95	7	0003557F	13332480	5	7	54	461	754	249
96	8	0003567F	13332480	5	7	48	452	760	249
97	9	0007077F	4444160	5	9	56	427	760	249
98	10	0007137F	79994880	5	9	53	454	772	249
99	11	00071777	53329920	5	9	56	455	778	249
100	12	0007177E	17776640	5	9	50	436	784	249
101	13	0007333F	3333120	5	9	**59**	453	766	249
102	14	0007335F	39997440	5	9	56	460	778	249
103	15	00073377	19998720	5	9	53	**469**	772	249
104	16	0007337D	39997440	5	9	50	446	784	249
105	17	0007356F	13332480	5	9	50	456	784	249
106	18	00073577	159989760	5	9	50	461	784	249
107	19	000F333D	555520	5	11	53	**392**	790	249
108	20	000F3357	19998720	5	11	53	452	790	249
109	21	000F3567	13332480	5	11	**47**	468	**796**	249
110	22	0017173D	13332480	5	11	53	442	790	249
111	23	00171B3D	39997440	5	11	**47**	463	**796**	249
112	24	00171B57	106659840	5	11	53	462	790	249

Function of the weight of 14 and 18

№	$№_{\text{local}}$	f	$\lvert\{f\}_{\mathfrak{GU}(V_5)\mathfrak{H}_0}\rvert$	$\deg f$	$\mathrm{nl}\,(f)$	4	3	2	1
113	1	00003FFF	14880	4	2	48	**312**	**505**	244
114	2	00011FFF	1111040	4	4	47	389	637	244
115	3	00030FFF	416640	4	6	51	384	697	244
116	4	000317FF	9999360	4	6	49	411	709	244
117	5	000355FF	1666560	4	6	49	431	709	244
118	6	000356FF	1666560	3	6	**33**	427	715	244
119	7	0003577F	13332480	4	6	48	427	715	244
120	8	000707FF	1666560	4	8	49	378	721	244
121	9	000713FF	9999360	4	8	47	425	733	244
122	10	0007177F	53329920	4	8	49	422	745	244
123	11	0007337F	39997440	4	8	49	437	745	244
124	12	0007357F	79994880	4	8	48	438	751	244
125	13	00073777	53329920	4	8	48	448	751	244
126	14	0007377D	53329920	4	8	47	434	757	244
127	15	000F1777	13332480	4	10	51	434	757	244
128	16	000F177E	4444160	3	10	35	410	763	244
129	17	000F333F	277760	4	10	**53**	422	745	244
130	18	000F335F	9999360	4	10	51	449	757	244
131	19	000F337D	4999680	4	10	49	416	769	244
132	20	000F356F	3333120	3	10	35	**455**	763	244
133	21	000F3577	39997440	4	10	49	446	769	244
134	22	0017173F	19998720	4	10	51	429	757	244
135	23	0017177E	6666240	4	10	49	396	769	244
136	24	00171B3F	19998720	3	10	35	445	763	244
137	25	00171B5F	159989760	4	10	50	445	763	244
138	26	00171B7D	79994880	4	10	49	436	769	244
139	27	00171F3D	159989760	4	10	49	441	769	244
140	28	00173D3D	13332480	4	10	50	**455**	763	244
141	29	00173D5B	79994880	4	10	48	452	775	244
142	30	011717BC	6666240	4	12	47	443	**781**	244

Function of the weight of 15 and 17

№	$№_{\text{local}}$	f	$\lvert\{f\}_{\mathfrak{GU}(V_5)\mathfrak{H}_0}\rvert$	$\deg f$	$\mathrm{nl}\,(f)$	4	3	2	1
143	1	00007FFF	1984	5	1	**32**	**200**	**400**	241
144	2	00013FFF	238080	5	3	39	333	568	241
145	3	00031FFF	1666560	5	5	39	369	664	241
146	4	000357FF	6666240	5	5	43	395	670	241
147	5	00070FFF	277760	5	7	35	308	688	241
148	6	000717FF	13332480	5	7	41	387	712	241
149	7	000733FF	4999680	5	7	41	407	712	241
150	8	000735FF	9999360	5	7	45	413	718	241
151	9	0007377F	79994880	5	7	44	419	724	241
152	10	00077777	4444160	5	7	44	434	724	241
153	11	0007777B	13332480	5	7	48	425	730	241
154	12	000F177F	13332480	5	9	42	405	730	241
155	13	000F337F	4999680	5	9	41	426	736	241
156	14	000F357F	19998720	5	9	45	437	742	241
157	15	000F3777	13332480	5	9	44	**443**	748	241
158	16	000F377D	13332480	5	9	48	424	754	241
159	17	0017177F	13332480	5	9	41	396	736	241
160	18	00171B7F	79994880	5	9	45	422	742	241
161	19	00171F3F	79994880	5	9	45	432	742	241
162	20	00171F77	159989760	5	9	44	428	748	241
163	21	00171F7E	53329920	5	9	48	414	754	241
164	22	00173D3F	79994880	5	9	44	438	748	241
165	23	00173D5F	159989760	5	9	48	434	754	241
166	24	00173D7E	39997440	5	9	48	439	754	241
167	25	001F373D	13332480	5	11	47	425	760	241
168	26	001F3757	106659840	5	11	47	440	760	241
169	27	0117177E	888832	5	11	47	335	760	241
170	28	011717BD	39997440	5	11	47	415	760	241
171	29	01171BD7	63995904	5	11	47	435	760	241
172	30	01171BFC	39997440	5	11	**51**	441	**766**	241

Function of the weight of 16

№	$№_{\text{local}}$	f	$\lvert\{f\}_{\mathfrak{GL}(V_5)\mathfrak{H}_0}\rvert$	$\deg f$	$\mathrm{nl}(f)$	4	3	2	1
173	1	0000FFFF	62	1	0	**2**	**60**	**280**	240
174	2	00017FFF	15872	4	2	32	270	**490**	240
175	3	00033FFF	59520	3	4	16	326	616	240
176	4	00035FFF	833280	4	4	40	362	622	240
177	5	00071FFF	555520	4	6	32	342	682	240
178	6	000737FF	9999360	4	6	40	394	694	240
179	7	0007777F	8888320	4	6	44	410	700	240
180	8	000F0FFF	8680	2	8	**8**	**204**	664	240
181	9	000F17FF	1666560	4	8	36	366	706	240
182	10	000F33FF	312480	3	8	20	402	712	240
183	11	000F35FF	1249920	4	8	44	418	718	240
184	12	000F377F	9999360	4	8	42	425	730	240
185	13	000F7777	555520	3	8	26	**446**	736	240
186	14	000F777B	1666560	4	8	50	432	742	240
187	15	001717FF	833280	3	8	20	362	712	240
188	16	00171BFF	4999680	4	8	44	398	718	240
189	17	00171F7F	53329920	4	8	42	410	730	240
190	18	00173D7F	39997440	4	8	46	426	736	240
191	19	00173F3F	9999360	4	8	42	425	730	240
192	20	00173F5F	39997440	4	8	46	426	736	240
193	21	00173F7D	9999360	3	8	26	426	736	240
194	22	00173F7E	19998720	4	8	50	422	742	240
195	23	00177E7E	1666560	4	8	50	432	742	240
196	24	001F1F77	13332480	4	10	40	422	742	240
197	25	001F373F	9999360	4	10	40	432	742	240
198	26	001F375F	79994880	4	10	44	438	748	240
199	27	001F377D	39997440	4	10	48	429	754	240
200	28	0117177F	444416	4	10	32	360	730	240
201	29	011717BF	19998720	4	10	40	412	742	240
202	30	011717FE	6666240	4	10	48	384	754	240
203	31	01171BDF	53329920	4	10	44	428	748	240
204	32	01171BFD	79994880	4	10	48	424	754	240

Continued on the next page

Function of the weight of 16

№	$№_{\text{local}}$	f	$\lvert\{f\}_{\mathfrak{GL}(V_5)\mathfrak{H}_0}\rvert$	$\deg f$	$\mathrm{nl}(f)$	4	3	2	1
205	33	01171FF6	39997440	4	10	48	429	754	240
206	34	01173DED	31997952	4	10	52	440	760	240
207	35	011F377C	1666560	3	12	32	400	760	240
208	36	011F37BC	1666560	4	12	**56**	436	**766**	240
209	37	011F37D6	5332992	3	12	32	440	760	240
210	38	033F566A	27776	2	12	32	240	760	240

B Appendix 2. Plane weight characteristics of function of 5 variables

Function of the weight of 16 with plane weight characteristic

№	f	dim	0	1	2	3	4	5	6	7	8
1	0000FFFF	1	256	240	-	-	-	-	-	-	-
		2	960	0	280	-	-	-	-	-	-
		3	560	0	0	0	60	-	-	-	-
		4	60	0	0	0	0	0	0	0	2
2	00017FFF	1	256	240	-	-	-	-	-	-	-
		2	750	280	210	-	-	-	-	-	-
		3	350	210	0	30	30	-	-	-	-
		4	30	30	0	0	0	0	0	2	0
3	00033FFF	1	256	240	-	-	-	-	-	-	-
		2	624	448	168	-	-	-	-	-	-
		3	294	224	56	32	14	-	-	-	-
		4	46	0	14	0	0	0	2	0	0
4	00035FFF	1	256	240	-	-	-	-	-	-	-
		2	618	456	166	-	-	-	-	-	-
		3	258	272	44	32	14	-	-	-	-
		4	22	32	6	0	0	0	2	0	0
5	00071FFF	1	256	240	-	-	-	-	-	-	-
		2	558	536	146	-	-	-	-	-	-
		3	278	210	96	30	6	-	-	-	-
		4	30	24	0	6	0	2	0	0	0

Continued on the next page

Function of the weight of 16 with plane weight characteristic

№	f	dim	0	1	2	3	4	5	6	7	8
6	000737FF	1	256	240	-	-	-	-	-	-	-
		2	546	552	142	-	-	-	-	-	-
		3	226	276	84	28	6	-	-	-	-
		4	22	28	8	2	0	2	0	0	0
7	0007777F	1	256	240	-	-	-	-	-	-	-
		2	540	560	140	-	-	-	-	-	-
		3	210	294	84	26	6	-	-	-	-
		4	18	30	12	0	0	2	0	0	0
8	000F0FFF	1	256	240	-	-	-	-	-	-	-
		2	576	512	152	-	-	-	-	-	-
		3	416	0	192	0	12	-	-	-	-
		4	54	0	0	0	8	0	0	0	0
9	000F17FF	1	256	240	-	-	-	-	-	-	-
		2	534	568	138	-	-	-	-	-	-
		3	254	222	120	18	6	-	-	-	-
		4	26	28	0	4	4	0	0	0	0
10	000F33FF	1	256	240	-	-	-	-	-	-	-
		2	528	576	136	-	-	-	-	-	-
		3	218	256	136	0	10	-	-	-	-
		4	42	0	16	0	4	0	0	0	0
11	000F35FF	1	256	240	-	-	-	-	-	-	-
		2	522	584	134	-	-	-	-	-	-
		3	202	288	108	16	6	-	-	-	-
		4	18	32	8	0	4	0	0	0	0
12	000F377F	1	256	240	-	-	-	-	-	-	-
		2	510	600	130	-	-	-	-	-	-
		3	195	290	116	14	5	-	-	-	-
		4	20	28	8	4	2	0	0	0	0
13	000F7777	1	256	240	-	-	-	-	-	-	-
		2	504	608	128	-	-	-	-	-	-
		3	174	312	120	8	6	-	-	-	-
		4	36	0	24	0	2	0	0	0	0
14	000F777B	1	256	240	-	-	-	-	-	-	-

Continued on the next page

Function of the weight of 16 with plane weight characteristic

№	f	dim	0	1	2	3	4	5	6	7	8
		2	498	616	126	-	-	-	-	-	-
		3	188	292	124	12	4	-	-	-	-
		4	12	32	16	0	2	0	0	0	0
15	001717FF	1	256	240	-	-	-	-	-	-	-
		2	528	576	136	-	-	-	-	-	-
		3	258	224	104	32	2	-	-	-	-
		4	42	0	16	0	4	0	0	0	0
16	00171BFF	1	256	240	-	-	-	-	-	-	-
		2	522	584	134	-	-	-	-	-	-
		3	222	272	92	32	2	-	-	-	-
		4	18	32	8	0	4	0	0	0	0
17	00171F7F	1	256	240	-	-	-	-	-	-	-
		2	510	600	130	-	-	-	-	-	-
		3	210	278	104	26	2	-	-	-	-
		4	20	28	8	4	2	0	0	0	0
18	00173D7F	1	256	240	-	-	-	-	-	-	-
		2	504	608	128	-	-	-	-	-	-
		3	194	296	104	24	2	-	-	-	-
		4	16	30	12	2	2	0	0	0	0
19	00173F3F	1	256	240	-	-	-	-	-	-	-
		2	510	600	130	-	-	-	-	-	-
		3	195	290	116	14	5	-	-	-	-
		4	20	28	8	4	2	0	0	0	0
20	00173F5F	1	256	240	-	-	-	-	-	-	-
		2	504	608	128	-	-	-	-	-	-
		3	194	296	104	24	2	-	-	-	-
		4	16	30	12	2	2	0	0	0	0
21	00173F7D	1	256	240	-	-	-	-	-	-	-
		2	504	608	128	-	-	-	-	-	-
		3	194	296	104	24	2	-	-	-	-
		4	36	0	24	0	2	0	0	0	0
22	00173F7E	1	256	240	-	-	-	-	-	-	-
		2	498	616	126	-	-	-	-	-	-

Continued on the next page

Function of the weight of 16 with plane weight characteristic

№	f	dim	0	1	2	3	4	5	6	7	8
		3	198	284	116	20	2	-	-	-	-
		4	12	32	16	0	2	0	0	0	0
23	00177E7E	1	256	240	-	-	-	-	-	-	-
		2	498	616	126	-	-	-	-	-	-
		3	188	292	124	12	4	-	-	-	-
		4	12	32	16	0	2	0	0	0	0
24	001F1F77	1	256	240	-	-	-	-	-	-	-
		2	498	616	126	-	-	-	-	-	-
		3	198	284	116	20	2	-	-	-	-
		4	22	24	8	8	0	0	0	0	0
25	001F373F	1	256	240	-	-	-	-	-	-	-
		2	498	616	126	-	-	-	-	-	-
		3	188	292	124	12	4	-	-	-	-
		4	22	24	8	8	0	0	0	0	0
26	001F375F	1	256	240	-	-	-	-	-	-	-
		2	492	624	124	-	-	-	-	-	-
		3	182	302	116	18	2	-	-	-	-
		4	18	26	12	6	0	0	0	0	0
27	001F377D	1	256	240	-	-	-	-	-	-	-
		2	486	632	122	-	-	-	-	-	-
		3	191	286	124	18	1	-	-	-	-
		4	14	28	16	4	0	0	0	0	0
28	0117177F	1	256	240	-	-	-	-	-	-	-
		2	510	600	130	-	-	-	-	-	-
		3	260	210	120	30	0	-	-	-	-
		4	30	20	12	0	0	0	0	0	0
29	011717BF	1	256	240	-	-	-	-	-	-	-
		2	498	616	126	-	-	-	-	-	-
		3	208	276	108	28	0	-	-	-	-
		4	22	24	8	8	0	0	0	0	0
30	011717FE	1	256	240	-	-	-	-	-	-	-
		2	486	632	122	-	-	-	-	-	-
		3	236	122	144	18	0	-	-	-	-

Continued on the next page

Function of the weight of 16 with plane weight characteristic

№	f	dim	0	1	2	3	4	5	6	7	8
		4	14	28	16	4	0	0	0	0	0
31	01171BDF	1	256	240	-	-	-	-	-	-	-
		2	492	624	124	-	-	-	-	-	-
		3	192	294	108	26	0	-	-	-	-
		4	18	26	12	6	0	0	0	0	0
32	01171BFD	1	256	240	-	-	-	-	-	-	-
		2	486	632	122	-	-	-	-	-	-
		3	196	282	120	22	0	-	-	-	-
		4	14	28	16	4	0	0	0	0	0
33	01171FF6	1	256	240	-	-	-	-	-	-	-
		2	486	632	122	-	-	-	-	-	-
		3	191	286	124	18	1	-	-	-	-
		4	14	28	16	4	0	0	0	0	0
34	01173DED	1	256	240	-	-	-	-	-	-	-
		2	480	640	120	-	-	-	-	-	-
		3	180	300	120	20	0	-	-	-	-
		4	10	30	20	2	0	0	0	0	0
35	011F377C	1	256	240	-	-	-	-	-	-	-
		2	480	640	120	-	-	-	-	-	-
		3	220	240	144	16	0	-	-	-	-
		4	30	32	0	0	0	0	0	0	0
36	011F37BC	1	256	240	-	-	-	-	-	-	-
		2	474	648	118	-	-	-	-	-	-
		3	184	288	132	16	0	-	-	-	-
		4	6	32	24	0	0	0	0	0	0
37	011F37D6	1	256	240	-	-	-	-	-	-	-
		2	480	640	120	-	-	-	-	-	-
		3	180	300	120	20	0	-	-	-	-
		4	30	32	0	0	0	0	0	0	0
38	033F566A	1	256	240	-	-	-	-	-	-	-
		2	480	640	120	-	-	-	-	-	-
		3	380	240	0	0	0	-	-	-	-

Continued on the next page

Function of the weight of 16 with plane weight characteristic

№	f	dim	0	1	2	3	4	5	6	7	8
		4	30	32	0	0	0	0	0	0	0

C Appendix 3. Proofs of the propositions

C.1 Proof of the Theorem 1

Proof. All further reasonings are given for an arbitrary fixed function f.

First, let us prove that if there is an unbalanced plane of dimension $k > 1$, then there are unbalanced planes of all smaller dimensions. It is enough to show that there is an unbalanced plane of dimension $k-1$. For an arbitrary plane $L \oplus a$ of dimension k, let us consider an arbitrary subspace $L' < L$ of dimension $k-1$. Then $L \oplus a = (L' \oplus a) \cup (L' \oplus a \oplus b)$, where $b \in L \backslash L'$. Since the function f is unbalanced on the plane $L \oplus a$, then at least on one of the planes $L' \oplus a$ and $L' \oplus a \oplus b$ having dimensions $k-1$, the function f will also be unbalanced.

Let us show that for any Boolean function f there is a hyperplane on which the weight of the function is different from 2^{n-2}. Each linear function l_u corresponds to a partition of the space V_n into two hyperplanes: $L_u^0 = \{x \in V_n | \langle x, u \rangle = 0\}$ and $L_u^1 = \{x \in V_n | \langle x, u \rangle = 1\}$. Let $w_u^0 = \mathrm{wt}\left(f|_{L_u^0}\right)$ and $w_u^1 = \mathrm{wt}\left(f|_{L_u^1}\right)$. Then $\mathrm{dist}\,(f, l_u) = w_u^0 + (2^{n-1} - w_u^1) = 2^{n-1} + w_u^0 - w_u^1$. Considering that $W_f(u) = 2^n - 2 \cdot \mathrm{dist}\,(f, l_u)$ (see. (1)), $W_f(u) = 2 \cdot (w_u^0 - w_u^1)$.). It follows from Parseval equality (2) that there exists a vector $u \in V_n$ such that $W_f(u) = 2 \cdot (w_u^0 - w_u^1) \neq 0$. It follows that $w_u^0 \neq w_u^1$, i.e. $w_u^0 \neq 2^{n-2}$ or $w_u^1 \neq 2^{n-2}$. Thus, there is at least one hyperplane on which the function f is unbalanced.

Since for any function f f there is a hyperplane, i.e. a plane of dimension $n-1$, on which the function is unbalanced, then, as shown above, there are also unbalanced planes of all smaller dimensions for it. □

C.2 Proof of the Theorem 1

Proof. . The number of vertices on the k-th tier of the graph G G is equal to the number of different planes of dimension k of the space V_n. There are $\prod_{i=1}^{k} \frac{2^n - 2^{i-1}}{2^k - 2^{i-1}}$ different subspaces of dimension k of the space V_n. For any two subspaces, their adjacent classes do not coincide. In this case, for any subspace there are exactly 2^{n-k} distinct adjacent classes. □

C.3 Proof of the Theorem 2

Proof. Let u u be a vertex on k-th, $k = 2, \ldots, n$, , tier of the graph G, which corresponds to the plane $L \oplus a$, $\dim L = k$. Vertices v entering into arcs (u, v), are in one-to-one correspondence with the hyperplanes of the space L. According to Statement 1 the number of such hyperplanes is

$$\prod_{i=1}^{k-1} \frac{2^k - 2^{i-1}}{2^{k-1} - 2^{i-1}} = 2^{k+1} - 2.$$

□

C.4 Proof of the Theorem 3

Proof. Let us estimate the number of incoming arcs to the vertex $v \in V$ on the k-th tier, $k = 1, \ldots, n-1$, which corresponds to the plane $L \oplus a$, where $\dim L = k, a \in V_n$. Let $\{v_1, \ldots v_k\}$ is a basis of the subspace L.

If there is an arc $(u, v) \in E$, then, by definition of the graph, the vertex u corresponds to the plane $M \oplus b$, where $\dim M = k+1, b \in V_n$ and $L \oplus a \subset M \oplus b$. Let us note that if $L \oplus a \subset M \oplus b$, then $a \in M \oplus b$, hence $M \oplus b = M \oplus a$ and $L \subset M$. Since $L \subset M$, the subspace M could be represented as: $L \cup \{L \oplus v_{k+1}\}$. Hence, one of the bases of the subspace M M is the union of the basis $\{v_1, \ldots v_k\}$ and the vector v_{k+1}.

The number of different vectors $v_{k+1} \notin L$ is $2^n - 2^k$, and addition of any vector from $L \oplus v_{k+1}$ to the basis of the subspace L leads to the same subspace M. Hence, the number of different ways to define subspace M is $\frac{2^n - 2^k}{2^k} = 2^{n-k} - 1$. □

C.5 Proof of the Theorem 4

Proof. To prove this statement, it should only be noted that the intersection of two planes is a plane. Therefore, the maximum intersection of two planes of dimension k can be a plane of dimension $k-1$, which will be a vertex adjacent to both vertices on the k-the tier. □

C.6 Proof of the Theorem 2

Proof. As already noted in Theorem 1, $W_f(u) = 2^n - 2 \cdot \mathrm{dist}\,(f, l_{u,0}) = 2 \cdot (w_{u,0} - w_{u,1})$, where $w_u^0 = \mathrm{wt}\left(f|_{L_u^0}\right)$ and $w_u^1 = \mathrm{wt}\left(f|_{L_u^1}\right)$, while $L_u^b = \{x \in V_n | \langle x, u \rangle = b\}$, $b \in \{0, 1\}$. Since the function f is balanced, then $w_u^0 + w_u^1 = 2^{n-1}$.

If $W_f(u) = 0$, then $w_u^0 = w_u^1 = 2^{n-2}$, that is, the function is balanced on hyperplanes. If $W_f(u) \neq 0$, then $w_u^0 \neq w_u^1$. Without limiting generality, let $w_u^0 = 2^{n-2} + d$, $0 < d \leqslant 2^{n-2}$. Then $w_u^1 = 2^{n-2} - d$, since $w_u^0 + w_u^1 = 2^{n-1}$. Therefore, on both hyperplanes, the weight of the function f differs from 2^{n-2} by an amount equal to d. □

C.7 Proof of the Theorem 3

Proof. To prove this statement, it is sufficient to note that any pair of vectors from the carrier of the function $f \in \mathcal{F}_n$ or from the set $V_n \backslash 1_f$ forms a plane of dimension 1, on which the function takes the value 1 or 0 respectively, that is, is constant. Since under hypothesis $|1_f| = w$, the number of such pairs is

$$\binom{w}{2} + \binom{2^n - w}{2} = \frac{w(w-1)}{2} + \frac{(2^n - w)(2^n - w - 1)}{2}.$$

□

C.8 Proof of the Theorem 4

Proof. According to Statemt 2, the number of outgoing arcs from the vertex on the k-th tier is $2 \cdot (2^k - 1)$. Let us note the following two facts. When partitioning an unbalanced plane of dimension k into two subplanes of dimension $k-1$, at least one of them is unbalanced. Two vertices on the k-th level can have at most one vertex on the $k-1$-th level adjacent to each of them (Theorem 4).

Let the number of vertices corresponding to unbalanced planes on the k-th tier be equal to N. Let us consistently estimate the «contribution» of each such vertex to the total number of unbalanced planes of dimension $k-1$. When considering one such vertex, we can say that there are at least $2^k - 1$ vertices on the $k-1$-th tier, which correspond to unbalanced planes. When considering each next vertex, it is necessary to take into account that the unbalanced planes on the $k-1$-th tier being added have already been taken into account when considering the previous vertices. Thus, consideration of vertex $v_i, 1 \leqslant i \leqslant N$,

increases the total number of unbalanced planes on $k-1$-th tier by no less than $(2^k-1)-(i-1)$ (in the worst case, one intersection occurs with each vertex taken into account when considering each of the previous $i-1$ vertices). Let $t=\min(2^k-1, N-1)$. Then, after considering all N vertices on the k-th tier for the value $S_f(k-1)$, the following estimate is valid:

$$S_f(k-1) \geqslant (2^k-1)+(2^k-1)-1+(2^k-1)-2+\ldots+(2^k-1)-t = \\ = t\cdot(2^k-1)-\sum_{i=1}^{t} i = t\cdot(2^k-1)-\frac{t\cdot(t+1)}{2}.$$

□

C.9 Proof of the Corollary 1

Proof. For a Boolean function f, such that $\mathrm{wt}(f) \leqslant 2^{n-1}$, the inequality (5) does not contradict that f can be constant on planes of dimension k if

$$\frac{\mathrm{wt}(f)}{2^{n-k}} \leqslant \left(1-\frac{1}{2^{n-k}}\right)\cdot(2^{n-k}-\mathrm{nl}(f)).$$

This holds for k, such that $k \leqslant n-\log_2\left(\frac{\mathrm{wt}(f)}{2^{n-1}-\mathrm{nl}(f)}+1\right)$. Exactly such values of k the parameter $\mathrm{AD}(f)$ can take. □

C.10 Proof of the Theorem 6

Proof. Let's use the ratio (4). Given that $W_f(u) = 2^n - 2\cdot\mathrm{dist}(f, l_u)$, we obtain the equality

$$\mathrm{wt}(f|_{a\oplus L}) = \\ = 2^{k-1} - \frac{1}{2^{n-k+1}}\cdot\left(2^n\cdot\sum_{u\in L^\perp}(-1)^{\langle u,a\rangle} - 2\cdot\sum_{u\in L^\perp}\mathrm{dist}(f,l_u)(-1)^{\langle u,a\rangle}\right).$$

With $a\notin L$, $\sum_{u\in L^\perp}(-1)^{\langle u,a\rangle}=0$ is valid, and with $a\in L$, $\sum_{u\in L^\perp}(-1)^{\langle u,a\rangle}=2^{n-k}$ is valid (see. [6]). Hence

$$\mathrm{wt}(f|_{a\oplus L}) = 2^{k-1}+\frac{1}{2^{n-k}}\cdot\sum_{u\in L^\perp}\mathrm{dist}(f,l_u)(-1)^{\langle u,a\rangle}, \text{with } a\notin L, \text{ and}$$

$$\mathrm{wt}\,(f|_L) = -2^{n-1} + 2^{k-1} + \frac{1}{2^{n-k}} \sum_{u \in L^\perp} \mathrm{dist}\,(f, l_u)\,, \text{with } a \in L.$$

Let us consider the case of $a \in L$. Let us note that $-2^{n-1} + 2^{k-1} = -2^{n-1} \cdot (1 - 1/2^{n-k})$. Since $\mathrm{dist}\,(f, l_u) \geqslant \mathrm{nl}\,(f)$, the following inequality is valid

$$\mathrm{wt}\,(f|_L) \geqslant 2^{k-1} - 2^{n-1} + \frac{\mathrm{wt}\,(f)}{2^{n-k}} + \frac{2^{n-k}-1}{2^{n-k}} \cdot \mathrm{nl}\,(f) = \\ = \frac{\mathrm{wt}\,(f)}{2^{n-k}} - \left(1 - \frac{1}{2^{n-k}}\right) \cdot (2^{n-1} - \mathrm{nl}\,(f)).$$

Given that $\mathrm{dist}\,(f, l_u) \leqslant 2^n - \mathrm{nl}\,(f)$, we obtain an inequality

$$\mathrm{wt}\,(f|_L) \leqslant 2^{k-1} - 2^{n-1} + \frac{\mathrm{wt}\,(f)}{2^{n-k}} + \left(1 - \frac{1}{2^{n-k}}\right) \cdot (2^n - \mathrm{nl}\,(f)) = \\ = \frac{\mathrm{wt}\,(f)}{2^{n-k}} + \left(1 - \frac{1}{2^{n-k}}\right) \cdot (2^{n-1} - \mathrm{nl}\,(f)).$$

Thus, the following inequality holds true

$$\left|\mathrm{wt}\,(f|_L) - \frac{\mathrm{wt}\,(f)}{2^{n-k}}\right| \leqslant \left(1 - \frac{1}{2^{n-k}}\right) \cdot (2^{n-1} - \mathrm{nl}\,(f)). \qquad (6)$$

Let us consider the case of $a \notin L$. The following relations are valid

$$\mathrm{wt}\,(f|_{a \oplus L}) = \\ = 2^{k-1} + \frac{\mathrm{wt}\,(f)}{2^{n-k}} + \frac{1}{2^{n-k}} \cdot \sum_{i=1}^{2^{n-k-1}-1} \mathrm{dist}\,(f, l_{u_i}) - \frac{1}{2^{n-k}} \cdot \sum_{j=1}^{2^{n-k-1}} \mathrm{dist}\left(f, l_{u_j}\right) \leqslant \\ \leqslant 2^{k-1} + \frac{\mathrm{wt}\,(f)}{2^{n-k}} + \frac{2^{n-k-1}-1}{2^{n-k}} \cdot (2^n - \mathrm{nl}\,(f)) - \frac{2^{n-k-1}}{2^{n-k}} \cdot \mathrm{nl}\,(f) = \\ = \frac{\mathrm{wt}\,(f)}{2^{n-k}} + \left(1 - \frac{1}{2^{n-k}}\right)(2^{n-1} - \mathrm{nl}\,(f)).$$

Similarly, the lower estimate $\mathrm{wt}\,(f|_{a \oplus L})$ is obtained, which coincides with the estimate obtained for $\mathrm{wt}\,(f|_L)$.

Thus, the required estimate is proved for an arbitrary $a \in V_n$. $\square$

C.11 Proof of the Theorem 8

Proof. Multiplication by a nondegenerate matrix and addition of an arbitrary vector from V_n takes an arbitrary plane of the space V_n to some plane of the same space, and the dimensions of these planes coincide. Thus, the number of planes on which the weight of the function f has a given value does not change.

Adding a function h of degree not greater than 0 means either adding zero, which does not change the value of the function on any of the arguments, or adding a function identically equal to 1, which leads to the inversion of all values of the function. At the same time, the planar weight characteristic of the function does not change, since it depends on the absolute value of the deviation of the weight of the function on the plane from the half cardinality of the plane. □

C.12 Proof of the Theorem 7

Proof. Any non-constant affine function $f = l_{u,a} \in \mathcal{F}_n$ takes the value a on the subspace $L = \{0^n, u\}^\perp$ of dimension $n-1$ and its opposite value $a \oplus 1$ on the plane $L' = V_n \backslash L$. Since the intersection of the planes is a plane, any other planes of dimension $n-1$ intersect with L and L' exactly on half of the vectors, therefore the function f on these planes is balanced. Thus, it is proved that $pwc_{n-1}(f) = (2^{n+1} - 4, 0, \ldots, 0, 2)$.

Let us prove a statement for dimensions k, $2 \leqslant k \leqslant n - 2$. Since L and L' do not intersect, the planes of smaller dimensions contained in them also do not intersect (in terms of the graph introduced in Section 5.1, this means that the sets of vertices that can be reached from vertices corresponding to L and L', do not intersect). The numbers of planes of dimensions $n-2, n-3, \ldots, 2$, contained in L and L', coincide and are equal to $\mathcal{P}_{n-1,n-2}, \mathcal{P}_{n-1,n-3}, \ldots, \mathcal{P}_{n-1,2}$ respectively. And the function f is constant on all these planes. Moreover, any plane on which the function f is constant is a subset of either L or L'.

To complete the proof, it remains to show that for the function f there is no plane on which f is non-constant and unbalanced. Indeed, if such a plane exists, then the cardinality of at least one of its intersections with the planes L and L' will be different from the cardinality of two (intersection powers are $2^{k-1} - w$ and $2^{k-1} + w$, where $w \neq 0$), which is contrary to the fact that these intersections are planes. □

The MOR Cryptosystem in Orthogonal and Symplectic Groups in Odd Characteristic

Ayan Mahalanobis[1], Anupam Singh[1],
Pralhad Shinde[1], and Sushil Bhunia[2]

[1]IISER Pune, India
[2]IISER Mohali, India
ayan.mahalanobis@gmail.com

Abstract

In this paper we study the MOR cryptosystem with finite Chevalley groups. There are four infinite families of finite classical Chevalley groups. These are: special linear groups $\mathrm{SL}(d,q)$, orthogonal groups $\mathrm{O}(d,q)$ and symplectic groups $\mathrm{Sp}(d,q)$. The MOR cryptosystem over $\mathrm{SL}(d,q)$ was studied by the first author, "A simple generalization of the ElGamal cryptosystem to non-abelian groups II, Communications in Algebra 40 (2012), no. 9, 3583–3596". In that case, the hardness of the MOR cryptosystem was found to be equivalent to the discrete logarithm problem in $\mathbb{F}_{q^d}$. In this paper, we show that the MOR cryptosystem over $\mathrm{Sp}(d,q)$ has the security of the discrete logarithm problem in $\mathbb{F}_{q^d}$. However, it seems likely that the security of the MOR cryptosystem for the family of orthogonal groups is $\mathbb{F}_{q^{d^2}}$.

Keywords: MOR cryptosystem, Chevalley groups, public-key cryptography.

1 Introduction

This paper is a study of the MOR cryptosystem using the orthogonal and symplectic groups over **finite fields of odd characteristic**. We only study split orthogonal groups in this paper and refer to it as orthogonal groups. This paper follows a paper by the first author [13] and uses a Gaussian elimination algorithm developed by Bhunia et. al. [3]. It is recommended that the reader reads [3, Sections 4] or [2, Appendix A] before reading this paper. In an earlier paper [13], we saw that the security of the MOR cryptosystem over $\mathrm{SL}(d,q)$ rests on the discrete logarithm problem in $\mathbb{F}_{q^d}$. Though this information is useful, however it says that there is no point in using the MOR cryptosystem over $\mathrm{SL}(d,q)$ – one might as well use the ElGamal cryptosystem over matrices of size d over $\mathbb{F}_q$. We would refer to this situation as an *unusable* MOR cryptosystem.

In this paper, we show that the MOR cryptosystem over symplectic groups is unusable. However, the MOR cryptosystem over orthogonal groups have good potential. This paper was published as a part of a book chapter [2].

This paper is in the direction of generalizing the ElGamal cryptosystem with the hope that something practical and useful will come out of this generalization. This line of research is relevant today in the light of Joux's attack on the discrete logarithm problem in finite fields of small characteristic [1, 10] and recent developments in building quantum computers. Several attempts towards *non-abelian cryptography* were made by many eminent mathematicians. To name a few, Maze et. al. [7, 8] developed SAP and Shpilrain and Zapata developed CAKE [18], both work with non-abelian structures. There is an interesting cryptosystem in the work of Climent et. al. [5] and an interesting key exchange protocol in Kahrobaei et. al. [11] and Glukhov [9].

1.1 Notations and terminology

We have used ${}^{T}X$ to denote the transpose of the matrix X. This was necessary to avoid any confusion that might arise when using X^{-1} and ${}^{T}X$ simultaneously. In this paper, we use K and $\mathbb{F}_q$ interchangeably, while each of them is **a finite field of odd characteristic**. The matrix te_{ij} is used to denote the matrix unit with t in the $(i,j)^{\text{th}}$ place and zero everywhere else. All other notations used are standard.

2 The MOR Cryptosystem

This section provides a brief introduction to MOR cryptosystem. For further details a reader can consult [14, Section 3].

2.1 The MOR cryptosystem

Let $G = \langle g_1, g_2, \ldots, g_s \rangle$ be a finite group. Let ϕ be a non-identity automorphism.

- **Public-key:** Let $\{\phi(g_i)\}_{i=1}^{s}$ and $\{\phi^{\mathtt{m}}(g_i)\}_{i=1}^{s}$ is public.
- **Private-key:** The integer $\mathtt{m}$ is private.

Encryption:
To encrypt a plaintext $\mathfrak{M} \in G$, get an arbitrary integer $\mathtt{r} \in [1, |\phi|]$ compute $\phi^{\mathtt{r}}$

and $\phi^{\mathtt{rm}}$. The ciphertext is $(\phi^{\mathtt{r}}, \phi^{\mathtt{rm}}(\mathfrak{M}))$.

Decryption:

After receiving the ciphertext $(\phi^{\mathtt{r}}, \phi^{\mathtt{rm}}(\mathfrak{M}))$, the user knows the private key $\mathtt{m}$. So she computes $\phi^{\mathtt{mr}}$ from $\phi^{\mathtt{r}}$ and then computes $\mathfrak{M}$. It is known [14, Theorem 3] that the hardness to break a MOR cryptosystem depends on the Diffie-Hellman problem in the automorphism group. In a practical implementation of a MOR cryptosystem there are two things that matter the most.

a The number of generators. As we saw, the automorphism ϕ is presented as action on generators. Larger the number of generators bigger is the public-key.

b Efficient algorithm to solve the word problem. This means, given $G = \langle g_1, g_2, \ldots, g_s \rangle$ and $g \in G$, is there an efficient algorithm to write g as word in $g_1, g_2, \ldots, g_s$? The reason of this importance is immediate – the automorphisms are presented as action on generators and if one has to compute $\phi(g)$, then the word problem must be solved.

The obvious question is: what are the right groups for the MOR cryptosystem? In this paper, we pursue a study of the MOR cryptosystem using **finite Chevalley groups** of classical type; in particular, orthogonal and symplectic groups.

3 Description of automorphisms of classical groups

This paper studies the MOR cryptosystem for split orthogonal and symplectic groups over a field of odd characteristics. As we discussed before, MOR cryptosystem is presented as action on generators of the group. Then to use an automorphism on an arbitrary element, one has to solve the word problem in that group with respect to that set of generators.

The generators and the Gaussian elimination algorithm to solve the word problem is described in Bhunia et. al. [3, Section 5]. We will be very brief here.

Let V be a vector space of dimension d over a field K of odd characteristic. Let $\beta\colon V \times V \to K$ be a bilinear form. By fixing a basis of V we can associate a matrix to β. We shall abuse the notation slightly and denote the matrix of the bilinear form by β itself. Thus $\beta(x, y) = {}^{T}x\beta y$ where x, y are column vectors. We will work with non-degenerate bilinear forms and that means $\det \beta \neq 0$.

A symmetric or skew-symmetric bilinear form β satisfies $\beta = {}^T\beta$ or $\beta = -{}^T\beta$ respectively.

Definition 1 (Orthogonal Groups). *A square matrix X of size d is called orthogonal if ${}^TX\beta X = \beta$ where β is symmetric. It is well known that the orthogonal matrices form a group known as the orthogonal group.*

Definition 2 (Symplectic Group). *A square matrix of size d is called symplectic if ${}^TX\beta X = \beta$ where β is skew-symmetric. And the set of symplectic matrices form a symplectic group.*

We write the dimension of V as $d = 2l + 1$ or $d = 2l$ for $l \geqslant 1$. We fix a basis and index it by $0, 1, 2, \ldots, l, -1, -2, \ldots, -l$ in the odd dimension and by $1, 2, \ldots, l, -1, -2, \ldots, -l$ in the even dimension. We consider the non-degenerate bilinear forms β on V given by the following matrices:

a: The odd orthogonal group: The form β is symmetric with $d = 2l + 1$ and $\beta = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 0 & I_l \\ 0 & I_l & 0 \end{pmatrix}$.

b: The symplectic group: The form β is skew-symmetric with $d = 2l$ and $\beta = \begin{pmatrix} 0 & I_l \\ -I_l & 0 \end{pmatrix}$.

c: The even orthogonal group of classical type: The form β is symmetric with $d = 2l$ and $\beta = \begin{pmatrix} 0 & I_l \\ I_l & 0 \end{pmatrix}$.

where I_l is the identity matrix of size l over K.

We now describe the automorphism group of the orthogonal and symplectic groups. This helps us in picking the right set of automorphisms for the MOR cryptosystem.

Definition 3 (Orthogonal similitude group). *The orthogonal similitude group is defined as the set of matrices X of size d as follows: $GO(d, q) = \left\{X \in GL(d, q) \mid {}^TX\beta X = \mu\beta, \mu \in \mathbb{F}_q^\times\right\}$ where $d = 2l + 1$ or $2l$ and β is of type a and c respectively.*

Definition 4 (Symplectic similitude group). *The symplectic similitude group is denoted by $GSp(2l, q) = \left\{X \in GL(2l, q) \mid {}^TX\beta X = \mu\beta, \mu \in \mathbb{F}_q^\times\right\}$ where β is of type b.*

Here μ depends on the matrix X and is called the similitude factor. The similitude factor μ defines a group homomorphism from the similitude group to $\mathbb{F}_q^\times$ and the kernel is the orthogonal group $\mathrm{O}(d,q)$ when β is symmetric and symplectic group $\mathrm{Sp}(2l,q)$ when β is skew-symmetric respectively [12, Section 12]. Note that scalar matrices λI for $\lambda \in \mathbb{F}_q^\times$ belong to the center of similitude groups. The similitude groups are analog of what $\mathrm{GL}(d,q)$ is for $\mathrm{SL}(d,q)$. For a discussion of the diagonal automorphisms of Chevalley groups we need the diagonal subgroups of the similitude groups.

Definition 5 (Diagonal group)**.** *The diagonal groups are defined to be the group of non-singular diagonal matrices in the corresponding similitude group and are as follows: in the case of $GO(2l+1,q)$ it is*

$$\left\{\mathrm{diag}(\alpha,\lambda_1,\cdots,\lambda_l,\mu\lambda_1^{-1},\cdots,\mu\lambda_l^{-1}) \mid \lambda_1,\ldots,\lambda_l,\alpha^2=\mu\in\mathbb{F}_q^\times\right\}$$

and in the case of $GO(2l,q)$ and $GSp(2l,q)$ it is

$$\left\{\mathrm{diag}(\lambda_1,\cdots,\lambda_l,\mu\lambda_1^{-1},\cdots,\mu\lambda_l^{-1}) \mid \lambda_1,\ldots,\lambda_l,\mu\in\mathbb{F}_q^\times\right\}.$$

Conjugation by these diagonal elements produce diagonal automorphisms in the respective Chevalley groups.

To build a MOR cryptosystem we need to work with the automorphism group of Chevalley groups. In this section we describe the automorphism group of classical groups following Dieudonne [6].

Conjugation Automorphisms: For $t \in G$ the map given by $g \mapsto tgt^{-1}$ is an automorphism of G, called an **inner automorphism**. More generally if N is a normal subgroup of G then the conjugation maps $n \mapsto gng^{-1}$ for $n \in N$ are called conjugation automorphisms of G.

Central Automorphisms: Let $\chi\colon G \to \mathcal{Z}(G)$ be a homomorphism to the center of the group. Then the map $g \mapsto \chi(g)g$ is an automorphism of G, known as the central automorphism. There are no non-trivial central automorphisms for perfect groups, for example, the adjoint Chevalley groups $\mathrm{SL}(l+1,K)$ and $\mathrm{Sp}(2l,K)$, $|K| \geqslant 4$ and $l \geqslant 2$. In case of orthogonal group, the center is of two elements $\{I,-I\}$. Any map χ maps $\Omega_d(K)$ to identity. This implies that there are at most four central automorphisms in this case.

Field Automorphisms: Let $f \in \mathrm{Aut}(K)$. In terms of matrices, field automorphisms amount to replacing each term of the matrix by its image under f.

Graph Automorphisms: A symmetry of Dynkin diagram induces such automorphisms. This way we get automorphisms of order 2 for $\mathrm{SL}(l+1,K)$, $l \geqslant 2$ and $O^+(2l,K)$, $l \geqslant 4$. We also get an automorphisms of order 3 for $\mathrm{O}^+(4,K)$.

In the case of $\mathrm{SL}(d,q)$ for $d \geqslant 3$, the map $x \mapsto A^{-1}{}^T x^{-1} A$ where

$$A = \begin{pmatrix} 0 & \cdots & 0 & 0 & 0 & 1 \\ 0 & \cdots & 0 & 0 & -1 & 0 \\ 0 & \cdots & 0 & 1 & 0 & 0 \\ 0 & \cdots & -1 & 0 & 0 & 0 \\ \vdots & \iddots & \vdots & \vdots & \vdots & \vdots \\ (-1)^{l-1} & \cdots & 0 & 0 & 0 & 0 \end{pmatrix}$$

explicitly describes the graph automorphism.

In the case of $\mathrm{O}(2l,q)$ for $l \geqslant 5$, the graph automorphism is given by $x \mapsto B^{-1}xB$ where B is a permutation matrix obtained from identity matrix of size $2l \times 2l$ by switching the l^{th} row and $-l^{\text{th}}$ row. This automorphism is a conjugating automorphism.

Theorem 1 (Dieudonne)**.** *Let K be a field of odd characteristic and $l \geqslant 2$.*

1. *For the group $SL(l+1,K)$ any automorphism is of the form $\iota\gamma\theta$ where ι is a conjugation automorphism defined by elements of $GL(l+1,K)$ and γ is a graph automorphism for the special linear group.*

2. *For the group $O(d,K)$ any automorphism is of the form $c_\chi\iota\theta$ where c_χ is a central automorphism, ι a conjugation automorphism by $GO(d,K)$ elements (this includes the graph automorphism of even orthogonal groups).*

3. *For the group $Sp(2l,K)$ any automorphism is of the form $\iota\theta$ where ι is a conjugation automorphism by $GSp(2l,K)$ elements.*

In all cases θ denotes a field automorphism.

In the above theorem, conjugation automorphisms are given by conjugation by elements of a larger group and it includes the group of inner automorphisms. We introduce diagonal automorphisms to make it more precise. The conjugation automorphisms ι can be written as a product of ι_g and δ where ι_g is an inner automorphism and δ is a diagonal automorphism.

Diagonal Automorphisms: The adjoint Chevalley group $\mathcal{L}(K)$ is normalized by $\hat{H}$ which is a subgroup of $\mathrm{Aut}(\mathcal{L}_K)$. Thus for $h(\chi) \in \hat{H}$ which is not in H gives an automorphism $g \to h(\chi)gh(\chi)^{-1}$ (which is not an inner automorphism). Such automorphisms are called diagonal automorphism. The explicit action on generators is as follows: $h(\chi)x_r(t)h(\chi)^{-1} = x_r(\chi(r)t)$. The group $\hat{G}$ is identified in [17, Chapter III, Section 6] with corresponding similitude group. In the case of special linear groups, diagonal automorphisms are given by conjugation by diagonal elements of $\mathrm{PGL}(l+1,q)$ on $\mathrm{PSL}(l+1,q)$. In the case of symplectic and orthogonal groups, diagonal automorphisms are given by conjugation by corresponding diagonal group elements defined in Section 5.

Let K be a finite field and $G = \mathcal{L}(K)$ be an adjoint Chevalley group over K. Steinberg described the automorphisms of these groups. We have the following theorem [4, Theorem 12.5.1] and [19],

Theorem 2 (Steinberg)**.** *Let $G = \mathcal{L}(K)$ where $\mathcal{L}$ is a simple Lie algebra and $K(= \mathbb{F}_q)$ is a finite field. Let $\phi \in \mathrm{Aut}(G)$. Then there exist inner, diagonal, graph and field automorphisms, denoted by ι, δ, γ and θ respectively, such that $\phi = \iota\delta\gamma\theta$.*

4 Security of the proposed MOR cryptosystem

The purpose of this section is to show that for a secure MOR cryptosystem over the classical Chevalley groups we have to look at automorphisms that act by conjugation, like the inner automorphisms. There are other automorphisms that also act by conjugation, like the diagonal automorphism and the graph automorphism for odd-order orthogonal groups. Then we argue what is the hardness of our security assumptions.

Let ϕ be an automorphism of one of the classical Chevalley groups G: $\mathrm{SL}(l+1,q), \mathrm{O}(2l+1,q), \mathrm{Sp}(2l,q)$, or $\mathrm{O}(2l,q)$. The automorphisms of these groups are described in Section 3. From Theorem 1 we know that $\phi = c_\chi\iota\delta\gamma\theta$ where c_χ is a central automorphism, ι is an inner automorphism, δ is a diagonal automorphism, γ is a graph automorphism and θ is a field automorphism.

The group of central automorphisms are too small and the field automorphisms reduce to a discrete logarithm in the field $\mathbb{F}_q$. So there is no benefit of using these in a MOR cryptosystem. Also there are not many graph automorphisms in classical Chevalley groups other than special linear groups and

odd-order orthogonal groups. In the odd-order orthogonal groups these automorphisms act by conjugation. Recall here that, our automorphisms are presented as action on generators. It is clear [13, Section 7] that if we can recover the conjugating matrix from the action on generators, the security is a discrete logarithm problem in $\mathbb{F}_{q^d}$ or else the security is a discrete logarithm problem in $\mathbb{F}_{q^{d^2}}$.

So from these we conclude that for a secure MOR cryptosystem we must look at automorphisms that act by conjugation, like the inner automorphisms. Inner automorphisms form a normal subgroup of $\mathrm{Aut}(G)$ and usually constitute the bulk of automorphisms. If ϕ is an inner automorphism, say $\iota_g \colon x \mapsto gxg^{-1}$, we would like to determine the conjugating element g. For the special linear group, it was done in [13]. We will follow the steps there for the present situation too. However, before we do that, let us digress briefly to observe that $G \to \mathrm{Inn}(G)$ given by $g \mapsto \iota_g$ is a surjective group homomorphism. Thus if G is generated by $g_1, g_2, \ldots, g_s$ then $\mathrm{Inn}(G)$ is generated by $\iota_{g_1}, \ldots, \iota_{g_s}$. Let $\phi \in \mathrm{Inn}(G)$. If we can find $g_j, j = 1, 2, \ldots, r$, generators, such that $\phi = \prod_{j=1}^{r} \iota_{g_j}$ then $\phi = \iota_g$ where $g = \prod_{j=1}^{r} g_j$. This implies that our problem is equivalent to solving the word problem in $\mathrm{Inn}(G)$. Note that solving word problem depends on how the group is represented and it is not invariant under group homomorphisms. Thus the algorithm described earlier to solve the word problem in the classical Chevalley groups does not help us in the present case.

4.1 Reduction of security

In this subsection, we show that for special linear and symplectic groups, the security of the MOR cryptosystem is the hardness of the discrete logarithm problem in $\mathbb{F}_{q^d}$. This is the same as saying that we can find the conjugating matrix up to a scalar multiple. We further show that the method that works for special linear and symplectic groups does not work for orthogonal groups. Let ϕ be an automorphism that works by conjugation, i.e., $\phi = \iota_g$ for some g and we try to determine g. For a description of the generators (elementary matrices) we refer to [3, Section 5].

Step 1: The automorphism ϕ is presented as action on generators $x_r(t) = I + te_r$ where $r = (i, j);\ i \neq j,\ 1 \leqslant i, j \leqslant d$ for the special linear group. For symplectic group $r = (i, j); i, j \in \{\pm 1, \pm 2, \ldots, \pm l\}$. For the even orthogonal

group, $r = (i,j); i,j \in \{\pm 1, \pm 2, \ldots, \pm l\}\,;\ \pm i \neq \pm j$. For the odd orthogonal group $r = (i,j); -l \leqslant i \leqslant l$ and $j \in \{\pm 1, \pm 2, \ldots \pm l\}\,;\ \pm i \neq \pm j$.

Thus $\phi\left(x_r\left(t\right)\right) = g(I + te_r)g^{-1} = I + tge_rg^{-1}$. This implies that we know ge_rg^{-1} for all possible r. We first claim that we can determine $N = gD$ where D is sparse, in fact, diagonal in the case of special linear and symplectic groups.

In the case of special linear groups, write $g = [G_1, \ldots, G_i, \ldots, G_d]$, where G_i are column vectors of g. Then $ge_{i,j} = [G_1, \ldots, G_d]\, e_{i,j} = [0, \ldots, 0, G_i, 0 \ldots, 0]$ where G_i is at the j^{th} place. Multiplying this with g^{-1} on the right, i.e., computing $ge_{i,j}g^{-1}$ determines G_i up to a scalar multiple d_i (say). Thus, we know $N = gD$ where $D = \operatorname{diag}(d_1, \ldots, d_{l+1})$.

For the symplectic groups, we do the similar computation with the generators $I + te_{i,-i}$ and $I + te_{-i,i}$. Write g in the column form as $[G_1, \ldots G_l, G_{-1}, \ldots, G_{-l}]$. Now,

1. $[G_1, \ldots G_l, G_{-1}, \ldots, G_{-l}]\, e_{i,-i} = [0, \ldots, 0, G_i, 0, \ldots, 0]$ where G_i is at $-i^{\text{th}}$ place. Multiplying this further with g^{-1} gives us scalar multiple of G_i, say d_iG_i.

2. $[G_1, \ldots G_l, G_{-1}, \ldots, G_{-l}]\, e_{-i,i} = [0, \ldots, 0, G_{-i}, 0, \ldots, 0]$ where G_{-i} is at i^{th} place. Multiplying this with g^{-1} gives us scalar multiple of G_{-i}, say $d_{-i}G_i$.

Thus we get $N = gD$ where D is a diagonal matrix $\operatorname{diag}(d_1, \ldots, d_l, d_{-1}, \ldots, d_{-l})$.

For the even orthogonal, write $g = [G_1, \ldots G_l, G_{-1}, \ldots, G_{-l}]$. Now computing ge_rg^{-1} gives the following equations:

1. $[G_1, \ldots G_l, G_{-1}, \ldots, G_{-l}]\, (e_{i,j} - e_{-j,-i})g^{-1} = [0, \ldots, 0, G_i, 0 \ldots, 0, G_{-j}, 0, \ldots]\, g^{-1}$ where G_i is at j^{th} place and G_{-j} is at $-i^{\text{th}}$ place. This gives us a linear combination of the columns G_i and G_{-j}.

2. $[G_1, \ldots G_l, G_{-1}, \ldots, G_{-l}]\, (e_{i,-j} - e_{j,-i})g^{-1} = [0, \ldots, 0, G_i, 0 \ldots, 0, G_j, 0, \ldots]\, g^{-1}$ where G_i is at $-j^{\text{th}}$ place and G_j is at $-i^{\text{th}}$ place. This will give us a linear combination of the columns G_i and G_j.

3. $[G_1, \ldots G_l, G_{-1}, \ldots, G_{-l}]\, (e_{-i,j} - e_{-j,i})g^{-1} = [0, \ldots, 0, G_{-i}, 0 \ldots, 0, G_{-j}, 0, \ldots]\, g^{-1}$ where G_{-i} is at j^{th} place and G_{-j} is at i^{th} place. This will give us a linear combination of the columns G_{-i} and G_{-j}.

Thus we get $N = gD$ where D is of the form $\begin{pmatrix} W & X \\ Y & Z \end{pmatrix}$ with W a diagonal matrix, Y anti-diagonal, X has first column nonzero and Z has the last column nonzero. This is not a diagonal matrix. One can do a similar computation for the odd-orthogonal group.

Step 2: Compute $N^{-1}\phi(x_r(t))N = D^{-1}g^{-1}(gx_r(t)g^{-1})gD = I + D^{-1}e_rD$ which is equivalent to computing $D^{-1}e_rD$ for $r \in \Phi$.

In the case of special linear groups we have D a diagonal. Thus by computing $D^{-1}e_{i,j}D$ we determine $d_i^{-1}d_j$ for $i \neq j$ and form a matrix $\mathrm{diag}(1, d_2^{-1}d_1, \ldots, d_l^{-1}d_1)$ and multiply this to N we get d_1g. Hence we can determine g up to a scalar matrix.

For symplectic groups, we can do similar computation as D is diagonal. First compute $D^{-1}(e_{i,j} - e_{-j,-i})D$ to get $d_i^{-1}d_j$ and $d_{-i}^{-1}d_{-j}$ for $i \neq j$. Now compute $D^{-1}e_{i,-i}D, D^{-1}e_{-i,i}D$ to get $d_id_{-i}^{-1}, d_{-i}d_i^{-1}$. We form a matrix

$$\mathrm{diag}(1, d_2^{-1}d_1, \ldots, d_l^{-1}d_1, d_{-1}^{-1}d_{-2}.d_{-2}^{-1}d_2.d_2^{-1}d_1, \ldots, d_{-l}^{-1}d_{-1}.d_{-1}^{-1}d_1)$$

and multiply it to $N = gD$ to get d_1g. Thus we can determine g up to a scalar multiple and then the attack follows [13, Section 7.1.1].

However, in the case of orthogonal groups, the matrix D is not a diagonal matrix and the above method to determine g does not work.

Remark 1. *An observant reader would ask the question: why does this attack works for the special linear and symplectic groups but not for orthogonal groups? The answer lies in a closer look at the generators (elementary matrices) for these groups.*

In the special linear groups the generators are the elementary transvections of the form $I + te_{i,j}$ where $i \neq j$ and $t \in \mathbb{F}_q$. Then the attack goes on smoothly as we saw earlier. However, when we look at generators of the form $I + te_{i,j} - te_{-j,-i}$, where $t \in \mathbb{F}_q$ and $i \neq j$; conjugating by them gets us a linear sum of the i^{th} and j^{th} column, not scalar multiple of one particular column. This stops the attack from going forward. However in the symplectic groups there are generators of the form $I + e_{i,-i}$ and $I + e_{-i,i}$ for $1 \leqslant i \leqslant l$. These generators make the attack possible for the symplectic groups. However there are no such generators for the orthogonal groups and so this attack turns out to be impossible for orthogonal groups.

5 The case for 2-generators and prime fields

One serious objection against a MOR cryptosystem is the size of the key [15, Section 7]. The reason is simple: we saw that in a MOR cryptosystem the automorphisms are presented as action on generators. Now bigger the number of generators, larger the key-size.

On the other hand, many of the finite simple groups can be generated by two elements. However, a set of generators is not enough. We must be able to compute the image of an arbitrary element. When the automorphism is presented as action on generators, we need an efficient solution to the word problem in order to do that. We have demonstrated [3, Section 5] that we have one set of generators, the elementary matrices, for which the word problem is easy.

The theme of this section is that for symplectic and even-orthogonal groups, there are two generators and for the odd-orthogonal group there are three generators. Over the **prime field of odd characteristic** one can easily compute the word corresponding to the elementary matrices over these generators.

So one can present the automorphisms ϕ and ϕ^m as action on these few generators and then compute the action of these automorphisms on the elementary matrices later. This substantially reduces the key-size. To do this we use the technique of *straight line programs*, which is popular in computational group theory. These are programs, but in practice are actually easy to use formulas. Say for example, we want to compute $x_{i,j}(t)$ for some $t \in \mathbb{F}_q$. We have loaded matrices $w^{i-1}x_{1,2}(\cdot)w^{(i-1)}$ in memory in such a way that this formula takes as input t and put it in the $(1,2)$ position of the matrix $x_{1,2}(\cdot)$ and do the matrix multiplication. This is one straight line program. Since these programs are loaded in memory, computation is much faster. This is somewhat similar to a time-memory trade-off. We have built a series of these straight line programs, where one straight line program can use other straight line programs and have written down the length of these programs. The length is nothing but the number of matrices in the formula.

Using the symplectic group in the MOR cryptosystem is straightforward. However, using orthogonal groups is little tricky because of the presence of λ in the output of the Gaussian elimination algorithm [3, Section 5]. It is well-known that the elementary matrices without w_i – the row interchange matrices, generate Ω the commutator subgroup of a orthogonal group. However in between the commutator and the whole group there is another important subgroup,

$W\Omega = \langle \Omega, w_i \rangle$ for some i. From the algorithm point of view, it is the subgroup of all the matrices for which the λ is a square. Now once the λ is a square, and we can efficiently compute the square root, we can write this matrix down as product of elementary matrices and it is easy to implement the MOR cryptosystem. It is well known that if $p \equiv 3 \pmod 4$, then it is easy to compute the square root. Only for this reason, in the latter part of this section and for orthogonal groups we concentrate on $p \equiv 3 \pmod 4$.

5.1 Symplectic group $Sp(2l, p)$

Let p be an odd prime. It is known [20] that the group $Sp(2l, p)$ is generated by two elements:

$$x = \quad x_{1,2}(1) \tag{1}$$

$$w = \begin{pmatrix} 0 & 1 \\ -I_{2l-1} & 0 \end{pmatrix} \tag{2}$$

We will refer these two elements as **Steinberg generators**. However in the context of the MOR cryptosystem we need to know how to go back and forth between these two generating sets – Steinberg generators and elementary matrices [3, Section 5]. To write w as a product of elementary matrices is easy, just put this generator through our Gaussian elimination algorithm. Here we demonstrate the other way round, that is, how to write elementary matrices as a product of x and w. In what follows, we denote the length of SLP's by $L(n, i)$, where $n = j - i$ and $1 \leqslant i < j \leqslant l$.

$$\begin{aligned}
n = 1, \quad x_{i,j}(t) &= w^{i-1} x_{1,2}(t) w^{-(i-1)}, \\
n = 2, \quad x_{i,j}(t) &= [x_{i,j-1}(t), x_{j-1,j}(1)], \\
n = 3, \quad x_{i,j}(t) &= [x_{i,j-1}(t), x_{j-1,j}(1)], \\
\vdots \qquad \vdots &\qquad \vdots \\
n = l-1, \quad x_{i,j}(t) &= [x_{i,j-1}(t), x_{j-1,j}(1)].
\end{aligned}$$

Here

$$L(n, i) = \begin{cases} 2i - 1 & \text{for } n = 1, \\ 2L(n-1) + 4(i+n) - 6 & \text{for } n = 2, 3, \ldots l-1. \end{cases}$$

Now $w^l = (-1)^{l-1}\begin{pmatrix} 0 & I_l \\ -I_l & 0 \end{pmatrix}$ and $x_{j,i}(t) = w^l x_{i,j}(-t)w^{-l}$, so length of this SLP is $L(n,i) + 2l$. Hence we get all $x_{i,j}(t)$ for $1 \leqslant i \neq j \leqslant l$. Number of SLP is l. Next observe that,

elements	indices	equation	length	
$x_{1,-l}(t)$		$wx_{l-1,l}(t)w^{-1}$	$2l-1$	
$x_{1,-i}(t)$	$2 \leqslant i \leqslant l-1$	$[x_{i,l}(t), x_{1,-l}(1)]$	$2(L(l-i,i)+2l-1)$	
$x_{i,-j}(t)$	$2 \leqslant i \leqslant l-1$ $(i+1 \leqslant j \leqslant l)$	$[x_{i,1}(t), x_{1,-j}(1)]$	$2(L(i-1,1)+4l-1)$ $2(L(i-1,1)+2L(l-j,j)$ $+6l-2)$	$j = l$ $j \neq l$
$x_{i,-i}(t)$	$i = 1,2,\ldots,l-1$	$[x_{i,i+1}(\frac{t}{2}), x_{i,-(i+1)}(1)]$	$2(2L(l-2,1)+10l-5)$ $2\,(L(1,i)+2L(i-1,1)+$ $4L(l-(i+1),i+1)$ $+12l-4)$	$i = l-1$ $i \neq l-1$
$x_{l,-l}(t)$		$[x_{l,l-1}(\frac{t}{2}), x_{l-1,-l}(1)]$	$2(2L(l-2,1)+12l-5)$	

So we generate all $x_{i,-j}(t)$ for $1 \leqslant i < j \leqslant l$ and $x_{i,-i}(t)$ for $1 \leqslant i \leqslant l$. Now $w^l x_{i,-j}(t)w^{-l} = x_{-i,j}(t)$ for $1 \leqslant i < j \leqslant l$ and $w^l x_{i,-i}(t)w^{-l} = x_{-i,i}(t)$ for $1 \leqslant i \leqslant l$, then we get $x_{-i,j}(t)$ and $x_{-i,i}(t)$. Total number of SLPs is $l + (3+1) + (2+1) = l + 7$. Hence we generate all the elementary matrices [3, Section 5] using only two generators x and w. It is shown in Ree [16] that elementary matrices generate the symplectic group $Sp(2l,p)$. Hence $Sp(2l,p)$ is generated by only two generators x and w.

5.2 Orthogonal group $O(2l,p)$

Let $p \equiv 3 \pmod 4$ be a prime. It is known [20] that the group $O(2l,p)$ is generated by two elements:

$$x = x_{1,2}(1), \tag{3}$$

$$w = \left(\begin{array}{ccc|ccc} 0 & \cdots & 0 & 0 & \cdots & 1 \\ -1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \cdots & -1 & 0 & 0 & \cdots & 0 \\ \hline 0 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & -1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & -1 & 0 \end{array}\right). \tag{4}$$

We will refer these two elements as **Steinberg generators**. As we discussed earlier, in context of the MOR cryptosystem we need to know how to go back and forth between two generating sets – Steinberg generators and elementary matrices [3, Section 5]. To write w as a product of elementary matrices is easy, just put this generator through our Gaussian elimination algorithm. Here we demonstrate the other way round, that is, how to write elementary matrices as a product of x and w. In what follows, we denote the length of SLP's by $L(n,i)$, where $n = j - i$ and $1 \leqslant i < j \leqslant l$.

$$\begin{aligned} n = 1, \;\; x_{i,j}(t) &= \; w^{i-1}x_{1,2}(t)w^{-(i-1)}, \\ n = 2, \;\; x_{i,j}(t) &= \; [x_{i,j-1}(t), x_{j-1,j}(1)], \\ n = 3, \;\; x_{i,j}(t) &= \; [x_{i,j-1}(t), x_{j-1,j}(1)], \\ \vdots \qquad \vdots \qquad &\vdots \\ n = l-1, \;\; x_{i,j}(t) &= \; [x_{i,j-1}(t), x_{j-1,j}(1)]. \end{aligned}$$

Here

$$L(n,i) = \begin{cases} 2i-1 & \text{for } n = 1, \\ 2L(n-1) + 4(i+n) - 6 & \text{for } n = 2, 3, \ldots l-1. \end{cases}$$

Now $w^l = (-1)^{l-1}\begin{pmatrix} 0 & -I_l \\ -I_l & 0 \end{pmatrix}$ and $x_{j,i}(t) = w^l x_{i,j}(-t)w^{-l}$, so length of this SLP is $L(n,i) + 2l$. Hence we get all $x_{i,j}(t)$ for $1 \leqslant i \neq j \leqslant l$. The number of SLPs is l. Next observe the following:

elements	indices	equation	length	
$x_{1,-l}(t)$		$wx_{l-1,l}(t)w^{-1}$	$2l-1$	
$x_{1,-i}(t)$	$2 \leqslant i \leqslant l-1$	$[x_{i,l}(t), x_{1,-l}(1)]$	$2(L(l-i,i) + 2l - 1)$	
$x_{i,-j}(t)$	$2 \leqslant i \leqslant l-1$	$[x_{i,1}(t), x_{1,-j}(1)]$	$2(L(i-1,1) + 2L(l-j,j)$ $+6l-2)$	$j \neq l$
	$(i+1 \leqslant j \leqslant l)$		$2(L(i-1,1) + 4l - 1)$	$j = l$

So we generate all $x_{i,-j}(t)$ for $i < j$. Now $w^l x_{i,-j}(t)w^{-l} = x_{-i,j}(t)$, and we get $x_{-i,j}(t)$ and the total number of SLPs is $l + 4$. It is shown by Ree [16] that elementary matrices $x_{i,j}(t)$ generate $\Omega(2l,p)$, the commutator subgroup of $O(2l,p)$. Hence we generate $\Omega(2l,p)$, using only two elements x and w. Since we generate $x_{i,j}(t)$ and $w_{i,j}$ as a product of $x_{i,j}(t)$ and $w = w_{1,2}(1)w_{2,3}(1)\ldots w_{l-1,l}(1)w_l$, so we are able to generate w_l. Here $w_{i,j}(t) = x_{i,j}(t)x_{j,i}(-t^{-1})x_{i,j}(t)$ for $i \neq j$ and $w_l = I - e_{l,l} - e_{-l,-l} + e_{l,-l} + e_{-l,l}$. Now we know $w_{l-1} = w_l w_{l,l-1}(1)w_{l-1,-l}(1)$, so we generate w_{l-1}. Hence by

induction, we generate $w_i = w_{i+1}w_{i+1,i}(1)w_{i,-(i+1)}(1)$ for $i = l-1, \ldots, 1$. Here $w_{i,-j}(t) = x_{i,-j}(t)(1)x_{-i,j}(t^{-1})x_{i,-j}(t)$, for $i < j$. Hence we generate all the elementary matrices defined earlier [3, Section 5] using only two generators x and w. So we generate a new subgroup $W\Omega(2l,p)$ of $O(2l,p)$, which is a normal subgroup of $O(2l,p)$. In our algorithm output matrix is $d(\lambda) = \text{diag}\,(1,1,\cdots,\lambda,1,1,\cdots,\lambda^{-1})$. If $\lambda \in F_p^{\times 2}$, say $\lambda \equiv t^2 \pmod{p}$, then $t \equiv \lambda^{\frac{p+1}{4}} \pmod{p}$, since $p \equiv 3 \pmod{4}$. Then

$$\begin{aligned} d(\lambda) &= \text{diag}\,(1,\cdots,t^2,1,\cdots,t^{-2}) \\ &= w_{l-1,l}(1)\text{diag}\,(1,\cdots,t^2,1,1,\cdots,t^{-2},1)w_{l-1,l}(-1) \\ &= w_{l-1,l}(1)w_{l-1,l}(t)w_{l-1,l}(-1)w_{l-1,-l}(t)w_{l-1,-l}(-1)w_{l-1,l}(-1). \end{aligned}$$

Hence we generate $W\Omega(2l,p)$ using only two generators x and w.

5.3 Orthogonal group $O(2l+1,p)$

Let $p \equiv 3 \pmod{4}$ be a prime. It is known [20] that the group $O(2l+1,p)$ is generated by these elements:

$$x = x_{0,1}(1), \tag{5}$$

$$w = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -I_{2l-1} & 0 \end{pmatrix}, \tag{6}$$

$$w_l = I - e_{l,l} - e_{-l,-l} + e_{l,-l} + e_{-l,l}. \tag{7}$$

We will refer these two elements as **Steinberg generators**. However in context of the MOR cryptosystem we need to know how to go back and forth between these two generating sets – Steinberg generators and elementary matrices defined earlier [3, Section 5]. To write w as a product of elementary matrices is easy, just put this generator through our Gaussian elimination algorithm. Here we demonstrate the other way round, that is, how to write elementary matrices as a product of w and x. First we compute, $x_{0,i}(t) = w^{i-1}x_{0,1}(1)w^{-(i-1)}$ which is of length $2i-1$ for $1 \leqslant i \leqslant l$. Now

$$w^l = (-1)^l \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & I_l \\ 0 & I_l & 0 \end{pmatrix}$$

and $x_{i,0}(t) = w^l x_{0,i}(-t)w^{-l}$ for $1 \leqslant i \leqslant l$ and length of this SLP is $2l + 2i - 1$. So we get $x_{i,0}(t)$ and $x_{0,i}(t)$ for $i = 1, 2, \ldots, l$. Again we have $x_{1,2}(t) = [x_{1,0}(\frac{t}{2}), x_{0,2}(1)]$ and length of this SLP is $4l + 8$. In what follows, we denote the length of SLP's by $L(n, i)$, where $n = j - i$ and $1 \leqslant i < j \leqslant l$.

$$\begin{aligned} n = 1, \;\; x_{i,j}(t) &= \; w^{i-1} x_{1,2}(t) w^{-(i-1)}, \\ n = 2, \;\; x_{i,j}(t) &= \; [x_{i,j-1}(t), x_{j-1,j}(1)], \\ n = 3, \;\; x_{i,j}(t) &= \; [x_{i,j-1}(t), x_{j-1,j}(1)], \\ \vdots \quad\quad \vdots \quad\quad &\vdots \\ n = l - 1, \;\; x_{i,j}(t) &= \; [x_{i,j-1}(t), x_{j-1,j}(1)]. \end{aligned}$$

Here

$$L(n, i) = \begin{cases} 2i + 4l + 6 & \text{for } n = 1, \\ 2L(n - 1, i) + 4(i + n + 2l + 2) & \text{for } n = 2, 3, \ldots l - 1. \end{cases}$$

As $x_{j,i}(t) = w^l x_{i,j}(-t)w^{-l}$, so the length of this SLP is $L(n, i) + 2l$. Hence we generate all $x_{i,j}(t)$ for $1 \leqslant i \neq j \leqslant l$ and the number of SLPs is $3 + (l - 1) + 1 = l + 3$. Next observe that,

elements	indices	equation (SLP)	length	
$x_{1,-l}(t)$		$wx_{l-1,l}(t)w^{-1}$	$6l + 6$	
$x_{1,-i}(t)$	$2 \leqslant i \leqslant l - 1$	$[x_{i,l}(t), x_{1,-l}(1)]$	$24l + 20$ $2L(l - i, i) + 12(l + 1)$	$i = l - 1$ $i \neq l - 1$
$x_{i,-j}(t)$	$2 \leqslant i \leqslant l - 1$ $(i + 1 \leqslant j \leqslant l)$	$[x_{i,1}(t), x_{1,-j}(1)]$	$2L(i - 1, 1) + 4L(l - j - n, j - n)$ $+4(7l + 6)$ $2L(i - 1, 1) + 4(7l + 5)$ $2L(i - 1, 1) + 10l + 6$	$j < l - 1$ $j = l - 1$ $j = l$

So we generate all $x_{i,-j}(t)$ for $i < j$. Now $w^l x_{i,-j}(t)w^{-l} = x_{-i,j}(t)$, and we have $x_{-i,j}(t)$. The total number of SLPs is $l + 7$. It is shown in Ree [16] that elementary matrices $x_{i,j}(t)$'s generate $\Omega(2l + 1, p)$, the commutator subgroup of $O(2l + 1, p)$ which is of index 4. So we generate $\Omega(2l + 1, p)$, using only two generators x and h. Now we know $w_{l-1} = w_l w_{l,l-1}(1) w_{l-1,-l}(1)$, so we generate w_{l-1}. Hence inductively we can generate $w_i = w_{i+1} w_{i+1,i}(1) w_{i,-(i+1)}(1)$ for $i = l - 1, \ldots, 1$. Here $w_{i,j}(t) = x_{i,j}(t)x_{j,i}(-t^{-1})x_{i,j}(t)$ for $i \neq j$ and $w_{i,-j}(t) = x_{i,-j}(t)x_{-i,j}(t^{-1})x_{i,-j}(t)$ for $i < j$. Hence we generate all the elementary matrices defined earlier [3, Section 5] using only two generators x and w and an extra element w_l. Hence we generate a new subgroup $W\Omega(2l + 1, p)$ of the orthogonal group $O(2l + 1, p)$, containing Ω, which is indeed a nor-

mal subgroup of $O(2l+1,p)$. In our algorithm the output matrix is $d(\lambda) = \mathrm{diag}\,(1,1,\cdots,\lambda,1,\cdots,\lambda^{-1})$. If $\lambda \in F_p^{\times 2}$, say $\lambda \equiv t^2 \pmod{p}$, here $t \equiv \lambda^{\frac{p+1}{4}} \pmod{p}$, since $p \equiv 3 \pmod{4}$. Then

$$\begin{aligned} d(\lambda) &= \mathrm{diag}\,(1,1,\cdots,t^2,1,\cdots,t^{-2}) \\ &= w_{l-1,l}(1)\mathrm{diag}\,(1,1,\cdots,t^2,1,1,\cdots,t^{-2},1)w_{l-1,l}(-1) \\ &= w_{l-1,l}(1)w_{l-1,l}(t)w_{l-1,l}(-1)w_{l-1,-l}(t)w_{l-1,-l}(-1)w_{l-1,l}(-1). \end{aligned}$$

Hence we generate $W\Omega(2l+1,p)$ using x, w and w_l.

Remark 2. *Let $d(\zeta) = \mathrm{diag}\,(1,1,\cdots,\zeta,1,\cdots,\zeta^{-1})$, where ζ is non-square in $F_p^{\times}$. Then the group $\langle W\Omega, d(\zeta)\rangle$ is the orthogonal group.*

6 Conclusion

This section is similar to [13, Section 8]. An useful public-key cryptosystem is a delicate dance between speed and the security. So one must talk about speed along with security.

The implementation of the MOR cryptosystem that we have in mind uses the row-column operations. Let $\langle g_1, g_2, \ldots, g_s\rangle$ be a set of generators for the orthogonal or symplectic group as described before. As is the custom with a MOR cryptosystem, the automorphisms ϕ and ϕ^m are presented as action on generators, i.e., we have $\phi(g_i)$ and $\phi^m(g_i)$ as matrices for $i = 1, 2, \ldots, s$.

To encrypt a message in this MOR cryptosystem, we compute ϕ^r. We do that by *square-and-multiply* algorithm. For this implementation, squaring and multiplying is almost the same. So we will refer to both squaring and multiplication as multiplication. Note that multiplication is composing of automorphisms.

The implementation that we describe in this paper, can work in parallel. Each instance computes $\phi^r(g_i)$ for $i = 1, 2, \ldots, s$. First thing that we do is write the matrix of $\phi(g_i)$ as a word in generators. So essentially the map ϕ becomes a map $g_i \mapsto w_i$ where w_i is a word in generators of some fixed length. Then multiplication becomes essentially a replacement, replace all instances of g_i by w_i. This can be done very fast. However, the length of the replaced word can become very large. The obvious question is, how soon are we going to write this word as a matrix. This is a difficult question to answer at this stage and depends on available computational resources.

Once we decide how often we change back to matrices, how are we going

to change back to matrices? There can be a fairly easy *time-memory* trade-offs. Write all words up to a fixed length and the corresponding matrix as a pre-computed table and use this table to compute the matrices. Once we have matrices, we can multiply them together to generate the final output. There are also many obvious relations among the generators of these groups. One can just store and use them. The best strategy for an efficient implementation is yet to be determined. It is clear now that there are many interesting and novel choices.

The benefits of this MOR cryptosystem are:

This can be implemented in parallel easily.

This implementation doesn't depend on the size of the characteristic of the field. This is an important property in light of Joux's recent improvement of the index-calculus attacks [1].

For parameters and complexity analysis of this cryptosystem, we refer to [13, Section 8]. Assume that we take a prime of size 2^{160}, and we are using two generators presentation of ϕ for the even-orthogonal group. Then the security is the discrete logarithm problem in $\mathbb{F}_{p^{d^2}}$. Now if we take $d = 4$, then the security better than $\mathbb{F}_{2^{2560}}$. Our key size is about 8000 bits. Comparing with Monico [15, Section 7], where he says an ElGamal will have about 6080 bits, our system is quite comparable. Moreover, the MOR cryptosystem is better suited to handle large primes and can be easily parallelized.

Acknowledgments

The authors gratefully acknowledges the support of SERB.

References

[1] Barbulescu R., Gaudry P., Joux A., and Thome E., "A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic", In Eurocrypt 2014, 2014, 1 –16

[2] Bhunia S., Mahalanobis A., Shinde P., and Singh A., *Modern Cryptography – Theory, Technology, Adaptation and Integration. The MOR Cryptosystem in Classical Groups with a Gaussian Elimination Algorithm for Symplectic and Orthogonal Groups*, IntechOpen, 2019

[3] Bhunia S., Mahalanobis A., and Singh A., "Gaussian elimination in symplectic and split orthogonal groups", *Technical report, IISER Pune*, 2015, http://arxiv.org/abs/1504.03794.

[4] Carter R., *Pure and Applied Mathematics*, Simple groups of Lie type, **28**, John Wiley & Sons, 1972.

[5] Climent J.-J., Navarro P.R., and Tortosa L., "An extension of the noncommutative bergman's ring with a large number of noninvertible elements", Applicable Algebra in Engineering, Communication and Computing, **25**:5 (2014), 347–361.

[6] Dieudonne J., *On the automorphisms of the classical groups*, ed. With a supplement by Loo-Keng Hua, Memoirs of the American Mathematical Society, 1951.

[7] Monico C., Maze G., and Rosenthal J., "A public key cryptosystem based on action by semigroups", In Proceedings of IEEE International symposium on Information Theory, 2002.

[8] Monico C., Maze G., and Rosenthal J., "Public key cryptography based on semigroup actions", Adv. in Math. of Communications, **4**:1 (2007), 489–506.

[9] Glukhov M.M., "An analysis of some key distribution public systems based on non-abelian groups.", *Mat. Vopr. Kriptogr*, **4**, 2010, 5–22.

[10] Joux A., "A new index calculus algorithm with complexity $L(1/4 + o(1))$ in small characteristic", In SAC 2013, 2013, 355–379.

[11] Kahrobaei D., Koupparis C., and Shpilrain V., "Public key exchange using matrices over group rings", *Groups-Complexity-Cryptology*, **5**, 2013, 97–115.

[12] Knus M.-A., Merkurjev A., Rost M., and Tignol J.-P., *The book of involutions (English summary)*, **44**, American Mathematical Society Colloquium Publications, 1998.

[13] Mahalanobis A., "A simple generalization of the ElGamal cryptosystem to non-abelian groups II", *Communications in Algebra*, **40**:9 (2012), 3583–3596.

[14] Mahalanobis A., "The MOR cryptosystem and finite p-groups", *In Contemporary Mathematics*, **633**, 2015, 81–95.

[15] Monico C., "Cryptanalysis of matrix-based MOR system", *Communications in Algebra*, **44**, 2016, 218–227.

[16] Ree R., "On some simple groups defined by C. Chevalley", *Transactions of the American Mathematical Society*, **84**, 1957, 392–400.

[17] Seligman G.B., *Modular Lie Algebras*, Springer-Verlag, 1967.

[18] Shpilrain V. and Zapata G., "Combinatorial group theory and public key cryptography", *Applicable Algebra in Engineering, Communication and Computing*, **17**, 2006, 291–302.

[19] Steinberg R., "Automorphisms of finite linear groups", *Canadian Journal of Mathematics*, **12**, 1960, 606–615.

[20] Steinberg R., "Generators of simple groups", *Canadian journal of Mathematics*, **14**, 1962, 277–283.

Random Number Generators Based on Permutations Can Pass the Collision Test

Alexey Urivskiy

JSC InfoTeCS, Russia
alexey.urivskiy@mail.ru, urivskiy@infotecs.ru

Abstract

In this paper, we investigate pseudorandom number generators based on random permutations which in cryptographic applications are modeled by block ciphers with random keys. We give a simple method to calculate upper and lower bounds on the probability to observe a collision in an output sequence of finite length given the respective bounds on conditional probability of the next symbol to appear given a prefix. We found that the difference between the upper and lower bounds on collision probability can be made extremely small for any practical parameters of interest. Moreover the collision probability for a true random number generator (RNG) always lies within these bounds. This implies that the investigated generators will pass the collision test, i.e. they are indistinguishable by this test from a true RNG.

Keywords: pseudorandom number generator, permutation, unpredictability, collision, block cipher.

1 Introduction

Random numbers are of crucial importance for cryptographic applications. True randomness is obtained from some nondeterministic physical processes. However, it could be a problem to find a true source of randomness, which is fast enough, without memory, and with uniform output distribution. So typically deterministic pseudorandom number generators (PRNG) are used in applications to generate randomly looking numbers. It is assumed that they produce sequences which are indistinguishable from truly random ones by any polynomial statistical test. In practice they must at least pass batteries of particular statistical tests.

There are different techniques for designing PRNGs [1]. One of the most widely spread is to use a block cipher in counter mode of operation — the well-known `CTR_DRBG` generator [1]. In fact, a block cipher with random key just

models a random permutation. In this paper, a PRNG based on two permutations is considered. To generate an output symbol the outputs of two permutations are XORed together.

We will show that using two permutations allows the PRNG to pass the collision test for output sequences of practical lengths. If only one permutation is used and the output symbol is a half of the permutation output we get similar results.

2 PRNG Description

Let V_n be the set of all binary strings (vectors) of length n with the bitwise eXclusive OR addition defined on it. To every string $z_{n-1}||z_{n-2}||\ldots||z_0$ from V_n we put into one-to-one correspondence the integer $2^{n-1}z_{n-1} + 2^{n-2}z_{n-2} + \ldots + 2z_1 + z_0$, which is an element of the residue ring $\mathbb{Z}_N = \{0, \ldots, N-1\}$, where $N = 2^n$.

A permutation σ on V_n is a one-to-one mapping of $\mathbb{Z}_N$ to itself.

Consider the following PRNG, call it **G2I**, which is based on two permutations. For this PRNG the counter *count* is initialized by a randomly and uniformly chosen $IV \in V_n$, and as an input an integer s and 2 independent and randomly chosen permutations σ_1 and σ_2 are given. The output will be n-bit symbols $x_0, x_1, \ldots, x_{s-1}$.

G2I: for i from 0 to $s-1$ do:

$$count := (IV + i) \mod 2^n;$$
$$x_i := \sigma_1(count) \oplus \sigma_2(count).$$

The output sequences of **G2I** are periodic with period of $N = 2^n$. In the following, we consider output sequences of length at most N. So we neither consider practical details of initializing (seeding) the generator, nor the details of reseeding it.

This type of a generator was partially investigated in [2]. It was shown that for **G2I** the value of IV is not important when security is concerned, found the number of different output sequences and the conditional probability of the next output symbol to appear given a prefix. In this research, we go further and investigate collisions in output sequences.

3 Collision Test

In [2] it was established the following lemma.

Lemma 1. *In case* $s < N/2$ *for* ***G2I*** *the conditional probability* $P(x_s|x_{s-1},\ldots,x_0)$ *of the next output symbol* x_s *to appear given a prefix* $(x_{s-1},x_{s-2},\ldots,x_0)$ *is bounded as*

$$P_1(s+1)=\frac{N-2s}{(N-s)^2}\leqslant P(x_s|x_{s-1},\ldots,x_0)\leqslant\frac{N-s}{(N-s)^2}=P_2(s+1). \quad (1)$$

From Lemma 1 it follows that for $s/N \ll 1$ this conditional probability is close to $P_0 = 1/N$, which is the probability for the next symbol to appear for a true RNG.

Moreover, for a generalized variant of the generator based on $R > 2$ permutations it was shown [3] that the difference between the conditional probabilities for the investigated generator and for a true RNG decreases exponentially fast with R for a prefix of fixed length. So the construction of a PRNG, where outputs of multiple permutations are XORed together, is effective concerning unpredictability. However, the smaller R we could take, the more computationally efficient will be the generator. So the most interesting case is $R = 2$.

Usually, in practice, PRNGs are assessed through a battery of statistical tests. They try to distinguish the RNG under investigation from a true RNG. And each particular test tries to highlight a certain flavor of nonrandomness. One of the most known statistical tests for RNGs is the so called *collision test*. The collision test counts the number of occurrences of identical symbols in the output sequence. An RNG fails the test if the number of collisions falls outside a predefined interval.

However, for quite a good RNG with output symbols from a large alphabet it would require a huge amount of data to store or handle before at least once a collision could be found. So more typical approach is to estimate the critical length of output sequences when the investigated RNG might become distinguishable from a true RNG. Having collision test in mind, a common distinguishing criterion is the difference between the *collision probabilities*, i.e. the probability to find at least two identical symbols in a sequence of finite length, for the investigated generator and a true one. The critical length is when this difference become large enough, say compared to 1. This could be qualitatively explained as follows. Suppose we observe output sequences of length exactly s

of the generator. And the collision probability is $P_C(s)$. After observing the sequence we re-initialize the generator randomly. We can consider different starts to be independent. Then every start is a Bernoulli trial in which the probability of success is $P_C(s)$. We expect that after m trials we observe $mP_C(s)$ sequences with collisions. With some degree of certainty we could distinguish this generator from a true RNG if $mP_C(s) - mP_I(s) \sim 1$, where $P_I(s)$ is the collision probability for a true RNG. If $P_C(s) - P_I(s) < \delta(s)$ for some security parameter $\delta(s) = 1/m$, then the adversary cannot distinguish the generators after processing $s/\delta(s)$ symbols or $sn/\delta(s)$ bits. So if we allow only one start, here $m = 1$, the critical length s^* is determined as a length when $P_C(s^*) - P_I(s^*) \sim 1$.

For any particular instance of the RNG the length of output sequences is deliberately limited far before the critical length is reached. The larger this critical length, the better the RNG. The exact value of the admissible probabilities difference highly depends on the application or system's security requirements. More important, however, the functional dependence of the probability on the output length.

We are going to estimate the collision probability for **G2I**.

In [4], using provable security approach, it was proven that the sum of R random permutations XORed together gives a pseudorandom function, essentially a PRNG, when $s \ll O(N^{\frac{R}{R+1}})$ queries to the oracle are allowed. In our paper we describe a simple computational technique to estimate a bound for collision probability having a bound for conditional probability for the next symbol to appear and whenever the latter is expressed as $\frac{1+f(s)}{N}$, where $f(s) \ll 1$ is a polynomial with zero constant term. Compared to [4], where only a lower bound for collision probability is given, we were able to obtain both lower and upper bounds.

In [5], using probabilistic and combinatorial arguments, it was shown that **G2I** is secure when $s \ll O(N^{\frac{9}{10}})$, and even $s \ll O(N)$. Our results are obtained much easier, they are much simpler to follow, and can easily be used to have a particular security treatment. In a sense they are close to [5] concerning security, see further Section 6.

Evidently, our results are valid under collision test setting only, while the above mentioned ones are valid against arbitrary distinguisher. Nevertheless we think that they might be generalized to provable security setting. They are essentially based on double-sided estimations of conditional probability for the next symbol to appear which does not depend on particular test. However, a

discussion on this topic would go far beyond the scope of this paper.

4 Collision probability

Bounds (1) give us the conditional probability $P(x_s|x_{s-1},\dots,x_0)$ for the $s+1$-th symbol x_s to appear provided we observed a prefix $(x_{s-1},\dots,x_0)$ of length s. Suppose now that all s symbols of the prefix are different. Let us estimate the probability $P_d(s+1)$ that the $s+1$-th symbol is different from all the previous ones:

$$P_d(s+1) = P\big(x_s \notin \{x_{s-1},\dots,x_0\}\big|x_i \neq x_j; i \neq j; i,j = \overline{0,s-1}\big).$$

Proposition 1. *The probability $P_d(s+1)$ to have all different elements in a prefix of length $s+1$ for **G2I** is bounded as*

$$1 - \frac{s(N-s)}{(N-s)^2} \leqslant P_d(s+1) \leqslant 1 - \frac{s(N-2s)}{(N-s)^2}. \tag{2}$$

Proof. The possible outcomes for x_s are that it is either equal to a particular symbol in the prefix or differs from all of them. All these outcomes are mutually exclusive. Evidently

$$P_d(s+1) + \sum_{i=0}^{s-1} P(x_s = x_i|x_{s-1},\dots,x_0) = 1.$$

Since estimation (1) is valid for any prefix, including the one with $x_i \neq x_j; i \neq j; i,j = \overline{0,s-1}$, and any next expected symbol x_s, we obtain that for any $i \in 0,\dots,s-1$

$$\frac{N-2s}{(N-s)^2} \leqslant P(x_s = x_i|x_{s-1},x_{s-2},\dots,x_0) \leqslant \frac{N-s}{(N-s)^2}.$$

Hence $1 - sP_2(s+1) \leqslant P_d(s+1) \leqslant 1 - sP_1(s+1)$ which leads immediately to (2). □

Using the general formula for the probability of joint events through conditional probabilities, it is easy to see that the probability $P_D(s+1)$ that all $s+1$

output symbols are different is

$$P_D(s+1) = P(x_i \neq x_j; i \neq j; i, j = \overline{0, s}) = \prod_{i=0}^{s} P_d(i+1). \tag{3}$$

The probability $P_C(s+1)$ to encounter a collision after observing $s+1$ output symbols of the generator is

$$P_C(s+1) = 1 - P_D(s+1). \tag{4}$$

Using (2), (3) and (4) we get

$$1 - \prod_{i=0}^{s} \left(1 - \frac{i(N-2i)}{(N-i)^2}\right) \leqslant P_C(s+1) \leqslant 1 - \prod_{i=0}^{s} \left(1 - \frac{i(N-i)}{(N-i)^2}\right). \tag{5}$$

Using the Taylor series expansion of the exponential function $e^z = 1 + z + \frac{z^2}{2} + o(z^2)$, we take the first-order approximation: $e^z \approx 1 + z$ for $z \ll 1$. Consider now $i \leqslant s \ll N/2$. In this case both $\dfrac{i(N-i)}{(N-i)^2}$ and $\dfrac{i(N-2i)}{(N-i)^2}$ are much smaller than 1. Therefore,

$$1 - \frac{i(N-i)}{(N-i)^2} \approx e^{-\dfrac{i(N-i)}{(N-i)^2}} \quad \text{and} \quad 1 - \frac{i(N-2i)}{(N-i)^2} \approx e^{-\dfrac{i(N-2i)}{(N-i)^2}}.$$

Now from (5) we obtain

$$1 - \exp\left(-\sum_{i=0}^{s} \frac{i(N-2i)}{(N-i)^2}\right) \leqslant P_C(s+1) \leqslant 1 - \exp\left(-\sum_{i=0}^{s} \frac{i(N-i)}{(N-i)^2}\right).$$

Let us compute the sums. Using the Taylor series expansion for the function $(1+z)^\alpha = 1 + \alpha z + \frac{\alpha(\alpha-1)}{2} z^2 + o(z^2)$ we get

$$\sum_{i=0}^{s} \frac{i(N-i)}{(N-i)^2} = \sum_{i=0}^{s} \frac{i}{N} \left(1 + \frac{i}{N} + \left(\frac{i}{N}\right)^2 + o\left(\left(\tfrac{i}{N}\right)^2\right)\right),$$

$$\sum_{i=0}^{s} \frac{i(N-2i)}{(N-i)^2} = \sum_{i=0}^{s} \frac{i}{N} \left(1 - \left(\frac{i}{N}\right)^2 + o\left(\left(\tfrac{i}{N}\right)^2\right)\right).$$

We take the known formulas for the sums

$$\sum_{i=0}^{s} i = \frac{s(s+1)}{2}, \quad \sum_{i=0}^{s} i^2 = \frac{s(s+1)(2s+1)}{6}, \quad \sum_{i=0}^{s} i^3 = \frac{s^2(s+1)^2}{4},$$

to obtain

$$\sum_{i=0}^{s} \frac{i(N-i)}{(N-i)^2} = \frac{s(s+1)}{2N} + \frac{s(s+1)(2s+1)}{6N^2} + \frac{s^2(s+1)^2}{4N^3} + o(s^4/N^3),$$

$$\sum_{i=0}^{s} \frac{i(N-2i)}{(N-i)^2} = \frac{s(s+1)}{2N} - \frac{s^2(s+1)^2}{4N^3} + o(s^4/N^3)$$

Assuming that s is large we come to the following lemma.

Lemma 2. *For **G2I** the probability $P_C(s+1)$ to find a collision in an output of length $s+1$ is bounded as*

$$1 - \exp\left(-\frac{s(s+1)}{2N} + \frac{s^4}{4N^3}\right) \leqslant$$

$$\leqslant P_C(s+1) \leqslant$$

$$\leqslant 1 - \exp\left(-\frac{s(s+1)}{2N} - \frac{s^3}{3N^2} - \frac{s^4}{4N^3}\right), \quad (6)$$

provided $s \ll N/2$.

Recall that for a true RNG the collision probability after observing $s+1$ symbols is estimated as

$$P_I(s+1) \simeq 1 - \exp\left(-\frac{s(s+1)}{2N}\right).$$

We see that $P_I(s+1)$ lies within bounds (6).

It is clear, that the described technique for calculating bounds on the collision probability is easily applicable when the conditional probability $P(x_s|\ldots)$ is expressed as $\frac{1+f(s)}{N}$, where $f(s) \ll 1$ is a polynomial with zero constant term. For instance, the collision probability bounds for a generator based on R permutations [3] can be straightforwardly calculated.

5 A PRNG on a Single Truncated Permutation

As we mentioned earlier, the collision probability for `CTR_DRBG` is strictly 0. Can this PRNG can be improved concerning collisions? Fortunately, the answer is yes.

Let us consider permutations Σ on V_{2n}. And we build a PRNG **G1LI** on a single permutation Σ by selecting some t bits out of $2n$ in every output, and discarding the other $2n-t$ bits by the function $T_t()$:

G1LI: for i from 0 to $s-1$ do:

$$count := (IV + i) \mod 2^{2n};$$
$$x_i := T_t(\Sigma(count)).$$

The output sequences of **G1LI** are periodic with period of $N^2 = 2^{2n}$.

Lemma 3. *The conditional probability $P(x_s|x_{s-1}, \ldots, x_0)$ of the next output symbol x_s to appear given a prefix $(x_{s-1}, x_{s-2}, \ldots, x_0)$ for **G1LI** is bounded as*

$$\frac{N^2/2^t - s}{N^2 - s} \leqslant P(x_s|x_{s-1}, x_{s-2}, \ldots, x_0) \leqslant \frac{N^2/2^t}{N^2 - s}. \tag{7}$$

Proof. It is easy to see that the operation of **G1LI** corresponds to a random sampling without replacement from a multiset of cardinality 2^{2n} which consists of all elements of V_t each repeated 2^{2n-t} times. Hence it is evident that we may select any t bits by T_t.

Clearly the *random* sampling gives us bounds (7) on the conditional probability. □

Consider the case $t = n$. In this case we discard half of the permutation output. As a result we will have n-bit symbols in the output. We get the corollary of Lemma 3

Corollary 1. *If $t = n$, for **G1LI** the conditional probability is bounded as*

$$\frac{N - s}{N^2 - s} \leqslant P(x_s|x_{s-1}, x_{s-2}, \ldots, x_0) \leqslant \frac{N}{N^2 - s}. \tag{8}$$

It is interestingly to note that **G1LI** with $t = n$ is exactly the same generator as the one on a single random permutation, where the output symbol is computed by XORing two n-bit halves of Σ.

Both bounds in Corollary 1 are tight. The lower bound is attained when the expected symbol and all the symbols in the prefix are the same, while the upper bound is attained when the expected symbol differs from any symbol in the prefix. In Lemma 1 only the upper bound in definitely tight, and this happens when the expected symbol and all the symbols in the prefix are the same.

Evidently, for **G2I** the lower bound in (1) turns to 0 when $s = N/2$, while for **G1LI** the lower bound in (8) turns to 0 when $s = N$. Consequently one may assume that **G2I** is worse than **G1LI**. It seems, however, that the lower bound (1) is not tight. In [2] it was proven that for **G2I** *any* first $N-1$ elements in the output are possible. In other words, for any prefix of length N there is at least one pair of permutations that give that prefix being XORed together. This means that $P(x_s|x_{s-1},\ldots,x_0) > 0$ for any $s \leqslant N-2$. Furthermore, this in particular means that recursive computation $x_s = \mathrm{F}(x_{s-1},\ldots,x_{s-1-p})$ for **G2I** is only possible for $p = N-1$, and the function F is just XORing of all $N-1$ output symbols. This does not contradict the lower bound in (1). The way the lower bound is obtained just shows that there are pairs of permutations which being XORed together give prefixes of length $N/2$ such that certain elements of V_n cannot be observed after those prefixes.

By applying the technique from section 4 to bounds (8) it is quite straightforward to obtain the bounds on collision probability for **G1LI**.

Lemma 4. *In case $t = n$ for **G1LI** the probability $P_C(s+1)$ to find a collision in an output of length $s+1$ is bounded as*

$$1 - \exp\left(-\frac{s^2}{2N} + \frac{s^3}{2N^2}\right) \leqslant P_C(s+1) \leqslant 1 - \exp\left(-\frac{s^2}{2N} - \frac{s^3}{3N^3}\right). \qquad (9)$$

From Lemma 4 it is clear that **G1LI** has got a similar to **G2I** behavior concerning collisions. This is defined by the term $\frac{s^3}{2N^2}$ in the exponents. If we assume that computing Σ of $2n$ bits is as twice as more expensive as σ of n bits, we will see that both **G1LI** and **G2I** have the same performance. However, while increasing length of internal values twice, computing the output will typically require 4 times more operations. The last assumption could be justified if Σ is implemented as a $2n$-bit cipher, and σ — as an n-bit cipher. In this case **G2I** looks a bit more preferable.

6 Security Discussion

Now we estimate when the difference between the upper and lower bounds in Lemma 2 is negligible. Evidently it is true for the range $s^2 < 2N$. So we are interested what happens for $s^2 > 2N$. In this case the difference of the bounds is estimated as $\exp\left(-\frac{s^2}{2N}\right)\left(\exp\left(\frac{s^4}{4N^3}\right) - \exp\left(-\frac{s^3}{3N^2} - \frac{s^4}{4N^3}\right)\right)$. If

$$s^4 \leqslant 4N^3,$$

then the value in brackets does not exceed e, so the difference is no greater than $\exp\left(-\frac{s^2}{2N}\right)$.

The difference $\delta(s) = |P_C(s+1) - P_I(s+1)|$ is no greater than the difference between the upper and the lower bounds. In particular, one can estimate that for $s = \sqrt[3]{N}$ we get $\delta \leqslant O(1/N)$, for $s = \sqrt{N}$: $\delta \leqslant O\left(1/\sqrt{N}\right)$, for $s = \sqrt[3]{N^2}$: $\delta \leqslant O\left(e^{-\frac{\sqrt[3]{N}}{2}}\right)$, and for $s = \sqrt[4]{N^3}$: $\delta \leqslant O\left(e^{-\frac{\sqrt{N}}{2}}\right)$.

As we discussed, if $\delta(s)$ came close to 1, then a distinguisher could be built. Therefore, it is hardly possible to construct a distinguisher which for any fixed ratio s/N, provided $s \ll N$ and $s < \sqrt[4]{4N^3}$, could tell the difference between **G2I** and a true random number generator by looking for collisions. In other words, the **G2I**, an idealized version of the PRNG on two block ciphers, will surely pass the collision test.

This result contrasts greatly with `CTR_DRBG` generator, for which the collision probability is exactly 0 whatever s is. Indeed $\delta(s_1) = P_I(s_1)$ for `CTR_DRBG` and $\delta(s_2) \leqslant \exp\left(-\frac{s_2^2}{2N}\right)\left(\exp\left(\frac{s_2^4}{4N^3}\right) - \exp\left(-\frac{s_2^3}{3N^2} - \frac{s_2^4}{4N^3}\right)\right)$ for **G2I**. If we fix security conditions by $\delta(s_1) = \delta(s_2) \ll 0$, then s_1 and s_2 will be connected by the following equation

$$\frac{s_1^2}{2N} \simeq \left(\frac{s_2^4}{2N^3} + \frac{s_2^3}{3N^2}\right)\exp\left(-\frac{s_2^2}{2N}\right)$$

provided $\frac{s_1^2}{N} \ll 1$ and $\frac{s_2^3}{N^2} \ll 1$.

Consider two examples. Let $N = 2^{64}$, $\delta = 2^{-34}$. Then for `CTR_DRBG` we obtain admissible $s_1 \simeq 2^{15}$, while for **G2I** we get $s_2 > 2^{35}$

Let $N = 2^{128}$, $\delta = 2^{-68}$. Then for `CTR_DRBG` we obtain admissible $s_1 \simeq 2^{30}$, while for **G2I** we get $s_2 > 2^{63}$

7 Conclusion

In this paper, we investigated pseudorandom number generators based on random permutations. We estimated upper and lower bounds for the probability to find at least two identical elements is an output sequence of a finite length. The difference between the upper and lower bounds is extremely small, and the collision probability for a true RNG always lies within these bounds. We showed that the PRNG on two n-bit permutations could pass the collision test for output sequences of lengths far beyond the birthday bound. Similar security and performance is achieved when a single $2n$-bit random permutation is used and only a half of bits is used as output.

References

[1] *ISO/IEC 18031:2011. Information technology – Security techniques – Random bit generation*, International standard.

[2] Urivskiy A., Rybkin A., Borodin M., "On some properties of PRNGs based on block ciphers in counter mode", *Electronic Notes on Discrete Mathematics*, **57** (2017), 211–218.

[3] Urivskiy A., "On Unpredictability of PRNGs Based on Multiple Block Ciphers", *Proc. XV International Symposium Problems of Redundancy in Information and Control Systems*, 2016, 162–165.

[4] Lucks S., "The Sum of PRPs Is a Secure PRF", Proc. Advances in Cryptology — EUROCRYPT 2000, LNCS 1807, 2000, 470–484.

[5] Patarin J., "A Proof of Security in $O(2n)$ for the Xor of Two Random Permutations", Proc. Information Theoretic Security — ICITS 2008, LNCS 5155, 2008, 232–248.

Probabilistic Properties of Modular Addition

Victoria Vysotskaya

JSC «InfoTeCS», Russia
NPK «Kryptonite», Russia
vysotskaya.victory@gmail.com

Abstract

We studied the applicability of differential cryptanalysis to cryptosystems based on operation of addition modulo 2^n. We obtained an estimate (accurate up to an additive constant) of expected value of entropy H_n in rows of DDT of corresponding mapping. Moreover, the k-th moments of 2^{H_n} are explored. In particular, asymptotic inequalities that describe the behavior of values $\mathbb{E}2^{H_n}$ and $\mathbb{D}2^{H_n}$ as $n \to \infty$ were obtained.

Keywords: modular addition, differential cryptanalysis, entropy of distribution.

1 Introduction

A number of cryptographic schemes use the operation of addition modulo 2^n for some $n > 1$. Denote $\mathbb{Z}_N$ the ring modulo N. The first function under consideration is $f : \mathbb{Z}_{2^n}^2 \to \mathbb{Z}_{2^n}$ defined by $f(x, y) = x \boxplus_n y$, where $\boxplus_n$ denotes addition in ring $\mathbb{Z}_{2^n}$, i.e. modulo 2^n, and $\oplus$ is bitwise exclusive-OR. We are interested in study of the function $P_n(\Delta x, \Delta f) : \mathbb{Z}_{2^n}^2 \to \mathbb{N}_0$:

$$P_n(\Delta x, \Delta f) = \frac{1}{2^{2n}} \Big| \{(x, y) \in \mathbb{Z}_{2^n}^2 : \Delta f = f(x \oplus \Delta x, y) \oplus f(x, y)\} \Big|.$$

(it is analogous to a special case of the differential probability of addition modulo 2^n studied in [1]).

In this work we study the properties of this operation through the concept of entropy. The article [2] investigated the function $2^n \cdot P_n(\Delta x, \Delta f)$, but all the results are similar in these two cases, therefore we will briefly describe what is already known.

The table of values of the function $P_n(\Delta x, \Delta f)$ is called a difference distribution table (DDT). The rows of this table are indexed by Δx and columns by Δf. In [2] it has been shown that this table has a special form: the table for

addition modulo 2^{n+1} is naturally expressed through a similar table for addition modulo 2^n. That is, if the matrix for $P_n(\Delta x, \Delta f)$ has the form

$$P_n = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$$

then matrix P_{n+1} has form

$$P_{n+1} = \frac{1}{2}\left[\begin{array}{cc|cc} 2A & B & 0 & B \\ C & D & C & D \\ \hline 0 & B & 2A & B \\ C & D & C & D \end{array}\right]$$

It was also shown that $A = D$ and $B = C$. This led to the following recurrent representation for the matrix P_n:

$$P_n = \left[\begin{array}{c|c} A_n & B_n \\ \hline B_n & A_n \end{array}\right], \tag{1}$$

where

$$A_n = \frac{1}{2}\left[\begin{array}{c|c} 2A_{n-1} & B_{n-1} \\ \hline B_{n-1} & A_{n-1} \end{array}\right], \quad B_n = \frac{1}{2}\left[\begin{array}{c|c} 0 & B_{n-1} \\ \hline B_{n-1} & A_{n-1} \end{array}\right]. \tag{2}$$

When considering $P_n(\Delta x, \Delta f)$ as a part of a cryptosystem from the point of view of differential cryptanalysis the following problem arises: for a given (or randomly chosen) Δx it is necessary to determine the minimum cardinality K_c of the set of numbers $\{\Delta f_1, \ldots, \Delta f_{K_c}\}$ such that

$$\sum_{i=1}^{K_c} P_n(\Delta x, \Delta f_i) \geqslant c,$$

where c, $0 < c \leqslant 1$, is some fixed constant. The value of K_c corresponds to the "degree of branching", that is, the coefficient by which the number of considered variants is multiplied when moving to the next round of the cryptosystem. In practice, it was found that for the distributions in DDT rows the described value $K_{\frac{1}{2}}$ does not exceed 2^H, where H is the entropy of this distribution (this is not true in the general case, for arbitrary distributions, it is enough to consider an example distribution $\{\frac{1}{4}, \frac{1}{2^n}, \ldots, \frac{1}{2^n}\}$ for sufficiently large n).

Therefore in this article we research the quantities H and 2^H since analysis of K_c seems much less trivial.

2 Main results

By definition the entropy in the i-th row of matrix P_n may be found according to the formula

$$H_n^i = -\sum_{j=0}^{2^n-1} P_n(i,j)\log_2 P_n(i,j),\ \ i = 0,\dots,2^n-1.$$

For convenience we denote

$$\alpha_n^i = \sum_{j=0}^{2^{n-1}-1} A_n(i,j), \quad \beta_n^i = \sum_{j=0}^{2^{n-1}-1} B_n(i,j)$$

and

$$\alpha_n = \sum_{i=0}^{2^{n-1}-1} \alpha_n^i, \quad \beta_n = \sum_{i=0}^{2^{n-1}-1} \beta_n^i.$$

Lemma 1.

$$H_{n+1}^i = \begin{cases} H_n^{i \bmod 2^n} + 1, & \text{if } \ i \in [2^{n-1}, 2^n-1] \cup [3\cdot 2^{n-1}, 2^{n+1}-1], \\ H_n^{i \bmod 2^n} + \beta_n^{i \bmod 2^n}, & \text{if } \ i \in [0, 2^{n-1}-1] \cup [2^n, 3\cdot 2^{n-1}-1]. \end{cases}$$

Proof. From (1) and (2) it is clear that for $i \in [2^{n-1}, 2^n-1] \cup [3\cdot 2^{n-1}, 2^{n+1}-1]$ the i-th row has the form $\frac{1}{2}\begin{bmatrix} a & b & a & b \end{bmatrix}$ and thus the entropy can be written as

$$H_{n+1}^i = -2\cdot \sum_{j=0}^{2^n-1} \frac{P_n(i,j)}{2}\log_2 \frac{P_n(i,j)}{2} = -\sum_{j=0}^{2^n-1} P_n(i,j)_{i,j}\log_2 \frac{P_n(i,j)}{2} =$$
$$= -\sum_{j=0}^{2^n-1} P_n(i,j)\log_2 P_n(i,j) + \sum_{j=0}^{2^n-1} P_n(i,j)\log_2 2 = H_n^{i \bmod 2^n} + 1.$$

On the other hand, for $i \in [0, 2^{n-1}-1] \cup [2^n, 3\cdot 2^{n-1}-1]$ we have the row of

form $\frac{1}{2}\begin{bmatrix}2a & b & 0 & b\end{bmatrix}$ and thus

$$H^i_{n+1} = -\sum_{j=0}^{2^{n-1}-1} P_n(i,j)\log_2 P_n(i,j) - 2\cdot\sum_{j=2^{n-1}}^{2^n-1}\frac{P_n(i,j)}{2}\log_2\frac{P_n(i,j)}{2} =$$

$$= -\sum_{j=0}^{2^{n-1}-1} P_n(i,j)\log_2 P_n(i,j) - \sum_{j=2^{n-1}}^{2^n-1} P_n(i,j)\log_2 P_n(i,j) + \sum_{j=2^{n-1}}^{2^n-1} P_n(i,j) =$$

$$= H_n^{i \bmod 2^n} + \beta_n^{i \bmod 2^n}.$$

□

Lemma 2. *For every $n \geqslant 1$*

$$\mathbb{E}H_{n+1} = \frac{n}{2} + \frac{\beta_n}{2^n} + \cdots + \frac{\beta_3}{8} + \frac{\beta_2}{4}.$$

Proof. Taking into account the previous lemma, we can write:

$$\mathbb{E}H_{n+1} = \frac{1}{2^{n+1}}\sum_{i=0}^{2^{n+1}-1} H^i_{n+1} = \frac{1}{2^n}\sum_{i=0}^{2^{n-1}-1}(H^i_n + \beta^i_n) + \frac{1}{2^n}\sum_{i=2^{n-1}}^{2^n-1}(H^i_n + 1) =$$

$$= \frac{1}{2^n}\sum_{i=0}^{2^n-1} H^i_n + \frac{1}{2^n}\sum_{i=0}^{2^{n-1}-1}\beta^j_n + \frac{1}{2} = \mathbb{E}H_n + \frac{\beta_n}{2^n} + \frac{1}{2}.$$

It remains to "unroll" this equality and note that $H_1 = 0$ and $\beta_1 = 0$. □

Lemma 3. *For every $n \geqslant 1$*

$$\beta_n = \frac{1}{3}\cdot 2^{n-1}(1 - 4^{1-n}).$$

Proof. Obviously, $\alpha^i_n + \beta^i_n = 1$, so $\alpha_n + \beta_n = 2^{n-1}$. From (2) it follows that

$$\beta_{n+1} = \beta_n + \frac{\alpha_n}{2}.$$

From the last two equalities it follows that

$$\beta_{n+1} = 2^{n-2} + \frac{\beta_n}{2}.$$

Unrolling this equality we come to

$$\beta_{n+1} = 2^{n-2} + \frac{\beta_n}{2} = 2^{n-2} + \frac{1}{2}\Big(\beta_{n-1} + \frac{\alpha_{n-1}}{2}\Big) = 2^{n-2} + 2^{n-4} + \frac{\beta_{n-1}}{4} =$$
$$= 2^{n-2} + 2^{n-4} + \cdots + 2^{-n} = \frac{2^{n-2}\big(1 - \big(2^{-2}\big)^n\big)}{1 - 2^{-2}} = \frac{1}{3} \cdot 2^n(1 - 4^{-n}).$$

□

Theorem 1. $\mathbb{E}H_n = \frac{2}{3}n + O(1)$ *as* $n \to \infty$.

Proof. Let us substitute values obtained in Lemma 3 into the representation of $\mathbb{E}H_{n+1}$ obtained in Lemma 2:

$$\mathbb{E}H_{n+1} = \frac{n}{2} + \frac{1}{6}(1 - 4^{1-n}) + \cdots + \frac{1}{6}(1 - 4^{-1}) = \frac{n}{2} + \frac{n}{6} + \frac{1}{3}(1 - 4^{1-n}) = \frac{2}{3}n + O(1).$$

So $\mathbb{E}H_n = \frac{2}{3}(n-1) + O(1) = \frac{2}{3}n + O(1)$. □

Now we will consider the q-th moment of a random variable 2^{H_n}:

$$\mathbb{E}\big(2^{H_n}\big)^q = \mathbb{E}2^{qH_n} = \frac{1}{2^n}\sum_{i=0}^{2^n-1} 2^{qe_{n,i}} = \frac{1}{2^n}\sum_{i=0}^{2^n-1} Q^{e_{n,i}},$$

where $e_{n,i}$ is the entropy in i-th row of matrix P_n and Q denotes 2^q. To avoid multilevel exponentiation we will use the notation $\mathcal{Q}(x) = Q^x$.

Corollary 1. $\mathbb{E}2^{qH_n} = \Omega\left(Q^{\frac{2}{3}n}\right)$.

Proof. It is sufficient to use the inequality of arithmetic and geometric means and the result of Theorem 1:

$$\mathbb{E}2^{qH_n} = \frac{1}{2^n}\sum_{i=0}^{2^n-1} 2^{qe_{n,i}} \geqslant \sqrt[2^n]{\prod_{k=1}^{2^n} 2^{qe_{n,i}}} = 2^{\mathbb{E}(qH_n)} = 2^{\frac{2}{3}qn} \cdot \Omega(1) = \Omega\left(Q^{\frac{2}{3}n}\right).$$

□

Lemma 4. *For* $i = 0, \ldots, 2^{n-1} - 1$

$$\beta_n^i = \begin{cases} 0, & \text{if } i = 0, \\ 2^{-(n-1-\lfloor \log_2 i \rfloor)}, & \text{otherwise.} \end{cases} \tag{3}$$

Proof. Let us prove by induction. For $n = 1$ the proposition is obvious as $B_1 = \left[\ 0\ \right]$. Now let's suppose that it is also true for β^i_{n-1}, $i = 0, \ldots, 2^{n-2} - 1$ and let us prove it for β^i_n.

For $2^{n-2} \leqslant i \leqslant 2^{n-1} - 1$ from (2) we get $\beta^i_n = \frac{1}{2}$ as the sum in any row of matrix $\left[\ B_{n-1} \,\middle|\, A_{n-1}\ \right]$ is 1. This agrees with (3) as $\lfloor \log_2 i \rfloor = n - 2$.

For $0 \leqslant i \leqslant 2^{n-2} - 1$ from (2) we have

$$\beta^i_n = \frac{1}{2}\beta^i_{n-1}.$$

and by the inductive hypothesis we come to (3). □

Remark. *The vector of values β^i_n has the following form:*

$$\left[0, \underbrace{\frac{1}{2^{n-1}}}_{1}, \underbrace{\frac{1}{2^{n-2}}, \frac{1}{2^{n-2}}}_{2}, \ldots, \underbrace{\frac{1}{8}, \ldots, \frac{1}{8}}_{2^{n-4}}, \underbrace{\frac{1}{4}, \ldots, \frac{1}{4}}_{2^{n-3}}, \underbrace{\frac{1}{2}, \ldots, \frac{1}{2}}_{2^{n-2}}\right].$$

For convenience we extend the definition (3) for $2^{n-1} \leqslant i \leqslant 2^n - 1$. Then according to Lemma 1,

$$e_{n,i} = \beta^{i \bmod 2^{n-1}}_{n-1} + \beta^{i \bmod 2^{n-2}}_{n-2} + \cdots + \beta^{i \bmod 4}_2.$$

Moreover, obviously, $e_{1,0} = e_{1,1} = 0$. For $k \in \{0, \ldots, n-2\}$ let us introduce sets

$$Z_k = \left\{ i \in \mathbb{Z} : 2^{n-k-1} \leqslant i \leqslant 2^{n-k} - 1 \right\}.$$

The set Z_k consists of integers which binary representation has the form $\underbrace{0 \ldots 0}_{k}\, 1\, \underbrace{* \cdots *}_{n-k-1}$. Let us denote $\omega_n = \sum_{i=0}^{2^n-1} \mathcal{Q}(e_{n,i})$. Then

$$\omega_n = \sum_{i=0}^{2^n-1} \mathcal{Q}(e_{n,i}) = \sum_{k=0}^{n-1} \sum_{i' \in Z_k} \mathcal{Q}\left(\sum_{c=1}^{k} \beta^{i' \bmod 2^{n-c}}_{n-c} + e_{n-k,i'} \right) + 1 =$$

$$= \sum_{k=0}^{n-1} \sum_{i' \in Z_k} \mathcal{Q}\left(\sum_{c=1}^{k} \beta^{i' \bmod 2^{n-c}}_{n-c} \right) \mathcal{Q}(e_{n-k,i'}) + 1$$

$$= \sum_{k=0}^{n-1} \mathcal{Q}\left(\sum_{c=0}^{k-1} 2^{-c} \right) \sum_{i' \in Z_k} \mathcal{Q}(e_{n-k,i'}) + 1 = \sum_{k=0}^{n-1} \mathcal{Q}\left(2 - 2^{-k+1}\right) \frac{\omega_{n-k}}{2} + 1.$$

Obviously,

$$\mathbb{E}\big(2^{H_n}\big)^q = \frac{\omega_n}{2^n}. \tag{4}$$

Thus we need to investigate the following recurrence relation:

$$f'(n) = \sum_{\ell=1}^{n-1} f'(\ell) \cdot \mathcal{Q}(2 - 2^{-n+\ell+1}) + 2, \tag{5}$$

First, we compare it with the similar relation:

$$\begin{aligned} f(n) &= \sum_{\ell=1}^{n-1} f(\ell) \cdot \mathcal{Q}(2 - 2^{-n+\ell+1}), n \geqslant 2 \\ f(1) &= 2. \end{aligned} \tag{6}$$

Let us denote $\Delta(n) = f'(n) - f(n)$.

Lemma 5. $\Delta(n) \leqslant f(n)$.

Proof. Let us prove by induction. Obviously,

$$0 = \Delta(1) \leqslant f(1) = 2.$$

Suppose the proposition is true for all $\ell \leqslant n$ (i.e. $\Delta(\ell) \leqslant f(\ell)$) and write down

$$\begin{aligned} f(n+1) &= \sum_{\ell=2}^{n} f(\ell) + \mathcal{Q}(2 - 2^{-n+\ell+1}) f(1), \\ \Delta(n+1) &= \sum_{\ell=2}^{n} \Delta(\ell) + \mathcal{Q}(2 - 2^{-n+\ell+1}) \Delta(1) + 2. \end{aligned}$$

For $n \geqslant 2$ we have $\mathcal{Q}(2 - 2^{-n+3}) \geqslant 1$, from which and the inductive hypothesis follows:

$$\begin{aligned} \Delta(n+1) &\leqslant \sum_{\ell=2}^{n} f(\ell) + \mathcal{Q}(2 - 2^{-n+\ell+1}) \Delta(1) + 2 \leqslant \\ &\leqslant \sum_{\ell=2}^{n} f(\ell) + 0 + 2 \leqslant \sum_{\ell=2}^{n} f(\ell) + \mathcal{Q}(2 - 2^{-n+\ell+1}) f(1) = f(n+1), \end{aligned}$$

and it is the required inequality. □

With the use of Lemma 5 we estimate $f'(n)$ as

$$f(n) \leqslant f'(n) = f(n) + \Delta(n) \leqslant 2f(n),$$

and will work with homogeneous equation (6).

Let us note that coefficients $\mathcal{Q}(2-2^{-n+\ell+1}) = Q^{2-2^{-n+\ell+1}}$ are bounded from above by the number $\mathcal{Q}(2)$. Then let us consider the next family of recurrence relations:

$$\widehat{f}_k(n) = \sum_{\ell=n-k}^{n-1} \mathcal{Q}(2-2^{-n+\ell+1})\widehat{f}_k(\ell) + \mathcal{Q}(2) \sum_{\ell=1}^{n-k-1} \widehat{f}_k(\ell),$$
$$\widehat{f}_k(1) = 2,$$

solutions to which bound $f(n)$ from above. Denote

$$\widehat{F}_k(n) = \sum_{\ell=1}^{n-1} \widehat{f}_k(\ell). \tag{7}$$

Then

$$\widehat{F}_k(n) - \widehat{F}_k(n-1) =$$
$$= \sum_{\ell=n-k}^{n-1} \mathcal{Q}(2-2^{-n+\ell+1})\big(\widehat{F}_k(\ell) - \widehat{F}_k(\ell-1)\big) + \mathcal{Q}(2)\widehat{F}_k(n-k-1),$$
$$\widehat{F}_k(1) = 2. \tag{8}$$

Note that this recurrence relation has constant "length" and can be solved using well-known methods. Let us first find the form of the characteristic polynomial

corresponding to this relation:

$$\lambda^{k+1} - \lambda^k =$$

$$= \sum_{\ell=n-k}^{n-1} (\lambda^{\ell-n+k+1} - \lambda^{\ell-n+k})\mathcal{Q}(2-2^{-n+\ell+1}) + \mathcal{Q}(2) =$$

$$= \sum_{\ell=n-k}^{n-1} \lambda^{\ell-n+k+1}\mathcal{Q}(2-2^{-n+\ell+1}) - \sum_{\ell=n-k}^{n-1} \lambda^{\ell-n+k}\mathcal{Q}(2-2^{-n+\ell+1}) + \mathcal{Q}(2) =$$

$$= \sum_{\ell=n-k}^{n-1} \lambda^{\ell-n+k+1}\mathcal{Q}(2-2^{-n+\ell+1}) - \sum_{\ell=n-k-1}^{n-2} \lambda^{\ell-n+k+1}\mathcal{Q}(2-2^{-n+\ell+2}) + \mathcal{Q}(2) =$$

$$= \mathcal{Q}(1)\lambda^{k+1-1} - \mathcal{Q}(2-2^{-k-1+2}) +$$

$$+ \sum_{\ell=n-k}^{n-2} \lambda^{\ell-n+k+1}\left(\mathcal{Q}(2-2^{-n+\ell+1}) - \mathcal{Q}(2-2^{-n+\ell+2})\right) + \mathcal{Q}(2).$$

Thus the final form of the characteristic polynomial is

$$\widehat{H}_k(\lambda) =$$

$$= \lambda^{k+1} - (1+\mathcal{Q}(1))\lambda^k - \sum_{\ell=0}^{k-2} \mathcal{Q}(2)\left(\mathcal{Q}(-2^{-k+\ell+1}) - \mathcal{Q}(-2^{-k+\ell+2})\right)\lambda^{\ell+1} -$$

$$- \mathcal{Q}(2)\left(1 - \mathcal{Q}(-2^{-k+1})\right).$$

We will denote $\widehat{\varphi}_s$ the coefficient of λ^s. Let $y_1, \ldots, y_{k+1}$ be the roots of this polynomial. It is known [3] that the solution to the equation (8) has form

$$\widehat{F}_k(n) = \widehat{\gamma}_1 y_1^n + \cdots + \widehat{\gamma}_{k+1} y_{k+1}^n \tag{9}$$

for some constant $\widehat{\gamma}_i$.

On the other hand, coefficients $\mathcal{Q}(2-2^{-n+\ell+1})$ decrease with growth of ℓ and reach the minimum value on the interval $\ell \in [1, n-k-1]$ at the point $\ell = n-k-1$, where the coefficient is $\mathcal{Q}(2-2^{-k})$. From this considerations we obtain a new family of recurrences limiting the original one from *below*:

$$\breve{f}_k(n) = \sum_{\ell=n-k}^{n-1} \mathcal{Q}(2-2^{-n+\ell+1})\breve{f}_k(\ell) + \mathcal{Q}(2-2^{-k}) \sum_{\ell=1}^{n-k-1} \breve{f}_k(\ell),$$

$$\breve{f}_k(1) = 2.$$

Just as it was done above we introduce

$$\breve{F}_k(n) = \sum_{\ell=1}^{n-1} \breve{f}_k(\ell). \tag{10}$$

Thus

$$\breve{F}_k(n) - \breve{F}_k(n-1) = \sum_{\ell=n-k}^{n-1} \mathcal{Q}(2-2^{-n+\ell+1})\big(\breve{F}_k(\ell) - \breve{F}_k(\ell-1)\big) + \mathcal{Q}(2-2^{-k})\breve{F}_k(n-k-1),$$

$$\breve{F}_k(1) = 2. \tag{11}$$

In this case the characteristic polynomial has the following form:

$$\breve{H}_k(\lambda) = \lambda^{k+1} - (1+\mathcal{Q}(1))\lambda^k - \sum_{\ell=0}^{k-2} \mathcal{Q}(2)\left(\mathcal{Q}(-2^{-k+\ell+1}) - \mathcal{Q}(-2^{-k+\ell+2})\right)\lambda^{\ell+1} - \\ - \mathcal{Q}(2)\left(\mathcal{Q}(-2^{-k}) - \mathcal{Q}(-2^{-k+1})\right). \tag{12}$$

We will denote $\breve{\varphi}_s$ the coefficient of λ^s. The solution to the equation (11) has the following form:

$$\breve{F}_k(n) = \breve{\gamma}_1 y_1^n + \cdots + \breve{\gamma}_{k+1} y_{k+1}^n, \tag{13}$$

where $y_1, \ldots, y_{k+1}$ are the roots of $\breve{H}_k(\lambda)$ and $\breve{\gamma}_i$ are some constants.

Consider the following family of polynomials ($t \in [0,1]$):

$$\widehat{u}_t(\lambda) = \lambda^{k+1} - (1+Q)\lambda^k - t\cdot\widehat{\varphi}_{k-1}\lambda^{k-1} - \cdots - t\cdot\widehat{\varphi}_0 \tag{14}$$

and the similar one for $\breve{\varphi}_i$ (denote it $\breve{u}_t(\lambda)$). We will prove the following lemmas describing these families (note that $\breve{\varphi}_i = \widehat{\varphi}_i$ for $i \geqslant 1$).

Lemma 6. *For every $t \in [0,1]$ the polynomials $\widehat{u}_t(\lambda)$ and $\breve{u}_t(\lambda)$:*

(a) have no root in the annulus $1 < |\lambda| \leqslant 2$, if $Q = 2$;

(b) have no root λ such that $|\lambda| = \frac{Q}{2} + 1$, if $Q > 2$.

Proof. We prove the case (a) by contradiction. Assume that $\widehat{u}_t(\lambda)$ has a root λ

such that $1 < |\lambda| \leqslant 2$. Then taking absolute values in both parts in the equality

$$\lambda^{k+1} - t \cdot \widehat{\varphi}_{k-1}\lambda^{k-1} - \cdots - t \cdot \widehat{\varphi}_0 = 3\lambda^k$$

and applying the triangle inequality, we get

$$|\lambda|^{k+1} + t \cdot \widehat{\varphi}_{k-1}|\lambda|^{k-1} + \cdots + t \cdot \widehat{\varphi}_0 \geqslant 3|\lambda|^k.$$

Then

$$|\lambda|^{k-1}\left(|\lambda|^2 - 3|\lambda| + t \cdot \widehat{\varphi}_{k-1}\right) \geqslant -t \cdot \widehat{\varphi}_{k-2}|\lambda|^{k-2} - \cdots - t \cdot \widehat{\varphi}_0.$$

Since the branches of the parabola $y(|\lambda|) = |\lambda|^2 - 3|\lambda| + t \cdot \widehat{\varphi}_{k-1}$ are directed upwards, it reaches its maximum on one of the boundaries of the considered segment. In our case

$$y(1) = y(2) = -2 + t \cdot \widehat{\varphi}_{k-1}.$$

That is,

$$|\lambda|^{k-1}(-2 + t \cdot \widehat{\varphi}_{k-1}) \geqslant -t \cdot \widehat{\varphi}_{k-2}|\lambda|^{k-2} - \cdots - t \cdot \widehat{\varphi}_0.$$

Dividing by $|\lambda|^{k-1}$ we get

$$-2 \geqslant -t \cdot \widehat{\varphi}_{k-1} - t \cdot \widehat{\varphi}_{k-2}|\lambda|^{-1} - \cdots - t \cdot \widehat{\varphi}_0|\lambda|^{-k+1}.$$

Noting that simultaneously $t \leqslant 1$ by premise and $|\lambda|^{-1} < 1$ in the considered annulus, we arrive at:

$$2 < \widehat{\varphi}_{k-1} + \widehat{\varphi}_{k-2} + \cdots + \widehat{\varphi}_0. \tag{15}$$

At the same time it is easy to prove that for $Q = 2$

$$\widehat{\varphi}_{k-1} + \widehat{\varphi}_{k-2} + \cdots + \widehat{\varphi}_0 = 2,$$

so we have come the contradiction with (15). The same line of reasoning works for $\breve{u}_t(\lambda)$ except that instead of the last equality we get strict inequality.

We turn to the case (b): $Q \geqslant 4$. If under this condition there is a root such that $|\lambda| = \frac{Q}{2} + 1$, then

$$\left(\frac{Q}{2} + 1\right)^{k+1} + \left(\frac{Q}{2} + 1\right)^{k-1} \cdot t \cdot \widehat{\varphi}_{k-1} + \cdots + t \cdot \widehat{\varphi}_0 \geqslant (Q+1) \cdot \left(\frac{Q}{2} + 1\right)^k.$$

As far as $\max\limits_i \widehat{\varphi}_i = \widehat{\varphi}_{k-1} = Q^{\frac{3}{2}} - Q$ and $t \leqslant 1$ then

$$\left(\frac{Q}{2}+1\right)^{k+1} - (Q+1)\left(\frac{Q}{2}+1\right)^{k} + \frac{2}{Q}\left(Q^{\frac{3}{2}}-Q\right)\left(\frac{Q}{2}+1\right)^{k} > 0$$

or

$$(\sqrt{Q}-2)^2 < 0,$$

which contradicts $Q \geqslant 4$. Absolutely the same arguments work for $\breve{u}_t(\lambda)$. $\square$

Lemma 7. *None of the derivatives of $\widehat{u}_t(\lambda)$ and $\breve{u}_t(\lambda)$ have a root λ such that $|\lambda| = \frac{Q}{2}+1$.*

Proof. We firstly note that polynomials $\widehat{u}_t(\lambda)$ and $\breve{u}_t(\lambda)$ differ only in the constant term, which implies equality of derivatives

$$\widehat{u}_t^{(s)}(\lambda) = \breve{u}_t^{(s)}(\lambda) \text{ for all } s \geqslant 1. \tag{16}$$

So we will prove the lemma only for $\widehat{u}_t(\lambda)$.

Suppose that there exists λ, $|\lambda| = \frac{Q}{2}+1$, such that $\widehat{u}_t^{(s)}(\lambda) = 0$. Then similarly to Lemma 6 we get:

$$\begin{aligned}(k+1)^{\underline{s}} \cdot \left(\frac{Q}{2}+1\right)^{k+1-s} + \\ + (k-1)^{\underline{s}} \cdot \left(\frac{Q}{2}+1\right)^{k-1-s} \cdot t\widehat{\varphi}_{k-1} + \cdots + 0^{\underline{s}} \cdot \left(\frac{Q}{2}+1\right)^{-s} \cdot t\widehat{\varphi}_0 \geqslant \\ \geqslant (Q+1)k^{\underline{s}} \cdot \left(\frac{Q}{2}+1\right)^{k-s}\end{aligned}$$

(here $x^{\underline{s}}$ denotes $x(x-1)\dots(x-s+1)$). As noted above, $\max\limits_i \widehat{\varphi}_i = Q^{\frac{3}{2}} - Q$, so

$$\begin{aligned}(k-1)^{\underline{s}}(Q^{\frac{3}{2}}-Q) \cdot \frac{2}{Q}\left(\frac{Q}{2}+1\right)^{k-s} \geqslant \\ \geqslant (Q+1)k^{\underline{s}} \cdot \left(\frac{Q}{2}+1\right)^{k-s} - (k+1)^{\underline{s}} \cdot \left(\frac{Q}{2}+1\right)^{k+1-s},\end{aligned}$$

therefore,

$$(k-s)(k-s+1)(Q^{\frac{3}{2}}-Q)\cdot\frac{2}{Q}\geqslant k(k-s+1)(Q+1)-k(k+1)\left(\frac{Q}{2}+1\right).$$

This inequality can be viewed as

$$a(Q,s)k^2+b(Q,s)k+c(Q,s)\geqslant 0.$$

But

$$\begin{cases} a(Q,s)<0, \text{ if } Q\neq 4,\\ a(Q,s)=0, \text{ otherwise.}\end{cases}$$

Moreover, in the case of $Q=4$, it is true that $b(Q,s)<0$. Thus, there exists a certain number k starting from which this inequality will not be satisfied. □

Lemma 8. *The polynomials $\widehat{u}_t(\lambda)$ and $\breve{u}_t(\lambda)$ have exactly one root λ such that $|\lambda|>\frac{Q}{2}+1$.*

Proof. For the considered polynomials it is known [4] that their roots are continuous functions of variable t. As

$$\widehat{u}_0(\lambda)=\breve{u}_0(\lambda)=\lambda^{k+1}-(1+Q)\lambda^k,$$

these two polynomials have 0 as a root of multiplicity k and $(1+Q)$ as a root of multiplicity one.

By Lemma 6, $\widehat{u}_t(\lambda)$ and $\breve{u}_t(\lambda)$ do not have roots in the annulus $1<|\lambda|\leqslant 2$ (for $Q=2$) or the circle $|\lambda|=\frac{Q}{2}+1$ (for $Q\geqslant 4$). Thus, all curves corresponding to the first k roots do not leave the circle $|\lambda|\leqslant 1$ (for $Q=2$) and the circle $|\lambda|<\frac{Q}{2}+1$ (for $Q\geqslant 4$). The curve corresponding to the last root does not leave the sets $|\lambda|>2$ and $|\lambda|>\frac{Q}{2}+1$ respectively. □

Note that $\widehat{H}_k(Q+1)<0$ since

$$\widehat{H}_k(Q+1)=(Q+1)^{k+1}-(Q+1)\cdot(Q+1)^k-\widehat{\varphi}_{k-1}\cdot(Q+1)^{k-1}-\cdots-\widehat{\varphi}_0,$$

and $\widehat{\varphi}_i > 0,\ i \in [0, k-1]$. On the other hand, $\widehat{\varphi}_i < Q^{\frac{3}{2}} - Q$ for $i \in [0, k-1]$, so

$$\begin{aligned}\widehat{H}_k(3Q) &= (3Q)^{k+1} - (Q+1)(3Q)^k - \widehat{\varphi}_{k-1} \cdot (3Q)^{k-1} - \cdots - \widehat{\varphi}_0 > \\ &> (3Q)^{k+1} - (Q+1)(3Q)^k - (Q^{\frac{3}{2}} - Q)\frac{(3Q)^k}{3Q-1} > \\ &> \frac{(3Q)^k}{3Q-1}\left(6Q^2 - Q^{\frac{3}{2}} - 4Q - 1\right) > \frac{(3Q)^k}{3Q-1}(5Q^2 - 4Q - 1) > 0,\end{aligned}$$

for $Q \geqslant 2$. Absolutely similar statements are true for $\breve{H}_k(Q+1)$ and $\breve{H}_k(3Q)$.

Hence by the intermediate value theorem both functions $\widehat{H}_k(\lambda)$ and $\breve{H}_k(\lambda)$ have a real root on the segment $[Q+1, 3Q]$ which can be found by halving the segment. In this case, for n steps the root can be found with an accuracy $O(2^{-n})$.

Then equalities (9) and (13) take form:

$$\widehat{F}_k(n) = \widehat{\gamma}_k \widehat{y}_k^n + \widehat{\rho}_k(n), \tag{17}$$

$$\breve{F}_k(n) = \breve{\gamma}_k \breve{y}_k^n + \breve{\rho}_k(n), \tag{18}$$

where $\widehat{y}_k, \breve{y}_k$ are maximum (by the absolute value) roots of polynomials $\widehat{H}_k(\lambda)$ and $\breve{H}_k(\lambda)$ respectively (they are real, positive and lie inside $[Q+1, 3Q]$ as we have proved). $\widehat{\gamma}_k$ and $\breve{\gamma}_k$ are some real positive constants. Next, we note that if $Q = 2$ then $\widehat{\rho}_k(n) = O(1)$ and $\breve{\rho}_k(n) = O(1)$ as $n \to \infty$. If $Q \geqslant 4$ then

$$\widehat{\rho}_k(n) = O\left(\left(\frac{Q}{2} + 1\right)^n\right), \quad \breve{\rho}_k(n) = O\left(\left(\frac{Q}{2} + 1\right)^n\right)$$

The case $Q = 2$ is illustrated on Fig. 1.

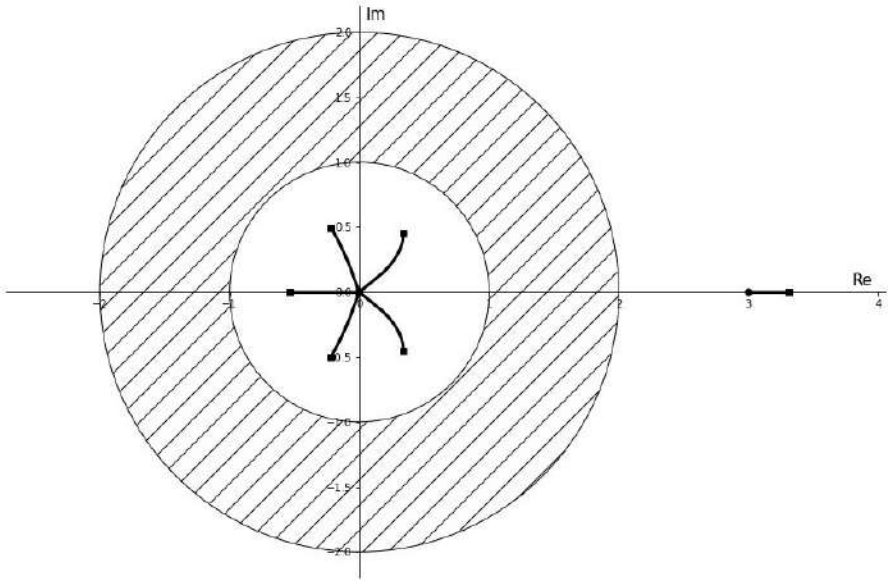


Figure 1: Trajectories traversed by roots of $\widehat{H}_5(\lambda)$ with t from 0 to 1; the round mark corresponds to $t = 0$, the square mark corresponds to $t = 1$

Lemma 9. *The difference $\widehat{y}_k - \breve{y}_k$ tends to zero as $k \to \infty$.*

Proof. Using Lemma 7, similarly to the proof of Lemma 8, it can be shown that the first and second derivatives of the functions $\widehat{H}_k(\lambda)$ and $\breve{H}_k(\lambda)$ have exactly one root, whose module exceeds $\frac{Q}{2} + 1$. We denote them by y'_k and y''_k respectively (by (16) these values are the same for $\widehat{H}_k$ and $\breve{H}_k$).

Since the function $\breve{H}_k(\lambda)$ can take negative values, $\min \breve{H}_k(\lambda) < 0$ and $\arg\min \breve{H}_k(\lambda) < \breve{y}_k$. At the same time $\arg\min \breve{H}_k(\lambda) = y'_k$. Thus $y'_k < \breve{y}_k$.

Carrying out similar reasoning, but considering $\breve{H}'_k(\lambda)$ instead of $\breve{H}_k(\lambda)$, it is easy to show that $y''_k < y'_k$. Then starting with some number k the following inequalities are held (see Fig. 2 for $Q = 2$):

$$\frac{Q}{2} + 1 \leqslant y''_k < y'_k < \breve{y}_k < \widehat{y}_k \leqslant 3Q.$$

Therefore functions $\widehat{H}_k(\lambda)$ and $\breve{H}_k(\lambda)$ are convex functions on $[y'_k, \widehat{y}_k]$, so for any $\delta \in [0, 1]$ holds the convexity inequality:

$$\widehat{H}_k\big(\delta y'_k + (1-\delta)\widehat{y}_k\big) \leqslant \delta \widehat{H}_k(y'_k) + (1-\delta)\widehat{H}_k(\widehat{y}_k).$$

Note that

$$\breve{y}_k = \delta y'_k + (1-\delta)\widehat{y}_k \quad \text{for } \delta = \frac{\widehat{y}_k - \breve{y}_k}{\widehat{y}_k - y'_k},$$

therefore, finally we get the following chain of inequalities:

$$\widehat{H}_k(\breve{y}_k) \leqslant \delta \widehat{H}(y'_k) + (1-\delta)\underbrace{\widehat{H}(\widehat{y}_k)}_{=0} = \frac{\widehat{y}_k - \breve{y}_k}{\widehat{y}_k - y'_k}\widehat{H}_k(y'_k) \leqslant$$

$$\leqslant \frac{\widehat{y}_k - \breve{y}_k}{\frac{5Q}{2} - 1}\widehat{H}_k\left(\frac{Q}{2}+1\right),$$

where at the last inequality we used the fact that $\widehat{H}_k(y'_k)$ is the minimum value of function $\widehat{H}_k$ on the ray $\left[\frac{Q}{2}+1, +\infty\right)$ and also that $\widehat{y}_k - y'_k \leqslant \frac{5Q}{2} - 1$. For function ν_k introduced in Lemma 7 the equality $\widehat{H}_k(\breve{y}_k) = -\nu_k$ obviously holds. Then we finally get:

$$\widehat{y}_k - \breve{y}_k \leqslant \frac{(-\frac{5Q}{2}+1)\nu_k}{\widehat{H}_k(\frac{Q}{2}+1)}.$$

It remains to show that the right part of the last inequality tends to zero

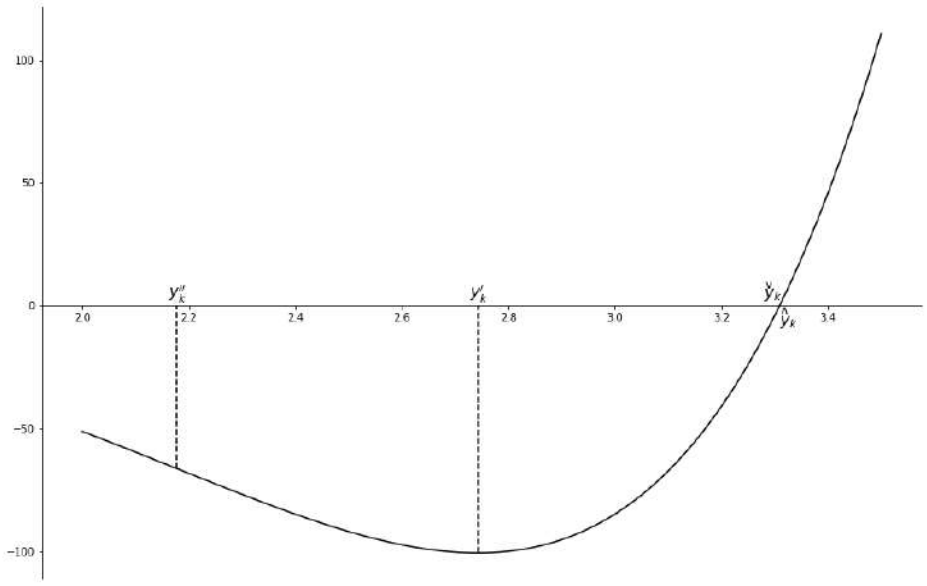


Figure 2: The plot of the function $\widehat{H}_5(\lambda)$

at $k \to \infty$. This follows from the tendency of ν_k to zero and also from the fact that

$$\widehat{H}_k\left(\frac{Q}{2}+1\right) =$$
$$= \left(\frac{Q}{2}+1\right)^{k+1} - (1+Q)\left(\frac{Q}{2}+1\right)^{k} - \varphi_{k-1}\left(\frac{Q}{2}+1\right)^{k} - \cdots - \varphi_0 =$$
$$= -\frac{Q}{2}\left(\frac{Q}{2}+1\right)^{k} - \varphi_{k-1}\left(\frac{Q}{2}+1\right)^{k} - \cdots - \varphi_0 \to -\infty \text{ as } k \to \infty.$$

□

Now we can estimate the value of $\mathbb{E}2^{H_n}$.

Theorem 2. *For all $\varepsilon > 0$ and all $q \in \mathbb{N}$ there exist real positive numbers $\widehat{z}$, $\check{z}$, c_1 and c_2 such that $|\widehat{z} - \check{z}| \leqslant \varepsilon$ and*

$$c_1\check{z}^n \lesssim \mathbb{E}2^{qH_n} \lesssim c_2\widehat{z}^n \text{ as } n \to \infty.$$

Proof. According to Lemma 8 polynomials $\widehat{H}_k(\lambda)$ and $\check{H}_k(\lambda)$ have exactly one root greater than $\left(\frac{Q}{2}+1\right)$. From (7) and (17) (also (10) and (18)) it follows that

$$\widehat{f}_k(n) = \widehat{F}_k(n) - \widehat{F}_k(n-1) \sim \widehat{\gamma}_k(\widehat{y}_k - 1)\widehat{y}_k^{n-1} = \widehat{\gamma}'_k\widehat{y}_k^n,$$
$$\check{f}_k(n) = \check{F}_k(n) - \check{F}_k(n-1) \sim \check{\gamma}_k(\check{y}_k - 1)\check{y}_k^{n-1} = \check{\gamma}'_k\check{y}_k^n.$$

At the same time,

$$\check{f}_k(n) \leqslant f(n) \leqslant \widehat{f}_k(n),$$

so

$$\check{\gamma}'_k\check{y}_k^n \lesssim f(n) \lesssim \widehat{\gamma}'_k\widehat{y}_k^n,$$
$$\check{\gamma}'_k\check{y}_k^n \lesssim f'(n) \lesssim 2\widehat{\gamma}'_k\widehat{y}_k^n = \widehat{\gamma}''_k\widehat{y}_k^n.$$

Finally,

$$c_1\check{z}^n = \check{\gamma}'_k \cdot \frac{\check{y}_k^n}{2^n} \lesssim \mathbb{E}2^{qH_n} \lesssim \widehat{\gamma}''_k \cdot \frac{\widehat{y}_k^n}{2^n} = c_2\widehat{z}^n,$$

moreover, Lemma 9 guarantees that $\widehat{z}$ and $\check{z}$ can be made arbitrarily close. □

Let us use the result of Theorem 2. Chose $\varepsilon = 10^{-20}$, Then such $\widehat{y}_k$ and $\check{y}_k$ exist that $|\widehat{y}_k - \check{y}_k| < \varepsilon$, that is they are both equal to $\widetilde{y}$ with the specified accuracy. This value will correspond to $\widetilde{z} = \frac{\widetilde{y}}{2}$. Moreover, value $\log_2 \widetilde{y} - 1$ is

interesting as

$$c_1 2^{n\cdot(\log_2 \widetilde{y}-1-\varepsilon)} \lesssim \mathbb{E}2^{qH_n} \lesssim c_2 \cdot 2^{n\cdot(\log_2 \widetilde{y}-1+\varepsilon)}.$$

Q	$\widetilde{y}$	$\widetilde{z}$	$\log_2 \widetilde{y} - 1$
2	3.30921306134212177240	1.65460653067106088620	0.72648818154049951037
4	5.80027271324371478340	2.90013635662185739172	1.53612073348070167305
8	10.53733221939675028493	5.26866610969837514246	2.39743775493525848727
16	19.61999911051941379160	9.80999955525970689580	3.29425307103935297681
32	37.19179236569642652549	18.59589618284821326274	4.21691237160283720288
64	71.45569997172021204310	35.72784998586010602155	5.15897719358341460680
128	138.69767829225482267831	69.34883914612741133915	6.11579982787398693748
256	271.32073664755570805747	135.66036832377785402874	7.08385550468282259524
512	533.89365096936984102274	266.94682548468492051137	8.06040858243800754807

Table 1: Approximate values associated with $\mathbb{E}Q^{H_n}$ for different values of Q

Now we can evaluate the variance of the value 2^{H_n}:

$$\mathbb{D}2^{H_n} = \mathbb{E}\big(2^{H_n}\big)^2 - \big(\mathbb{E}2^{H_n}\big)^2 = \mathbb{E}2^{2H_n} - \big(\mathbb{E}2^{H_n}\big)^2.$$

It is easy to observe from this table that $\big(\mathbb{E}2^{H_n}\big)^2 = o\big(\mathbb{E}2^{2H_n}\big)$. Thus, the variance $\mathbb{D}2^{H_n}$ can be estimated by the second moment:

$$c_1' \cdot 2^{(1.5361-\varepsilon)n} \lesssim \mathbb{D}2^{H_n} \lesssim c_2' \cdot 2^{(1.5361+\varepsilon)n}.$$

Finally we estimate the probability of deviating from the expectation $\mathbb{E}2^{H_n}$. We use Chebyshev's inequality:

$$\mathbb{P}\Big(\big|2^{H_n} - \mathbb{E}2^{H_n}\big| \geqslant a\Big) \leqslant \frac{\mathbb{D}2^{H_n}}{a^2}.$$

Choose $a = v^n\sqrt{\mathbb{D}2^{H_n}}, v > 1$ then

$$\mathbb{P}\Big(\big|2^{H_n} - \mathbb{E}2^{H_n}\big| \geqslant v^n\sqrt{\mathbb{D}2^{H_n}}\Big) \leqslant \frac{1}{v^{2n}} \to 0 \text{ as } n \to \infty.$$

Thus with probability tending to one

$$2^{H_n} \leqslant \mathbb{E}2^{H_n} + v^n\sqrt{\mathbb{D}2^{H_n}}$$

or, for example,

$$2^{H_n} = o\left(2^{0.76807n}\right) \text{ as } n \to \infty.$$

References

[1] Lipmaa H., Moriai S., "Efficient Algorithms for Computing Differential Properties of Addition", *Fast Software Encryption*, ed. Matsui M., 2002, 336–350.

[2] Vysotskaya V., *Some Properties of Modular Addition (Extended abstract)*, Cryptology ePrint Archive, https://eprint.iacr.org/2018/1103.

[3] Graham R. L, Knuth D. E., Patashnik O., *Concrete Mathematics: A Foundation for Computer Science*, (2 ed.), Addison-Welsey, 1994, ISBN: 978-0-201-55802-9, 672 pp.

[4] Tyrtyshnikov E., *A Brief Introduction to Numerical Analysis*, Birkhäuser Basel, 1997, ISBN: 978-0-8176-8136-4, 202 pp.

On the Way of Constructing $2n$-Bit Permutations from n-Bit Ones

Denis Fomin

National Research University Higher School of Economics, Russia
dfomin@hse.ru

Abstract

This work presents a generalisation of some known ways to construct $2n$-bit permutations using n-bit ones. Some new ways of constructing permutations on the basis of two well-known constructions will be proposed. Some new approaches presented in the work give a way to build permutations with low differential uniformity, high algebraic degree and high nonlinearity.

Keywords: S-box, permutation, boolean function, bent function.

1 Introduction

Permutations (or S-boxes) are core part of a huge class of modern cryptographic primitives such as block ciphers, hash functions and some stream ciphers. In recent years new ways of constructing permutations with low differential uniformity, high algebraic degree and high nonlinearity have been published [1, 2, 3, 15, 15]. Most of these works are devoted to the methods of constructing new classes of permutations on the basis of existing ones.

There are a lot of ways to build permutations from smaller one: constructions based on Feistel network [4, 5, 6], Misty network [7, 4, 8], SPN network [9, 10, 11] and some other constructions [12, 13, 15]. The first approach for constructing permutations is based on the so-called TU-decomposition [13, 14], which in can be considered as a generalisation of the Feistel network. Permutations built on this principle will be called "F-constructions" (Feistel-like constructions). The second approach is based on a way of representing an arbitrary permutation as a composition of transformations over spaces of smaller dimension. Permutations built on this principle will be called "G-constructions" (Generalised constructions). We will study some new ways to build permutations with low differential uniformity, high algebraic degree and high nonlinearity using these two approaches.

2 Definitions and Notations.

We will use the following notations and definitions. Let $\mathbb{F}_{2^n}$ be a finite field of size 2^n and V_n be the Boolean vector space of n elements.

Remark 1. *Every $a \in \mathbb{F}_{2^n}$ could be presented as a n-bit vector $a = (a_0, a_1, \ldots, a_{n-1})$, $a_i \in \mathbb{F}_2$, $i \in \overline{0, n-1}$. In this work we suppose that there is a bijective mapping from the field $\mathbb{F}_{2^n}$ to the vector space V_n.*

For any $a, b \in \mathbb{F}_{2^n}$ operation $\langle a, b \rangle$ is a dot product: $\sum_{i=0}^{n-1} a_i \cdot b_i$. For a boolean function $f : V_n \to V_1$ we can define the value $||f|| = \#\{x \in V_n : f(x) = 1\}$.

Let S be any function $S : \mathbb{F}_{2^n} \mapsto \mathbb{F}_{2^m}$. The security of the cryptographic functions strongly depends on the cryptographic properties of the used permutations, and properties of a permutation are the measures of resistance against known methods of cryptanalysis.

Definition 1. *The Walsh-Hadamard Transform (WHT) $W^S_{a,b}$ of a function S for fixed values $a \in \mathbb{F}_{2^n}$, $b \in \mathbb{F}_{2^m}$ is defined as follows:*

$$W^S_{a,b} = \sum_{x \in \mathbb{F}_{2^n}} (-1)^{\langle a,x \rangle + \langle b, S(x) \rangle}.$$

Definition 2. *The nonlinearity of a function S is denoted by N_S and defined by:*

$$N_S = 2^{n-1} - \frac{1}{2} \max_{a,b \neq 0} |W_S(a, b)| .$$

The linearity L_S of a S is defined as follows:

$$L_S = \frac{1}{2} \max_{a,b \neq 0} |W_S(a, b)| .$$

Definition 3. *A function $S : \mathbb{F}_{2^n} \mapsto \mathbb{F}_{2^m}$ is called a bent function when its nonlinearity is equal to $2^{n-1} - 2^{n/2-1}$.*

Let $n = 2m$, $x, y \in \mathbb{F}_{2^m}$. The Maiorana–McFarland construction [13] is the way to construct $2n$ bit bent-function from n bit functions and finite field multiplication: every function $g : V_m \times V_m \mapsto V_n$ that has the following form is a bent function:

$$g(x, y) = \pi(x) \cdot l(y) + f(x),$$

where $\pi : \mathbb{F}_{2^m} \mapsto \mathbb{F}_{2^m}$ is a permutation, $l : \mathbb{F}_{2^m} \mapsto \mathbb{F}_{2^m}$ is a linear permutation and $f : \mathbb{F}_{2^m} \mapsto \mathbb{F}_{2^m}$ is a function.

Definition 4. *The algebraic degree* $\deg(S)$ *of a function* S *is the minimum among all maximum numbers of variables of the terms in the algebraic normal form (ANF) of* $\langle a, S(x)\rangle$ *for all possible values* x *and* $a \neq 0$*:*

$$\deg(S) = \min_{a \in \mathbb{F}_{2^m} \backslash 0} \deg\left(\langle a, S(x)\rangle\right).$$

For any permutation on $\mathbb{F}_{2^n}$ the maximum value of the algebraic degree is $n-1$.

Definition 5. *For a given* $a \in \mathbb{F}_{2^n} \backslash 0, b \in \mathbb{F}_{2^m}$ *we consider*

$$\delta_S(a,b) = \#\left\{x \in \mathbb{F}_{2^n} | S(x+a) + S(x) = b\right\}.$$

The differential uniformity of a function S *is*

$$\delta_S = \max_{a \in \mathbb{F}_{2^n} \backslash 0, b} \delta_S(a,b).$$

We will say that two permutations S_1 and S_2 are linear equivalent if there exist two linear permutations L_1 and L_2: $S_1 = L_1 \circ S_2 \circ L_2$. We will also say that two permutations are affine equivalent if there exist two affine permutations A_1 and A_2: $S_1 = A_1 \circ S_2 \circ A_2$.

3 Chosen constructions and their properties

In this work we will build permutations over $\mathbb{F}_{2^m}$ and in our notation (see remark 1) it is equivalent to build permutation over V_{2m}. We can represent V_{2m} as a product: $V_m \times V_m$ as follows: $\overline{x} \in V_{2m}$, $\overline{x} = (x_0, \ldots, x_{m-1}, x_m, \ldots, x_{2m-1})$, $\overline{x} = (\overline{x}_1, \overline{x}_2)$, where $\overline{x}_1 = (x_0, \ldots, x_{m-1})$, $\overline{x}_2 = (x_m, \ldots, x_{2m-1})$. Moreover we will suppose that $\overline{x}_i$ is a representation of an element of the field $\mathbb{F}_{2^m}$.

3.1 Base constructions

In this work we will study two kinds of construction. The first one is based on the well-known TU-decomposition [13, 14]. Let F be a mapping $V_m \times V_m \mapsto V_m \times V_m$ and $F_1,\ F_2 : V_m \times V_m \mapsto V_m$ be the functions with the property: for any fixed value $\overline{v}_2$ the function $F_i(\overline{v}_1, \overline{v}_2)$, $i \in \overline{1,2}$ is a bijection. Then the definition $F_2^{-1}(\overline{x}_2, \overline{y}_2) = \overline{y}_1$ is correct and the following equations define the

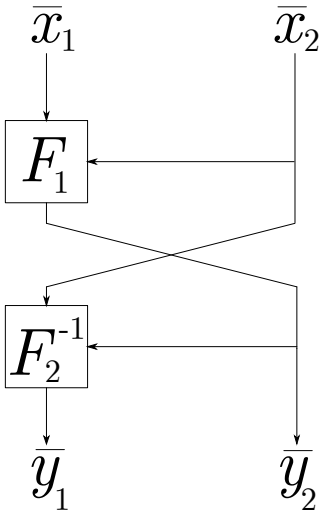


Figure 1: F construction

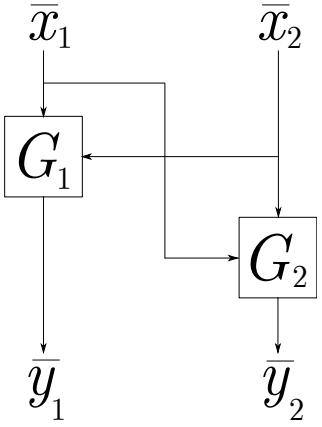


Figure 2: G construction

mapping $F(\overline{x}_1, \overline{x}_2) = (\overline{y}_1, \overline{y}_2)$ (see fig. 1):

$$\begin{cases} \overline{y}_2 = F_1(\overline{x}_1, \overline{x}_2) \\ \overline{x}_2 = F_2(\overline{y}_1, \overline{y}_2) \end{cases} \tag{1}$$

It's easy to show that the mapping F is correctly defined and F is a bijection [13, 15].

Proposition 1. *The amount of permutations that can be build by using the F-construction (see eq. (1)) is equal to $(2^m!)^{2^{m+1}}$.*

As we can see we can only build a limited number of permutations using F-construction. That's why we will also study the second type of construction.

Len $G\,(\overline{x}_1, \overline{x}_2) = (\overline{y}_1, \overline{y}_2)$ – be a permutation. Then we can define the mappings G_1 and G_2 as follows:

$$\begin{cases} \overline{y}_1 = G_1(\overline{x}_1, \overline{x}_2) \\ \overline{y}_2 = G_2(\overline{y}_1, \overline{y}_2) \end{cases} \tag{2}$$

Obviously, by defining mappings G_i, $i \in \overline{1,2}$ in a special way we can construct any permutation over $V_m \times V_m$ (see fig. 2).

3.2 Cryptographic properties of the chosen constructions

In this work we will study the ways to choose functions F_i and G_i, $i \in \overline{1,2}$ to build a permutations with high nonlinearity and low differential uniformity and high algebraic degree. Cryptographic properties of a permutation depends on it's sub-functions F_i and G_i, $i \in \overline{1,2}$.

Let $s'(x,y)$ be a function $V_m \times V_m \mapsto V_m$. We will call a punctured set of function $s'(x,y)$ the set of y such that $s'(x,y)$ is not a permutation of $x \in V_m$:

$$\dot{Y} = \{\dot{y} \,|\# \{s'(x,\dot{y}), x \in V_m\} < 2^m\}.$$

The value $\dot{y} \in \dot{Y}$ we will call a punctured value of a function s'.

If function s' have punctured values we can redefine it and construct a new function $s(x,y)$ such as s is a permutation of $x \in V_m$ for every fixed value $y \in V_m$:

$$s(x,y) = \begin{cases} s'(x,y), \ y \notin \dot{Y}; \\ \widehat{\pi}_y(x), \ y \in \dot{Y}; \end{cases}, \tag{3}$$

where $\widehat{\pi}_y(x)$ are permutations over V_m.

In this work we will focus on functions s' with only one punctured value $\dot{y}$. The general case can be examined similarly. Let's consider the following construction:

$$s(x,y) = \begin{cases} s'(x,y), \ \pi(y) \neq 0; \\ \widehat{\pi}(x), \ \pi(y) = 0; \end{cases}, \tag{4}$$

where $\pi, \widehat{\pi}$ are permutations over V_m, $s'(x,y) : V_{2m} \to V_m$ is a bijection for all fixed $y \neq \pi^{-1}(0)$. Let $g(x)$ be the function that is equal to the function $s(x,\dot{y})$.

3.2.1 Nonlinearity

Proposition 2. *Let* $s(x,y) = \begin{cases} s'(x,y), \ \pi(y) \neq 0; \\ \widehat{\pi}(x), \ \pi(y) = 0; \end{cases}$, *where* $\pi, \widehat{\pi}$ *are the permutations over* V_m, $s'(x,y) : V_{2m} \to V_m$ *is a bijection for any fixed* y, $y \neq \dot{y}$, $\dot{y}$ *is an punctured value of the function* $s(x,y)$.

Let $s(x,\dot{y}) = g(x)$. *Then the Walsh-Haramard Transform of the function*

$s(x, y)$ can be calculated as follows:

$$W^{s}_{\alpha\|\beta,\gamma} = \begin{cases} W^{s'}_{\alpha\|\beta,\gamma} + (-1)^{\langle\beta,\dot{y}\rangle}\left(W^{\widehat{\pi}}_{\alpha,\gamma} - W^{g}_{\alpha,\gamma}\right), & \alpha \neq 0; \\ W^{s'}_{0\|\beta,\gamma} + (-1)^{\langle\beta,\dot{y}\rangle}\left(2\left\|\langle\gamma, g(x)\rangle\right\| - 2^{m}\right), & \alpha = 0, \gamma \neq 0; \\ W^{s'}_{0\|\beta,0}, & \alpha = 0, \gamma = 0. \end{cases} \quad (5)$$

Proof. To prove the proposition we'll use the definition of the WHT:

$$\begin{aligned} W^{s}_{\alpha\|\beta,\gamma} &= \sum_{x,y\in V_m} (-1)^{\langle\alpha,x\rangle+\langle\beta,y\rangle+\langle\gamma,s(x,y)\rangle} = \\ &= \sum_{\substack{x,y\in V_m \\ y\neq\dot{y}}} (-1)^{\langle\alpha,x\rangle+\langle\beta,y\rangle+\langle\gamma,s'(x,y)\rangle} + \sum_{x\in V_m} (-1)^{\langle\alpha,x\rangle+\langle\beta,\dot{y}\rangle+\langle\gamma,\widehat{\pi}(x)\rangle} = \\ &= \sum_{\substack{x,y\in V_m \\ y\neq\dot{y}}} (-1)^{\langle\alpha,x\rangle+\langle\beta,y\rangle+\langle\gamma,s'(x,y)\rangle} + (-1)^{\langle\beta,\dot{y}\rangle} \sum_{x\in V_m} (-1)^{\langle\alpha,x\rangle+\langle\gamma,\widehat{\pi}(x)\rangle} \pm \\ &\pm (-1)^{\langle\beta,\dot{y}\rangle} \sum_{x\in V_m} (-1)^{\langle\alpha,x\rangle+\langle\gamma,g(x)\rangle} = W^{s'}_{\alpha\|\beta,\gamma} + (-1)^{\langle\beta,\dot{y}\rangle}\left(W^{\widehat{\pi}}_{\alpha,\gamma} - W^{g}_{\alpha,\gamma}\right). \end{aligned}$$

If both α and γ are equal to 0 then $W^{s}_{0\|\beta,0} = W^{s'}_{0\|\beta,0}$.

Let's consider the case when $\alpha = 0$ and $\gamma \neq 0$:

$$W^{s}_{0\|\beta,\gamma} = W^{s'}_{0\|\beta,\gamma} + (-1)^{\langle\beta,\dot{y}\rangle}\left(2 \cdot \left\|\langle\gamma, g(x)\rangle\right\| - 2^{m}\right).$$

□

Remark 2. *We can get the upper bound for the WHT of $s(x, y)$. If $\alpha \neq 0$ then*

$$\left|W^{s}_{\alpha\|\beta,\gamma}\right| \leqslant \left|W^{s'}_{\alpha\|\beta,\gamma}\right| + \left|W^{\widehat{\pi}}_{\alpha,\gamma}\right| + \left|W^{g}_{\alpha,\gamma}\right|.$$

And if $\alpha = 0$ and $\gamma \neq 0$:

$$\left|W^{s}_{0\|\beta,\gamma}\right| \leqslant \left|W^{s'}_{0\|\beta,\gamma}\right| + \left|2^{m} - 2 \cdot \|g(x)\|\right|.$$

According to the equations above we can suppose that more punctured values potentially lead to lower nonlinearity.

Let's consider the case when the function $g(x)$ is equal to 0.

Corollary 1. *Let* $s(x, y) = \begin{cases} s'(x, y), \ \pi(y) \neq 0; \\ \widehat{\pi}(x), \ \pi(y) = 0; \end{cases}$, *where* $\pi, \widehat{\pi} \in S(V_m)$,

$s'(x,y) : V_{2m} \to V_m$ is a bijection for all y, $\pi(y) \neq 0$. Let $\dot{y} = \pi^{-1}(0)$ be the punctured value of the function s and $s'(x,\dot{y}) = 0$. Then the WHT of the function $s(x,y)$ can be calculated as follows:

$$W^{s}_{\alpha\|\beta,\gamma} = \begin{cases} W^{s'}_{\alpha\|\beta,\gamma} + (-1)^{\langle\beta,\dot{y}\rangle} \cdot W^{\hat{\pi}}_{\alpha,\gamma}, & \alpha \neq 0; \\ 0, & \alpha = 0, \gamma \neq 0; \\ W^{s'}_{0\|\beta,0}, & \alpha = 0, \gamma = 0. \end{cases} \tag{6}$$

Proof. To prove it we can construct the similar reasoning as in the proposition 2:

$$W^{s}_{\alpha\|\beta,\gamma} = \sum_{x,y\in V_m} (-1)^{\langle\alpha,x\rangle+\langle\beta,y\rangle+\langle\gamma,s(x,y)\rangle} =$$

$$= W^{s'}_{\alpha\|\beta,\gamma} + (-1)^{\langle\beta,\dot{y}\rangle} \left(W^{\hat{\pi}}_{\alpha,\gamma} - \sum_{x\in V_m} (-1)^{\langle\alpha,x\rangle} \right).$$

Let's notice that

$$\sum_{x\in V_m} (-1)^{\langle\alpha,x\rangle} = \begin{cases} 2^m, & \text{if } \alpha = 0 \\ 0, & \text{otherwise.} \end{cases},$$

and it that case

$$W^{\hat{\pi}}_{\alpha,\gamma} = 0, \text{ if } \alpha = 0, \gamma \neq 0,$$
$$W^{\hat{\pi}}_{\alpha,\gamma} = 2^m, \text{ if } \alpha = 0, \gamma = 0.$$

Let's show that if $\gamma \neq 0$ then $W^{s'}_{0\|\beta,\gamma} = 2^m \cdot (-1)^{\langle\beta,\dot{y}\rangle}$:

$$W^{s'}_{0\|\beta,\gamma} = \sum_{x,y\in V_m} (-1)^{\langle\beta,y\rangle+\langle s'(x,y),\gamma\rangle} =$$

$$= \sum_{y\in V_m} (-1)^{\langle\beta,y\rangle} \sum_{x\in V_m} (-1)^{\langle s'(x,y),\gamma\rangle} = 2^m \cdot (-1)^{\langle\beta,\dot{y}\rangle}.$$

□

Remark 3. *Without loss of generality we'll suppose that $\dot{y} = 0$. According to equations (1),(2) we can choose F_i (or G_i), $i = 1, 2$ independently that's why if we have a function $s(x,y)$ with one punctured value $\dot{y} \neq 0$ then we can consider an affine-equivalent function with punctured value $\dot{y} = 0$.*

Remark 4. *If $N_{s'}$ is a bent function then $N_{s'} \leqslant N_s \leqslant N_{s'} + L_{\widehat{\pi}}$, otherwise $N_{s'} - L_{\widehat{\pi}} \leqslant N_s \leqslant N_{s'} + L_{\widehat{\pi}}$.*

3.2.2 Differential uniformity

According to the equations (1),(2) we choose two functions. We can choose both functions to be equal to the following two functions $s_1, s_2 : V_m \times V_m \mapsto V_m$ and s_i has one punctured value that is defined by permutations π_i:

$$s_1(x,y) = \begin{cases} s'_1(x,y), \ \pi_1(y) \neq 0; \\ \widehat{\pi}_1(x), \ \pi_1(y) = 0; \end{cases},$$

$$s_2(x,y) = \begin{cases} s'_2\left(y, s_1(x,y)\right), \ \pi_2\left(s_1(x,y)\right) \neq 0; \\ \widehat{\pi}_2(y), \ \pi_2(s_1(x,y)) = 0; \end{cases},$$

where for all $i \in \overline{1,2}$ $\pi_i, \widehat{\pi}_i \in S\left(V_m\right)$, $s'_i(x,y) : V_{2m} \to V_m$ is a bijection for all $y \neq \pi_i^{-1}(0)$.

It is still an open question how to calculate WHT for a linear combinations of functions $s_1(x,y)$ and $s_2(x,y)$ but we can proof the proposition that will help us to build permutation $S(x,y) = s_1(x,y) \| s_2(x,y)$ with low differential uniformity.

Proposition 3. *Let $a_1, a_2, b_1, b_2 \in V_m$, then the number of solutions of the following system of equations (number of pairs $x, y \in V_m$):*

$$\begin{cases} s_1(x,y) \oplus s_1(x \oplus a_1, y \oplus a_2) = b_1 \\ s_2(x,y) \oplus s_2(x \oplus a_1, y \oplus a_2) = b_2 \end{cases}$$

greater or equal to the number of solutions of the following system:

1. *$a_2 \neq 0$:*

$$\begin{cases} \pi_1(y) \neq 0 \\ \pi_1(y \oplus a_2) \neq 0 \\ \pi_2(s'_1(x,y)) \neq 0 \\ \pi_2(s'_1(x \oplus a_1, y \oplus a_2)) \neq 0 \\ s'_1(x,y) \oplus s'_1(x \oplus a_1, y \oplus a_2) = b_1 \\ s'_2(y, s'_1(x,y)) \oplus s'_2(y \oplus a_2, s'_1(x \oplus a_1, y \oplus a_2)) = b_2 \end{cases} \tag{7}$$

2. $a_1 \neq 0$, $a_2 = 0$ *the number of solutions of the system* (7) *and the number of solutions of the following system:*

$$\begin{cases} \pi_1(y) = 0 \\ s'_2(y, \widehat{\pi}_1(x)) \oplus s'_2(y, \widehat{\pi}_1(x \oplus a_1)) = b_2 \end{cases} \tag{8}$$

The proof of the proposition is quite obvious.

In fact the equations (7) give us the way of choosing functions $s'_1(x_i, y_i)$ and $s'_2(y_i, y_o)$ and (as we can see later) help to reduce the number of possible constructions. The equation (8) gives us the limitations to the permutation $\widehat{\pi}_2$ for the fixed function $s'_2(y_i, y_o)$. If we consider the permutation S^{-1} we can try to make the same limitations for the permutation $\widehat{\pi}_2$ for the fixed function s'_1.

3.2.3 Algebraic degree

Let us consider the algebraic degree of the function (4).

$$\langle a, s(x, y)\rangle = \left\langle a, s'(x, y) \cdot \overline{I_0}(y) + \pi(x) \cdot I_0(y)\right\rangle,$$

where $I_0(y)$ is a function that is equal to 1 only when $\pi(y) = 0$, and equal to 0 otherwise, and function $\overline{I_0}(y)$ is equal to 0 only when $\pi(y) = 0$ and 1 otherwise.

It's quite easy to show that $\deg(I_0) = m$ because $\pi(y)$ is a permutation. At the same time $1 \leqslant \deg(\pi) \leqslant m - 1$. In fact that $I_0(y)$ depends only on y, and $\pi(x)$ depends only on x and if $\deg(\pi) = m - 1$ then $\deg(s) = 2m - 1$. This property specifies the way of constructing functions with high algebraic degree.

3.3 One way to choose coordinate functions

As we described above the cryptographic properties of permutations F and G that are defined by the equations (1) and (2) respectively depend on cryptographic properties of coordinate functions F_i and G_i, $i \in \overline{1, 2}$. In this work we decided to consider only coordinate functions with one punctured value.

The corollary 1 says that we should choose function $s'(x, y) : V_m \times V_m \mapsto V_m$ and permutation π with highest possible nonlinearity. The section 3.2.3 says that such a coordinate function will have a high algebraic degree. The proposition 3 says how to choose a couple of coordinate function for constructing permutation to have smaller differential uniformity. Without loss of generality we suppose that $\pi(0) = 0$.

In this work we will focus on the constructions that are similar to the well known Maiorana–McFarland construction: $s'(x,y) = \psi(x) \cdot \phi(y)$, where ψ, ϕ are the permutations over V_m and "$\cdot$" is a multiplicative operator of the finite field $\mathbb{F}_{2^m}$. If either ψ or ϕ is a linear permutation, then s' is a bent-function.

4 Some examples of constructions and their cryptographic properties

This section provide some ways to build permutations based on equations (1) and (2). There is not a full list of possible constructions. We will lead the following plan:

- study their cryptographic properties but focus on the differential uniformity of the constructions;
- consider the monomial choice of some parameters to simplify the construction;
- find some parameters that provide a way to build permutation with rather good cryptographic properties in some special cases;
- focus on the most interesting way $m = 4$.

4.1 Construction "0"

Let us consider the F-construction (see eq. (1)). Let's choose the functions $F_1(\overline{x}_1, \overline{x_2})$, $F_2(\overline{y}_1, \overline{y}_2)$ on the following way:

$$F_1(\overline{x}_1, \overline{x_2}) = \begin{cases} \pi_1(\overline{x}_1) \cdot \overline{x}_2, \ \overline{x}_2 \neq 0; \\ \widehat{\pi}_1(\overline{x}_1), \ \overline{x}_2 = 0. \end{cases};$$

$$F_2(\overline{y}_1, \overline{y_2}) = \begin{cases} \pi_2(\overline{y}_1) \cdot \overline{y}_2, \ \overline{y}_2 \neq 0; \\ \widehat{\pi}_2(\overline{y}_1), \ \overline{y}_2 = 0. \end{cases}.$$

Then according to the equation (1) $\overline{y}_2 = F_1(\overline{x}_1, \overline{x_2})$, $\overline{x}_2 = F_2(\overline{y}_1, \overline{y}_2)$. As we can see both F_1 and F_2 are bent functions and could have rather high nonlinearity (with the proper choice of $\widehat{\pi}_i$).

Let's find how to calculate $\overline{y}_1$ using $\overline{x}_1$, $\overline{x}_2$. First, we consider the case: $\overline{x}_2 \neq 0$, $\overline{y}_2 \neq 0$, then

$$\overline{x}_2 = \pi_2(\overline{y}_1) \cdot \overline{y}_2 = \pi_2(\overline{y}_1) \cdot \pi_1(\overline{x}_1) \cdot \overline{x}_2 \Rightarrow \overline{y}_1 = \pi_2^{-1}\left(\pi_1(\overline{x}_1)^{-1}\right).$$

As we can see the value $\overline{y}_1$ does not depend on $\overline{x}_2$ and be a function of $\overline{x}_1$. It means that such a construction certainly has far from optimal cryptographic properties.

This example shows us that even the best choice (in terms of nonlinearity) of coordinate functions F_i, $i \in \overline{1,2}$ can make the whole construction have far from good cryptographic properties.

4.2 Construction "A"

Let's consider functions $F_1(\overline{x}_1, \overline{x_2})$, $F_2(\overline{y}_1, \overline{y}_2)$ in the following way ("AA" construction in [15]):

$$F_1(\overline{x}_1, \overline{x_2}) = \begin{cases} \pi_1(\overline{x}_1) \cdot \overline{x}_2, \ \overline{x}_2 \neq 0; \\ \widehat{\pi}_1(\overline{x}_1), \ \overline{x}_2 = 0. \end{cases};$$

$$F_2(\overline{y}_1, \overline{y_2}) = \begin{cases} \pi_2^{-1}(\overline{y}_1) \cdot \overline{y}_2^{-1}, \ \overline{y}_2 \neq 0; \\ \widehat{\pi}_2^{-1}(\overline{y}_1), \ \overline{y}_2 = 0. \end{cases}.$$

Let's find the formula to calculate $\overline{y}_1$. First we consider the case $\overline{y}_2 \neq 0$:

$$\overline{x}_2 = \pi_2^{-1}(\overline{y}_1) \cdot \overline{y}_2^{-1} \Rightarrow \overline{x}_2 \cdot \overline{y}_2 = \pi_2^{-1}(\overline{y}_1) \Rightarrow \overline{y}_1 = \pi_2(\overline{x}_2 \cdot \overline{y}_2).$$

If $\overline{y}_2 = 0$ then $\overline{y}_1 = \widehat{\pi}_2(\overline{x}_2)$.

Now we can denote the permutation $S_A : V_m \times V_m \mapsto V_m \times V_m$, $S_A(\overline{x}_1, \overline{x}_2) = (\overline{y}_1, \overline{y}_2)$ by the following equations:

$$\overline{y}_2 = \begin{cases} \pi_1(\overline{x}_1) \cdot \overline{x}_2, \ \overline{x}_2 \neq 0; \\ \widehat{\pi}_1(\overline{x}_1), \ \overline{x}_2 = 0. \end{cases};$$

$$\overline{y}_1 = \begin{cases} \pi_2(\overline{x}_2 \cdot \overline{y}_2), \ \overline{y}_2 \neq 0; \\ \widehat{\pi}_2(\overline{x}_2), \ \overline{y}_2 = 0. \end{cases}.$$

We can correctly define the function $\overline{y}_1 = F_2^{-1}(\overline{x}_2, \overline{y}_2)$. According to the remark

4 and the fact that $\pi_1(\overline{x}_1) \cdot \overline{x}_2$ is a bent function:

$$2^{2m-1} - 2^{m-1} \leqslant N_{F_1} \leqslant N_{\pi_1(\overline{x}_1)\cdot\overline{x}_2} + L_{\widehat{\pi}_1}.$$

Let's consider the value $\overline{y}_1$ as a function of two variables $\overline{x}_1$, $\overline{x}_2$:

$$\begin{aligned}\overline{y}_1 &= \begin{cases} \pi_2\left((\overline{x}_2)^2 \cdot \pi_1(\overline{x}_1)\right), & \overline{x}_2 \neq 0, \overline{x}_1 \neq \pi_1^{-1}(0); \\ \pi_2(\overline{x}_2 \cdot \widehat{\pi}_1(\overline{x}_1)), & \overline{x}_2 = 0, \overline{x}_1 \neq \widehat{\pi}_1^{-1}(0); \\ \widehat{\pi}_2(\overline{x}_2), & \overline{x}_2 \neq 0, \overline{x}_1 = \pi_1^{-1}(0); \\ \widehat{\pi}_2(\overline{x}_2), & \overline{x}_2 = 0, \overline{x}_1 = \widehat{\pi}_1^{-1}(0). \end{cases} = \\ &= \begin{cases} \pi_2\left((\overline{x}_2)^2 \cdot \pi_1(\overline{x}_1)\right), & \overline{x}_2 \neq 0, \overline{x}_1 \neq \pi_1^{-1}(0); \\ \pi_2(0), & \overline{x}_2 = 0, \overline{x}_1 \neq \widehat{\pi}_1^{-1}(0); \\ \widehat{\pi}_2(\overline{x}_2), & \overline{x}_2 \neq 0, \overline{x}_1 = \pi_1^{-1}(0); \\ \widehat{\pi}_2(0), & \overline{x}_2 = 0, \overline{x}_1 = \widehat{\pi}_1^{-1}(0). \end{cases} \end{aligned} \tag{9}$$

And the last cases means that $\overline{y}_1$ as a function of two variables $F_2^{-1}(\overline{x}_1, \overline{x}_2)$ has two punctured values. As we mentioned earlier more punctured values potentially leads to lower nonlinearity. If $\pi_1(0) \neq \widehat{\pi}_1(0)$ then $\overline{y}_1$ is equal to a constant $(\pi_2(0))$ for $2^m - 1$ values $\overline{x}_1 \neq \widehat{\pi}_1^{-1}(0)$. That fact says that differential uniformity is rather high:

$$\delta_{F_2^{-1}}(0 \| a_2, \pi_2(0)) \geqslant 2^m - 2.$$

So later to simplicity we will consider that $\pi_1^{-1}(0) = \widehat{\pi}_1^{-1}(0) = 0$ in that case:

1. equation (9) has a simple representation:

$$\overline{y}_1 = \begin{cases} \pi_2\left((\overline{x}_2)^2 \cdot \pi_1(\overline{x}_1)\right), & \overline{x}_1 \neq 0; \\ \widehat{\pi}_2(\overline{x}_2), & \overline{x}_1 = 0. \end{cases};$$

2. using remark 4:

$$N_{\pi_2\left((\overline{x}_2)^2\cdot\pi_1(\overline{x}_1)\right)} - L_{\widehat{\pi}_2} \leqslant N_{F_2^{-1}} \leqslant N_{\pi_2\left((\overline{x}_2)^2\cdot\pi_1(\overline{x}_1)\right)} + L_{\widehat{\pi}_2}.$$

According to our suppositions we denote that 0 is a fixed point of the all permutations π_i, $\widehat{\pi}_i$, $i \in \overline{1,2}$

Definition 6. *Let $\overline{x}_1, \overline{x}_2 \in V_m$ then the permutation $S_A = (\overline{y}_1, \overline{y}_2)$, where*

$$\overline{y}_1 = \begin{cases} \pi_2\left(\left(\overline{x}_2\right)^2 \cdot \pi_1\left(\overline{x}_1\right)\right), & \overline{x}_1 \neq 0; \\ \widehat{\pi}_2\left(\overline{x}_2\right), & \overline{x}_1 = 0. \end{cases} \tag{10}$$

$$\overline{y}_2 = \begin{cases} \pi_1\left(\overline{x}_1\right) \cdot \overline{x}_2, & \overline{x}_2 \neq 0; \\ \widehat{\pi}_1\left(\overline{x}_1\right), & \overline{x}_2 = 0. \end{cases} \tag{11}$$

we will call "A"-type permutation.

Proposition 4. *Let the permutation π_2 from equation* (10) *is a linear permutation. Then it has differential uniformity larger than $2^m - 2$.*

Proof. Let's $\overline{x}_2 \neq 0$, $\overline{x}_2 \neq a_2$. $\overline{x}_1 \neq 0$, $\overline{x}_1 \neq a_1$ in the equations (10), (11):

$$\begin{cases} \pi_1\left(\overline{x}_1 + a_1\right) \cdot \left(\overline{x}_2 + a_2\right) + \pi_1\left(\overline{x}_1\right) \cdot \overline{x}_2 = \beta_2 \\ \pi_2\left(\left(\overline{x}_2 + a_2\right)^2 \cdot \pi_1\left(\overline{x}_1 + a_1\right)\right) + \pi_2\left(\overline{x}_2^2 \cdot \pi_1\left(\overline{x}_1\right)\right) = \beta_1 \end{cases}.$$

Let's consider the case $a_1 = 0$, $a_2 \neq 0$. We know that π_2 is a permutation and $a_2 \neq 0$:

$$\begin{cases} \pi_1\left(\overline{x}_1\right) \cdot \left(\overline{x}_2 + a_2\right) + \pi_1\left(\overline{x}_1\right) \cdot \overline{x}_2 = \beta_2 \\ \pi_2\left(\left(\overline{x}_2 + a_2\right)^2 \cdot \pi_1\left(\overline{x}_1\right)\right) + \pi_2\left(\overline{x}_2^2 \cdot \pi_1\left(\overline{x}_1\right)\right) = \beta_1 \end{cases} \Rightarrow$$

$$\begin{cases} \pi_1\left(\overline{x}_1\right) = \beta_2 \cdot a_2^{-1} \\ \pi_2\left(\left(\overline{x}_2^2 + a_2^2\right) \cdot \pi_1\left(\overline{x}_1\right) + \overline{x}_2^2 \cdot \pi_1\left(\overline{x}_1\right)\right) = \beta_1 \end{cases} \Rightarrow$$

$$\begin{cases} \pi_1\left(\overline{x}_1\right) = \beta_2 \cdot a_2^{-1} \\ \pi_2\left(a_2^2 \pi_1\left(\overline{x}_1\right)\right) = \beta_1 \end{cases} \Rightarrow \begin{cases} \pi_1\left(\overline{x}_1\right) = \beta_2 \cdot a_2^{-1} \\ \pi_2\left(a_2 \cdot \beta_2\right) = \beta_1 \end{cases}$$

The value $\overline{x}_1$ is not equal to two values: $\overline{x}_1 \neq 0$, $\overline{x}_1 \neq a_1$ that's why the differential uniformity is greater than $2^m - 2$. □

As in the case of "0"-construction if π_2 is a linear permutation than $\pi_2\left(\left(\overline{x}_2\right)^2 \cdot \pi_1\left(\overline{x}_1\right)\right)$ is a bent function and F_2^{-1} potentially has larger nonlinearity in comparison with the case when π_2 is a nonlinear permutation. At the same time if π_2 is a linear permutation than the whole "A"-type permutation has rather large differential uniformity.

There is still an open question how to choose π_i, $\widehat{\pi}_i$, $i \in \overline{1,2}$. Let's consider a monomial choice of permutations π_i, $i \in \overline{1,2}$ and focus on the most interesting case $m = 4$.

We will study permutations $x \mapsto x^d$, $\mathrm{GCD}\,(d, 2^m - 2) = 1$ and following the Fermat's little theorem $d < 2^m - 2$. Equations (10), (11) has the following representation:

$$\overline{y}_2 = \begin{cases} \overline{x}_1^{\alpha} \cdot \overline{x}_2, \ \overline{x}_2 \neq 0; \\ \widehat{\pi}_1\left(\overline{x}_1\right), \ \overline{x}_2 = 0. \end{cases};$$

$$\overline{y}_1 = \begin{cases} \left(\overline{x}_2^2 \cdot \overline{x}_1^{\alpha}\right)^{\beta}, \ \overline{x}_1 \neq 0; \\ \widehat{\pi}_2\left(\overline{x}_2\right), \ \overline{x}_1 = 0. \end{cases} = \begin{cases} \overline{x}_2^{2\beta} \cdot \overline{x}_1^{\alpha\beta}, \ \overline{x}_1 \neq 0; \\ \widehat{\pi}_2\left(\overline{x}_2\right), \ \overline{x}_1 = 0. \end{cases}.$$

The proposition 4 says that permutation x^{β} should not be a linear one. There are only 8 d such as $\mathrm{GCD}\left(d, 2^4 - 2\right) = 1$: $d \in \{1, 2, 4, 7, 8, 11, 13, 14\}$ and if $d \in \{1, 2, 4, 8\}$ then x^d is a linear permutation.

Let's $\alpha \in \{1, 2, 4, 7, 8, 11, 13, 14\}$ and $\beta \in \{7, 11, 13, 14\}$. For any α the function $\overline{x}_1^{\alpha} \cdot \overline{x}_2$ is a bent function and f.

$$8 \leqslant L_{F_1} \leqslant 8 + L_{\widehat{\pi}_1} = 12,$$

because for any $\widehat{\pi}_i$ it's linearity is equal to or greater than 4. And similarly for F_2^{-1}. The considered function $\left(\overline{x}_2^2 \cdot \overline{x}_1^{\alpha}\right)^{\beta}$ is not a bent function and it's nonlinearity is equal to 16 and

$$12 \leqslant L_{F_2^{-1}} \leqslant 20.$$

We've implemented such a construction to build a permutation S_A and founded out that for $\alpha \in \{1, 2, 4, 7, 8, 11, 13, 14\}$ and $\beta \in \{7, 11, 13, 14\}$ we can find $\widehat{\pi}_i$, $i \in \overline{1,2}$ such as the permutation S_A has:

- $L_{F_1} = 12$,
- $L_{F_2^{-1}} = 20$,
- $L_{S_A} = 20$,
- $\delta_{S_A} = 6$,
- $\deg\left(S_A\right) = 7$.

We also experimentally founded out that $\widehat{\pi}_i$ could be any nonlinear monomial permutation.

We know that $N_{S_A} \geqslant \max\left\{N_{F_1}, N_{F_2^{-1}}\right\}$ so we've found permutations that have the best nonlinearity among all that have $N_{F_2^{-1}} = 20$.

It must be noted that π_i, $\widehat{\pi}_i$, $i \in \overline{1,2}$ may not be monomial permutations and using a personal computer and proposition 3 permutations with the same cryptographic properties could be easily found (an example can be found in [15]).

4.3 Construction "B"

Let's the functions $F_1(\overline{x}_1, \overline{x_2})$, $F_2(\overline{y}_1, \overline{y}_2)$ from the equation (1) are equal to:

$$F_1(\overline{x}_1, \overline{x_2}) = \begin{cases} \overline{x}_1 \cdot \pi_1(\overline{x}_2), \ \pi_1(\overline{x}_2) \neq 0; \\ \widehat{\pi}_1(\overline{x}_1), \ \pi_1(\overline{x}_2) = 0. \end{cases};$$

$$F_2(\overline{y}_1, \overline{y_2}) = \begin{cases} \overline{y}_1 \cdot \pi_2(\overline{y}_2)^{-1}, \ \pi_2(\overline{y}_2) \neq 0; \\ \widehat{\pi}_2^{-1}(\overline{y}_1), \ \pi_2(\overline{y}_2) = 0. \end{cases}.$$

According to equations above both F_1 and F_2 are bent functions.

Definition 7. *Let $\overline{x}_1, \overline{x}_2 \in V_m$ then the permutation $S_B = (\overline{y}_1, \overline{y}_2)$ that is defined as follows*

$$\overline{y}_1 = \begin{cases} \overline{x}_2 \cdot \pi_2(\overline{y}_2), \ \pi_2(\overline{y}_2) \neq 0; \\ \widehat{\pi}_2(\overline{x}_2), \ \pi_2(\overline{y}_2) = 0. \end{cases}; \tag{12}$$

$$\overline{y}_2 = \begin{cases} \overline{x}_1 \cdot \pi_1(\overline{x}_2), \ \pi_1(\overline{x}_2) \neq 0; \\ \widehat{\pi}_1(\overline{x}_1), \ \pi_1(\overline{x}_2) = 0. \end{cases}. \tag{13}$$

we will call "B"-type permutation.

It's easy to show that an inverse permutation for an "B"-type permutation is a "B"-type permutation:

$$\overline{x}_2 = \begin{cases} \overline{y}_1 \cdot \pi_2(\overline{y}_2)^{-1}, \ \pi_2(\overline{y}_2) \neq 0; \\ \widehat{\pi}_2^{-1}(\overline{y}_1), \ \pi_2(\overline{y}_2) = 0. \end{cases}. \tag{14}$$

$$\overline{x}_1 = \begin{cases} \overline{y}_2 \cdot \pi_1 \left(\overline{x}_2\right)^{-1}, \ \pi_1 \left(\overline{x}_2\right) \neq 0; \\ \widehat{\pi}_1^{-1} \left(\overline{y}_2\right), \ \pi_1 \left(\overline{x}_2\right) = 0. \end{cases} ; \tag{15}$$

As earlier we will suppose that $\pi_i(0) = 0$, $\widehat{\pi}_i(0) = 0$, $i \in \overline{1,2}$.

Proposition 5. *Let $H < S\left(V_m\right)$ — be the group of linear permutations. Than if $\pi_2 \in H$ or $\pi_1 \in x^{-1}H$ then $\delta^{S_B} \geqslant 2^m - 2$.*

Proof. First, we consider the case $\pi_2 \in H$. Let $a_1, b_1, b_2 \in V_m$ and $\overline{x}_2 \neq 0$, $\overline{x}_1 \neq a_1$, $\overline{x}_1 \neq 0$. Let's find the number of solutions of the following system:

$$\begin{cases} \overline{x}_1 \cdot \pi_1 \left(\overline{x}_2\right) + \left(\overline{x}_1 + a_1\right) \cdot \pi_1 \left(\overline{x}_2\right) = b_1 \\ \overline{x}_2 \cdot \pi_2 \left(\overline{x}_1 \cdot \pi_1 \left(\overline{x}_2\right)\right) + \overline{x}_2 \cdot \pi_2 \left(\left(\overline{x}_1 + a_1\right) \cdot \pi_1 \left(\overline{x}_2\right)\right) = b_2 \end{cases}$$

Using the fact that π_2 is a linear permutation:

$$\begin{cases} \overline{x}_1 \cdot \pi_1 \left(\overline{x}_2\right) + \left(\overline{x}_1 + a_1\right) \cdot \pi_1 \left(\overline{x}_2\right) = b_1 \\ \overline{x}_2 \cdot \pi_2 \left(\overline{x}_1 \cdot \pi_1 \left(\overline{x}_2\right)\right) + \overline{x}_2 \cdot \pi_2 \left(\left(\overline{x}_1 + a_1\right) \cdot \pi_1 \left(\overline{x}_2\right)\right) = b_2 \end{cases} \Rightarrow$$

$$\Rightarrow \begin{cases} a_1 \cdot \pi_1 \left(\overline{x}_2\right) = b_1 \\ \pi_2 \left(b_1\right) = b_2 \cdot \overline{x}_2^{-1} \end{cases}$$

And if we set any value to $\overline{x}_2 \neq 0$ and if $a_1 = b_1 \cdot \pi_1 \left(\overline{x}_2\right)^{-1}$, $\pi_2 \left(b_1\right) \cdot b_2^{-1} = \overline{x}_2^{-1}$ than the system above is true for any $\overline{x}_1 \neq a_1$, $\overline{x}_1 \neq 0$.

The case $\pi_2 \in x^{-1}H$ can be considering similar using equations (15), (14). □

Let's π_1 and π_2 be monomial permutations: $\pi_1 = x^\alpha$, $\pi_2 = x^\beta$ where α, β: $\mathrm{GCD}\left(\alpha, 2^4 - 2\right) = 1$, $\mathrm{GCD}\left(\beta, 2^4 - 2\right) = 1$. Then

$$\overline{y}_2 = \begin{cases} \overline{x}_1 \cdot \overline{x}_2^\alpha, \ \overline{x}_2 \neq 0; \\ \widehat{\pi}_1 \left(\overline{x}_1\right), \ \overline{x}_2 = 0. \end{cases} ;$$

$$\overline{y}_1 = \begin{cases} \overline{x}_2 \cdot \left(\overline{x}_1 \cdot \overline{x}_2^\alpha\right)^\beta, \ \overline{x}_1 \neq 0; \\ \widehat{\pi}_2 \left(\overline{x}_2\right), \ \overline{x}_1 = 0. \end{cases} = \begin{cases} \overline{x}_1^\beta \cdot \overline{x}_2^{\alpha\beta+1}, \ \overline{x}_1 \neq 0; \\ \widehat{\pi}_2 \left(\overline{x}_2\right), \ \overline{x}_1 = 0. \end{cases} .$$

And we will focus on the most interesting case $m = 4$. According to the proposition 5 $\alpha \in \{1, 2, 4, 8\}$, $\beta \in \{7, 11, 13, 14\}$.

Proposition 6. *Let $m = 4$ and $\pi_1 = x^\alpha$, $\pi_2 = x^\beta$ where α, β:*

$GCD\left(\alpha, 2^4-2\right) = 1$, $GCD\left(\beta, 2^4-2\right) = 1$. Than if $\alpha\beta + 1 \neq 14$ then $\delta_{S_B} \geqslant 2^m - 2$.

Proof. The value $\alpha\beta+1$ could be equal to 0, 8, 12 or 14. We'll consider $\alpha\beta+1 = 12$ (if $\alpha\beta+1$ is equal to 0 or 8 the proof is similar to the proof of the proposition 5).

Let $\overline{x}_i \neq 1, \overline{x}_i \neq 0, i \in \overline{1,2}$. Let's find the number of solutions of the following system:

$$\begin{cases} \overline{x}_1 \cdot \overline{x}_2^{\alpha} + (\overline{x}_1 + 1) \cdot (\overline{x}_2 + 1)^{\alpha} = 1 \\ \overline{x}_1^{\beta} \cdot \overline{x}_2^{12} + (\overline{x}_1 + 1)^{\beta} \cdot (\overline{x}_2 + 1)^{12} = 1 \end{cases} \Rightarrow$$

$$\Rightarrow \begin{cases} \overline{x}_1 = \overline{x}_2^{\alpha} \\ \overline{x}_2^{\beta\alpha} \cdot \overline{x}_2^{12} + (\overline{x}_2^{\beta} + 1^{\beta})^{\alpha} \cdot (\overline{x}_2 + 1)^{12} = 1 \end{cases} \Rightarrow$$

$$\Rightarrow \begin{cases} \overline{x}_1 + 1 = (\overline{x}_2 + 1)^{\alpha} \\ \overline{x}_2^{\beta\alpha} \cdot \overline{x}_2^{12} + (\overline{x}_2 + 1)^{\alpha\beta} \cdot (\overline{x}_2 + 1)^{12} = 1 \end{cases} \Rightarrow$$

$$\Rightarrow \begin{cases} \overline{x}_1 + 1 = (\overline{x}_2 + 1)^{\alpha} \\ \overline{x}_2^8 + (\overline{x}_2 + 1)^8 = 1 \end{cases} \Rightarrow \begin{cases} \overline{x}_1 + 1 = (\overline{x}_2 + 1)^{\alpha} \\ 1 = 1 \end{cases}$$

It's easy to show that if $\overline{x}_1$ is any possible value then $\overline{x}_2$ is not equal to 0 and 1. □

The proposition 6 gives us only 4 possible constructions:

1. $\pi_1(x) = x$, $\pi_2(x) = x^{13}$,
2. $\pi_1(x) = x^2$, $\pi_2(x) = x^{14}$,
3. $\pi_1(x) = x^4$, $\pi_2(x) = x^7$,
4. $\pi_1(x) = x^8$, $\pi_2(x) = x^{11}$.

We've implemented such a construction to build a permutation S_B and founded out that for all possible constructions we can find $\widehat{\pi}_i$, $i \in \overline{1,2}$ such as the permutation S_B has:

- $L_{S_A} = 20$,
- $\delta_{S_A} = 6$,
- $\deg(S_A) = 7$.

We also founded out that $\widehat{\pi}_i$ could be any nonlinear monomial permutation.

It must be noted that by analogy with "A"-type permutation π_i, $\widehat{\pi}_i$, $i \in \overline{1,2}$ may not be monomial permutations and using a personal computer and proposition 3 permutations with the same cryptographic properties could be easily found.

4.4 Construction "G"

Let's consider the construction that is defined by the equation (2). Let's show that using such a construction and propositions 1, 3 we can find a permutations with rather good cryptographic properties.

Let G_1 and G_2 be defined as follows (originally proposed in [16]):

$$\begin{aligned} G_1(\overline{x}_1, \overline{x}_2) = \overline{y}_1 = \begin{cases} \pi_1(\psi_1(\overline{x}_1) \cdot \phi_1(\overline{x}_2)), & \phi_1(\overline{x}_2) \neq 0; \\ \widehat{\pi}_1(\overline{x}_1), & \phi_1(\overline{x}_2) = 0. \end{cases} \\ G_2(\overline{x}_1, \overline{x}_2) = \overline{y}_2 = \begin{cases} \pi_2(\psi_2(\overline{x}_1) \cdot \phi_2(\overline{x}_2)), & \psi_2(\overline{x}_1) \neq 0; \\ \widehat{\pi}_2(\overline{x}_2), & \psi_2(\overline{x}_1) = 0. \end{cases} \end{aligned} \tag{16}$$

where π_i, $\widehat{\pi}_i$, ϕ_i, ψ_i, $i \in \{1,2\}$ are permutations.

Let's consider the most simple case, when π_i, ϕ_i, ψ_i, $i \in \{1,2\}$ are monomial permutations:

$$\begin{aligned} G_1(\overline{x}_1, \overline{x}_2) = \overline{y}_1 = \begin{cases} \overline{x}_1^{\alpha} \cdot \overline{x}_2^{\beta}, & \overline{x}_2 \neq 0; \\ \widehat{\pi}_1(\overline{x}_1), & \overline{x}_2 = 0. \end{cases} \\ G_2(\overline{x}_1, \overline{x}_2) = \overline{y}_2 = \begin{cases} \overline{x}_1^{\gamma} \cdot \overline{x}_2^{\delta}, & \overline{x}_1 \neq 0; \\ \widehat{\pi}_2(\overline{x}_2), & \overline{x}_1 = 0. \end{cases} \end{aligned} \tag{17}$$

It's easy to show that (17) is not always a permutation. The equation is defined a permutation (17) then and only then when

$$\begin{cases} G_1(\overline{x}_1, \overline{x}_2) = a_1 \\ G_2(\overline{x}_1, \overline{x}_2) = a_2 \end{cases}$$

has a solution for any $a_1, a_2 \in V_m$.

Let's consider the most interesting case $m = 4$. There are 8^4 sets of $(\alpha, \beta, \gamma, \delta)$ but using equation (7) we can cut this list to 748 possible constructions.

It's easy to show that set $(\alpha, \beta, \gamma, \delta)$ is linear equivalent to the following sets:

– $(\alpha \cdot d \pmod{2^m - 1}, \ \beta \cdot d \pmod{2^m - 1}, \ \gamma \cdot d \pmod{2^m - 1}, \ \delta \cdot d \pmod{2^m - 1})$ for any $d \in \{1, 2, 4, 8\}$;

– $(\alpha, \beta, \gamma, \delta)$, $(\gamma, \delta, \alpha, \beta)$, $(\beta, \alpha, \delta, \gamma)$, $(\delta, \gamma, \beta, \alpha)$.

And using linear equality we can enumerate the following 48 classes of permutations:

α	β	γ	δ	α	β	γ	δ	α	β	γ	δ	α	β	γ	δ
1	1	7	11	1	4	7	11	1	11	7	13	1	14	7	7
1	1	7	14	1	4	7	14	1	11	11	14	1	14	11	11
1	1	11	13	1	4	11	7	1	11	13	7	1	14	13	13
1	1	13	14	1	4	13	11	1	11	14	11	1	14	14	14
1	2	7	7	1	7	7	2	1	13	7	8	7	7	7	11
1	2	7	13	1	7	7	11	1	13	7	14	7	7	7	14
1	2	11	11	1	7	11	1	1	13	11	4	7	7	11	13
1	2	11	14	1	7	11	13	1	13	11	7	7	7	13	14
1	2	13	7	1	7	13	8	1	13	13	2	7	11	7	13
1	2	13	13	1	7	13	14	1	13	13	11	7	11	11	14
1	2	14	11	1	7	14	4	1	13	14	1	7	11	13	7
1	2	14	14	1	7	14	7	1	13	14	13	7	11	14	11

We've implemented such a construction to build a permutation and founded out that for all possible constructions we can find $\widehat{\pi}_i$, $i \in \overline{1, 2}$ the permutation has:

– $L_{S_A} = 20$,

– $\delta_{S_A} = 6$,

– $\deg(S_A) = 7$.

We also founded out that $\widehat{\pi}_i$ could be any nonlinear monomial permutation.

It must be noted that by analogy with "A"-type and "B"-type permutation that π_i, ϕ_i, ψ_i, $i \in \{1, 2\}$ may not be monomial permutations and using a personal computer and proposition 3 permutations with the same cryptographic properties could be easily found.

As example, if $m = 3$ then if (17) is a permutation then the equation (7) has too many solutions. But if we try to make a permutation G using the equation (16) we can build a permutation that has $N_G = 10$, $\delta_G = 4$ and $\deg(G) = 5$.

Conclusion

In this work we theoretically proved the cryptographic properties of the permutations that was originally proposed in [15]. It became possible to construct a new class of permutations using new results and theoretically proved their cryptographic properties.

References

[1] Yu Y., Wang M., and Li Y., "Constructing differential 4-uniform permutations from know ones", *IACR Cryptology ePrint Archive*, **2011:047**, 2011, http://eprint.iacr.org/2011/047.

[2] Fu S., Feng X., and Wu B., "Differentially 4-Uniform Permutations with the Best Known Nonlinearity from Butterflies", *IACR Cryptology ePrint Archive*, **2017:449**, 2017, http://eprint.iacr.org/2017/449.

[3] Peng J., and Tan C., "New differentially 4-uniform permutations by modifying the inverse function on subfields", *Cryptography and Communications*, **9**, 2017, 363-378.

[4] Canteaut A., Duval S., and Leurent G., "Construction of lightweight s-boxes using Feistel and MISTY structures (full version)", *IACR Cryptology ePrint Archive*, **2015:711**, 2015, http://eprint.iacr.org/2015/711.

[5] Lim C.H., "CRYPTON: A new 128-bit block cipher – specification and analysis", 1998.

[6] Gérard B., Grosso V., Naya-Plasencia M., and Standaert F.-X., "Block ciphers that are easier to mask: How far can we go?", *LNCS*, CHES, **8086**, ed. G. Bertoni and J.-S. Coron, Springer, 2013, 383–399.

[7] Matsui M., "New block encryption algorithm MISTY", *LNCS*, FSE, **1267**, ed. E. Biham, Springer, 1997, 54–68.

[8] Grosso V., Leurent G., Standaert F.-X., and Varici K., "Ls-designs: Bitslice encryption for efficient masked software implementations", *LNCS*, FSE, **8540**, ed. C. Cid and C. Rechberger, Springer, 2014, 18–37.

[9] Standaert F.-X., Piret G., Rouvroy G., Quisquater J.-J., and Legat J.-D., "ICEBERG: An involutional cipher efficient for block encryption in reconfigurable hardware", *LNCS*, FSE, **3017**, ed. Roy B.K. and Meier W., Springer, 2004, 279–299.

[10] Rijmen V. and Barreto P., "The KHAZAD block cipher", 2018.

[11] Lim C.H., "A revised version of Crypton – Crypton v1.0.", *LNCS*, FSE, **1636**, ed. Knudsen L.R., Springer, 1999, 31–45.

[12] Stallings W., "The Whirlpool secure hash function", *Cryptologia*, **30**, 2006, 55–67.

[13] McFarland R.L., "A family of difference sets in non-cyclic groups", *J. Comb. Theory*, **15**, 1973, 1–10.

[13] Biryukov A., Perrin L., and Udovenko A., "Reverse-engineering the s-box of Streebog, Kuznyechik and Stribobr1", *LNCS*, EUROCRYPT, **9665**, eds. M. Fischlin and J.-S. Coron, Springer, 2016, 372–402.

[14] Perrin L., "Cryptanalysis, Reverse-Engineering and Design of Symmetric Cryptographic Algorithms", 2017, base-search.net.

[15] Fomin D., "New Classes of 8-bit Permutations Based on a Butterfly Structure", *CTCrypt'18*, 2018.

[15] Perrin L, Udovenko A., and Biryukov A., "Cryptanalysis of a Theorem: Decomposing the Only Known Solution to the Big APN Problem (Full Version)", 2016, https://eprint.iacr.org/2016/539.

[15] Reynier A. and de la Cruz J., "On some methods for constructing almost optimal S-Boxes and their resilience against side-channel attacks", 2018, https://eprint.iacr.org/2018/618.

[16] Fomin D., "On approaches to construction of low-resource nonlinear transformations", *Review of applied industrial mathematics*, **25**, 2018, In Russian.

Matrix-Graph Approach for Studying Nonlinearity of Transformations on Vector Space

Vladimir Fomichev

[1]Security Code LLC, Moscow, Russia
[2]National Research Nuclear University «MEPhI», Russia
[3]Financial University under the Government of the Russian Federation, Russia
[4]Institute of Problems of Informatics (Russian Academy of Sciences), Russia
fomichev.2016@yandex.ru

Abstract

Let f be a transformation on a space P^n over a finite field P, $n > 1$, and f is given by the functions $f_0, \ldots, f_{n-1}$. We define the ternary matrix of nonlinearity $M_\Theta(f) = (m_{i,j})$, $0 \leqslant i < n$, $0 \leqslant j < n$, the element $m_{i,j}$ is equal to 0, or 1, or 2, if f_j depends on x_i fictitiously, or linearly, or nonlinearly. For any transformations $f^{(1)}, \ldots, f^{(t)}$ on P^n, $t \geqslant 1$, we prove the following inequality: $M_\Theta(f^{(1)} \cdot \ldots \cdot f^{(t)}) \leqslant M_\Theta(f^{(1)}) \cdot \ldots \cdot M_\Theta(f^{(t)})$. So the right side is the estimation of nonlinearity characteristics for the transformation $f^{(1)} \cdot \ldots \cdot f^{(t)}$. The ternary matrix M is called $\langle 2 \rangle$-primitive, if each element in M^t equals 2, $t \in \mathbb{N}$, the smallest t is called $\langle 2 \rangle$-exponent of matrix M ($\langle 2 \rangle \exp M$). The criterion is proved: ternary matrix M is $\langle 2 \rangle$-primitive if and only if M is primitive and contains the element "2", thereby, $0 \leqslant \langle 2 \rangle \exp M - \exp M \leqslant n$. We obtain the universal bound $\langle 2 \rangle \exp M \leqslant n^2 - n + 2$, and bounds for $\langle 2 \rangle$-primitive digraphs with circuit of length l, and also with loops.

Keywords: $\langle 2 \rangle$-primitive matrix (digraph), $\langle 2 \rangle$-exponent of matrix (digraph), cryptographic transformations, matrix of nonlinearity.

1 Introduction

Nonlinearity properties are necessary for the functions applied to protection of the data in information security systems. Differently the confidential parameters of the system (for ex., the keys) can be opened by the adversary by means of the quite simple decision of the system of linear equations.

Due to the wide usage of the composition of nonlinear functions in cryptographic algorithms, the task of calculating or evaluating the characteristics of the composition is relevant. Matrix-graph approach (MGA) is actively used for the estimation of the essential variables sets for the composition of nonlinear

transformations on the vector spaces. Mathematical basis of the MGA is made by the criteria of primitivity and local primitivity of sets of 0,1-matrices (or digraphs) and estimation of its exponents. The main results of this scientific direction, the history of which dates back to 1912 with the formulation of the problem by Frobenius, are presented in the review [1].

In this paper, the MGA is generalized and developed for estimation the characteristics of nonlinearity for the composition of transformations on the n-dimensional vector space. The proposed approach is based on the properties of ternary matrices of size $n \times n$ over the multiplicative semigroup $\{0, 1, 2\}$, and the properties of corresponding n-vertex digraphs, which arcs are labelled by the elements of the semigroup.

In this paper, we use the following notation:

$\mathbb{N}$ – the set of positive integers;

$\exp M (\exp \Gamma)$ – exponent of the matrix M (of the digraph Γ);

(i, j) – arc in digraph Γ, which incident at the vertices i and j;

$len\, w$ ($len\, c$) – length of the path w (of the circuit c) that equals to the number of the arcs in w (in c);

$w \bullet w'$ – concatenation of the paths w and w', where the last vertex of the path w coincides the first vertex of the path w';

$0 \leqslant i, j < n$ means that $0 \leqslant i < n$ and $0 \leqslant j < n$;

$\Longleftrightarrow$ – "if and only if".

2 Multiplicative monoids of ternary matrices and corresponding labelled digraphs

Let us consider a commutative semigroup $G = \{0, 1, 2\}$, where $\tau 0 = 0$ for any $\tau \in G$, $\tau\sigma = \max\{\tau, \sigma\}$ for any $\tau, \sigma \neq 0$. A matrix of any size over G is called the *ternary matrix*. We denote $(2)_n$ the matrix of size $n \times n$, in which each element equals 2. Call the ternary matrix *singular* if it contains all-zero row or all-zero column. Define the multiplication for ternary matrices $A = (a_{i,j})$ and $B = (b_{i,j})$: $AB = C = (c_{i,j})$, where C is a matrix of size $n \times n$, $c_{i,j} = \max\{a_{i,1}b_{1,j}, \ldots, a_{i,n}b_{n,j}\}$, and the multiplication is performed over

the semigroup G for any admissible i, j. Hence, $A(2)_n = (2)_n A = (2)_n$ for the non-singular matrix A (the matrix without all-zero rows and all-zero columns).

Denote $\mathbb{M}_n$ the monoid of all non-singular matrices of size $n \times n, n > 1$ (the multiplication in $\mathbb{M}_n$ is associative, the identity matrix is neutral by the multiplication). Define a partial order over the set of ternary matrices: $A \leqslant B \iff a_{i,j} \leqslant b_{i,j}$ for any admissible pairs (i, j). Let $A < B$ if $A \leqslant B$ and $a_{i,j} < b_{i,j}$ for some admissible pair (i, j). For $t \in \mathbb{N}$, $A, A', B, B' \in \mathbb{M}_n$, it follows from the rule of multiplication of ternary matrices that if $A \leqslant B$ and $A' \leqslant B'$, then $AA' \leqslant BB'$, hence $A^t \leqslant B^t$.

For $n > 1$, $0 \leqslant i, j < n$, there is the bijective correspondence between the ternary matrix $M = (m_{i,j})$ of size $n \times n$ and the labelled n-vertex digraph Γ, which arc (i, j) is assigned by the label "$m_{i,j}$". The label "0" is equivalent to the absence of an arc in the digraph. The matrix M over the semigroup G is called the *matrix of labels* of the digraph Γ and denoted by $M(\Gamma)$. The non-singular matrix corresponds to the digraph, in which each vertex has non-zero in-degrees and non-zero out-degrees. We denote $\mathbb{\Gamma}_n$ the multiplicative monoid of all labelled digraphs with the set of vertices $\{0, \ldots, n-1\}$. The digraph with n isolated vertices with loops is the identity (neutral) element in $\mathbb{\Gamma}_n$.

In $\Gamma \in \mathbb{\Gamma}_n$ we denote $(i, m_{i,j}, j)$ the arc (i, j) with the label $m_{i,j} \in \{0, 1, 2\}$. The semigroup multiplication operation for the digraphs Γ and Γ' is defined as follows: if there is the arc $(i, m_{i,r}, r)$ in Γ and there is the arc $(r, \mu_{r,j}, j)$ in Γ', then there is the arc $(i, m_{i,r}\mu_{r,j}, j)$ in $\Gamma\Gamma'$, where the multiplication of the labels is performed over G.

Due to the bijection $\mathbb{\Gamma}_n \leftrightarrow \mathbb{M}_n$, the arc $(i, m_{i,j}, j)$ in Γ corresponds to the element $m_{i,j}$ in M, where $m_{i,j}$ is placed in the ith row and jth column. The path $(v_0, \ldots, v_t)$ of length t from the vertex v_0 to the vertex v_t is labelled by the word $(m_1, \ldots, m_t)$, where m_s is the label of the arc $(v_{s-1}, v_s), s = 1, \ldots, t$. The product $m^{(t)} = m_1 \cdot \ldots \cdot m_t$ (which is calculated in the semigroup G) is called the *value of label* of the path $(v_0, \ldots, v_t)$. So, any path in Γ corresponds uniquely to the value of the label equals to 1 or 2. The path in Γ does not exist $\iff$ the label of the path contains "0", i.e. the value of the label equals to 0.

Theorem 1. *Let $\Gamma \in \mathbb{\Gamma}_n$, $M(\Gamma) = M = (m_{i,j})$, $t \in \mathbb{N}$, then $M(\Gamma^t) = M^t = (m^{(t)}_{i,j})$, where $m^{(t)}_{i,j}$ — the greatest value of the labels of all paths of length t from i to j.*

Proof. Use the inductive proof. For any pair (i, j) and $t = 1$, the proposition is

obvious, and $m_{i,j}^{(1)} = m_{i,j}$.

Suppose the proposition is true for $k < t$, where $t \geqslant 2$, and show that it is true for $k = t$.

Denote $E(j)$ the set of all vertices from which the arcs go to the vertex $j = 1, \ldots, n$. Without restricting the generality, let $E(j) = \{1, \ldots, r\}$ for any fixed j, then $m_{i,j} = 0$ as $i > r$. It follows from the equation $M^t = M^{t-1}M$ that

$$m_{i,j}^{(t)} = \max\{m_{i,1}^{(t-1)} m_{1,j}, \ldots, m_{i,n}^{(t-1)} m_{n,j}\} = \\ = \max\{m_{i,1}^{(t-1)} m_{1,j}, \ldots, m_{i,r}^{(t-1)} m_{r,j}\}.$$

In accordance with the inductive hypothesis, $m_{i,s}^{(t-1)}$ is equal to the greatest value of the labels of all paths from i to s of length $t-1$. This means that the product $m_{i,s}^{(t-1)} \cdot m_{s,j}$ is equal to the greatest value of the labels of all paths from i to j of length t provided that the vertex j is preceded by the vertex $s, s = 1, \ldots, r$. Then $m_{i,j}^{(t)}$ is the greatest value of the labels of all paths from i to j of length t. □

Corollary 1. *In Γ^t the arc (i, j) has the label with the value:*

1. *"0" $\Longleftrightarrow$ in Γ the vertex j is not reachable from the vertex i in t steps;*
2. *"1" $\Longleftrightarrow$ in Γ the label of any existing path from i to j of length t consists of t units;*
3. *"2" $\Longleftrightarrow$ in Γ the label of some path from i to j of length t contains the symbol "2".*

3 Nonlinear properties of transformations on the vector spaces

We denote $\{f_j(x_0, \ldots, x_{n-1}), j = 0, \ldots, n-1\}$ the set of the coordinate polynomials of the transformation $f\colon P^n \to P^n$. Let us associate the nonlinearity property with the characteristics of the coordinate functions. For $0 \leqslant i, j < n$, we construct the ternary matrix $M_\Theta(f) = (m_{i,j})$ of size $n \times n$, where the element $m_{i,j}$ in $M_\Theta(f)$ equals 0, or 1, or 2 $\Longleftrightarrow$ $f_j(x_0, \ldots, x_{n-1})$ depends on x_i fictitiously, or linearly, or nonlinearly . The corresponding labelled digraph $\Gamma_\Theta(f)$ with the set of vertices $\{0, \ldots, n-1\}$ is called the *digraph of*

nonlinearity of the transformation f. The function f is called *quite nonlinear* if $M_\Theta(f) = (2)_n$. Note, that any function satisfying the strict avalanche criterion is quite nonlinear [2, p.182].

For $0 \leqslant i, j < n$, $s = 1, \ldots, t$, we denote $f^{(s)}$ the transformation on P^n; $\{f_j^{(s)}(x_0, \ldots, x_{n-1})\}$ and $\{f_j^{[s]}(x_0, \ldots, x_{n-1})\}$ – the sets of coordinate polynomials of the transformations $f^{(s)}$ and $f^{(1)} \cdot \ldots \cdot f^{(s)}$; $M_\Theta(f^{(s)}) = (m_{i,j}^{(s)})$; $M_\Theta(f^{(1)} \cdot \ldots \cdot f^{(s)}) = (\mu_{i,j}^{[s]})$; $M_\Theta(f^{(1)}) \cdot \ldots \cdot M_\Theta(f^{(s)}) = (m_{i,j}^{[s]})$.

Theorem 2. *For any transformations $f^{(1)}, \ldots, f^{(t)}$ on P^n, $t \geqslant 1$, the following inequality is true*

$$M_\Theta(f^{(1)} \cdot \ldots \cdot f^{(t)}) \leqslant M_\Theta(f^{(1)}) \cdot \ldots \cdot M_\Theta(f^{(t)}).$$

Proof. Use the inductive proof. For $s = 1, \ldots, t$, $0 \leqslant i, j < n$, in the given notations, prove that $\mu_{i,j}^{[s]} \leqslant m_{i,j}^{[s]}$.

For $t = 1$ the theorem is obvious. For $t = 2$ by the rule of multiplication of ternary matrices we get

$$m_{i,j}^{[2]} = \max\{m_{i,0}^{(1)} \cdot m_{0,j}^{(2)}, \ldots, m_{i,n-1}^{(1)} \cdot m_{n-1,j}^{(2)}\}, \tag{1}$$

and by the rule of multiplication of transformations we get

$$f_j^{[2]}(x_0, \ldots, x_{n-1}) = f_j^{(2)}(f_0^{(1)}(x_0, \ldots, x_{n-1}), \ldots, f_{n-1}^{(1)}(x_0, \ldots, x_{n-1})). \tag{2}$$

Let $f_j^{(2)}(x_0, \ldots, x_{n-1})$ be a constant function, then due to (2) the function $f_j^{[2]}(x_0, \ldots, x_{n-1})$ is a constant too. Hence, $\mu_{i,j}^{[2]} = 0$, i.e. the theorem is correct.

Let $f_j^{(2)}(x_0, \ldots, x_{n-1})$ be a linear function, which essentially depends on arguments, for example, on $x_0, \ldots, x_r$, where $r < n$, and for $r < n-1$ does not essentially depend on $x_{r+1}, \ldots, x_{n-1}$. Then it follows from (2) that

$$f_j^{[2]}(x_0, \ldots, x_{n-1}) = a_0 f_0^{(1)}(x_0, \ldots, x_{n-1}) + \cdots + a_r f_r^{(1)}(x_0, \ldots, x_{n-1}), \tag{3}$$

where $a_0, \ldots, a_r$ – non-zero coefficients of the field P. On the condition, $m_{0,j}^{(2)} = \cdots = m_{r,j}^{(2)} = 1$, $m_{r+1,j}^{(2)} = \cdots = m_{n-1,j}^{(2)} = 0$ for $r < n-1$, and from the equation (1) we get

$$m_{i,j}^{[2]} = \max\{m_{i,0}^{(1)}, \ldots, m_{i,r}^{(1)}\}. \tag{4}$$

If the functions $f_0^{(1)}(x_0, \ldots, x_{n-1}), \ldots, f_r^{(1)}(x_0, \ldots, x_{n-1})$ do not essentially depend on x_i (i.e., fictitiously depend on x_i), then from the formula (3) it follows

that $f_j^{[2]}(x_0,\dots,x_{n-1})$ does not essentially depend on x_i. So, $\mu_{i,j}^{[2]}=0$, and the theorem is correct. If some of the functions $f_0^{(1)}(x_0,\dots,x_{n-1)},\dots,f_r^{(1)}(x_0,\dots,x_{n-1})$ essentially depend on x_i, then there is the linear or nonlinear dependence. Let for $l\leqslant r$, the functions $f_0^{(1)}(x_0,\dots,x_{n-1}),\dots,f_l^{(1)}(x_0,\dots,x_{n-1})$ linearly depend on x_i, and for $l<r$, the functions $f_{l+1}^{(1)}(x_0,\dots,x_{n-1}),\dots,f_r^{(1)}(x_0,\dots,x_{n-1})$ nonlinearly depend on x_i. Then for $l<r$, $m_{i,0}^{(1)}=\dots=m_{i,l}^{(1)}=1$, and $m_{i,l+1}^{(1)}=\dots=m_{i,r}^{(1)}=2$. Hence, for $l<r$, $m_{i,j}^{[2]}=2$ due to (4), so the theorem is correct. For $l=r$ due to (4) $m_{i,j}^{[2]}=1$, and due to (3) $f_j^{[2]}(x_0,\dots,x_{n-1})$ fictitiously or linearly depends on x_i. Therefore, $\mu_{i,j}^{[2]}\leqslant 1$, and the theorem is correct.

Let $f_j^{(2)}(x_0,\dots,x_{n-1})$ be a nonlinear function, which essentially depends on the arguments: for $0<p\leqslant r<n$, there is nonlinear dependence on $x_0,\dots,x_p$; for $p<r$ – linear dependence on $x_{p+1},\dots,x_r$; for $r<n-1$ – fictitious dependence on $x_{r+1},\dots,x_{n-1}$. Then for $0\leqslant i,j<n$, and $p<r$, the formula (1) transforms to

$$m_{i,j}^{[2]}=\max\{2m_{i,0}^{(1)},\dots,2m_{i,p}^{(1)},m_{i,p+1}^{(1)},\dots,m_{i,r}^{(1)}\},\tag{5}$$

and for $p=r$, the formula (1) transforms to

$$m_{i,j}^{[2]}=\max\{2m_{i,0}^{(1)},\dots,2m_{i,r}^{(1)}\}.\tag{6}$$

At the same time, it follows from (2) that

$$f_j^{[2]}(x_0,\dots,x_{n-1})=f_j^{(2)}(f_0^{(1)}(x_0,\dots,x_{n-1}),\dots,f_r^{(1)}(x_0,\dots,x_{n-1})).\tag{7}$$

If the functions $f_0^{(1)}(x_0,\dots,x_{n-1}),\dots,f_r^{(1)}(x_0,\dots,x_{n-1})$ fictitiously depend on x_i, then due to (7) $f_j^{[2]}(x_0,\dots,x_{n-1})$ fictitiously depends on x_i. So, $\mu_{i,j}^{[2]}=0$, and the theorem is correct for $t=2$.

Let some of $f_0^{(1)}(x_0,\dots,x_{n-1}),\dots,f_r^{(1)}(x_0,\dots,x_{n-1})$ essentially depend on x_i, i.e. there is linear or nonlinear dependence. If some of the functions $f_0^{(1)}(x_0,\dots,x_{n-1}),\dots,f_p^{(1)}(x_0,\dots,x_{n-1})$ essentially depend on x_i or for $p<r$ some of $f_{p+1}^{(1)}(x_0,\dots,x_{n-1}),\dots,f_r^{(1)}(x_0,\dots,x_{n-1})$ nonlinearly depends on x_i, then it holds from (5) and (6) $m_{i,j}^{[2]}=2$, and the theorem is correct. If for $p<r$, $f_0^{(1)}(x_0,\dots,x_{n-1}),\dots,f_p^{(1)}(x_0,\dots,x_{n-1})$ fictitiously depend on x_i, and some of $f_{p+1}^{(1)}(x_0,\dots,x_{n-1}),\dots,f_r^{(1)}(x_0,\dots,x_{n-1})$ linearly depends on x_i, then due to

(5) $m_{i,j}^{[2]} = 1$. Moreover, it follows from (7) that $f_j^{[2]}(x_0, \ldots, x_{n-1})$ fictitiously or linearly depends on x_i. Then $\mu_{i,j}^{[2]} \leqslant 1$, and the theorem is correct for $t = 2$.

Thus, the theorem is correct for any two transformations on P^n. Suppose, that the theorem is true for $t-1$, where $t > 2$. Let we prove that the theorem is true for t. Denote by h the product $f^{(1)} \cdot \ldots \cdot f^{(t-1)}$. Then $f^{(1)} \cdot \ldots \cdot f^{(t)} = h \cdot f^{(t)}$, and $M_\Theta(f^{(1)} \cdot \ldots \cdot f^{(t)}) = M_\Theta(h \cdot f^{(t)})$. It is proved above that $M_\Theta(h \cdot f^{(t)}) \leqslant M_\Theta(h) \cdot M_\Theta(f^{(t)})$. By the induction hypothesis, $M_\Theta(h) \leqslant M_\Theta(f^{(1)}) \cdot \ldots \cdot M_\Theta(f^{(t-1)})$. Hence,

$$M_\Theta(h \cdot f^{(t)}) \leqslant M_\Theta(f^{(1)}) \cdot \ldots \cdot M_\Theta(f^{(t)}).$$

□

Corollary 2. *If* $M_\Theta(f^{(1)}) \cdot \ldots \cdot M_\Theta(f^{(t)}) \neq (2)_n$, *for* $t \geqslant 1$, *then* $M_\Theta(f^{(1)} \cdot \ldots \cdot f^{(t)}) \neq (2)_n$.

From Corollary 1 and 2, we obtain that the transformation $f^{(1)} \cdot \ldots \cdot f^{(t)}$ is not quite nonlinear, if the multigraph $\Gamma_\Theta(f^{(1)}) \cup \cdots \cup \Gamma_\Theta(f^{(t)})$ is not strongly connected.

4 Generalized primitivity of ternary matrices and corresponding labelled digraphs

For $t \in \mathbb{N}$, the matrix $M \in \mathbb{M}_n$ is called the $\langle 2 \rangle$-*primitive* if $M^t = (2)_n$. The smallest t with this property is called the $\langle 2 \rangle$-*exponent* of the matrix M and denoted by $\langle 2 \rangle \exp M$. For $t \in \mathbb{N}$, $A, B \in \mathbb{M}_n$ such that $A \leqslant B$, it holds from $A^t \leqslant B^t$, that if A is $\langle 2 \rangle$-primitive, then B is $\langle 2 \rangle$-primitive too, and $\langle 2 \rangle \exp A \geqslant \langle 2 \rangle \exp B$; if B is not $\langle 2 \rangle$-primitive, then A is not $\langle 2 \rangle$-primitive too.

Example. Consider the ternary matrix $M = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 2 & 0 & 0 \end{pmatrix}$.

Calculate $\langle 2\rangle \exp M$:

$$M^2 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 2 & 0 & 0 \\ 0 & 1 & 2 & 0 \end{pmatrix}; M^4 = \begin{pmatrix} 1 & 2 & 0 & 0 \\ 0 & 1 & 2 & 0 \\ 0 & 0 & 1 & 2 \\ 2 & 2 & 0 & 1 \end{pmatrix}; \dots M^{12} = \begin{pmatrix} 1 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \end{pmatrix};$$

$M^{13} = (2)_4$. Hence, $\langle 2\rangle \exp M = 13$.

The labelled digraph Γ is called the *complete* $\langle 2\rangle$*-graph* and denoted by $\Gamma_n^{\langle 2\rangle}$, if the corresponding matrix of labels $M(\Gamma) = (2)_n$. For $t \in \mathbb{N}$, the labelled digraph $\Gamma \in \mathbb{\Gamma}_n$ is called $\langle 2\rangle$*-primitive* if $\Gamma^t = \Gamma_n^{\langle 2\rangle}$. The smallest t with this property we call the $\langle 2\rangle$*-exponent* of the labelled digraph Γ and denote by $\langle 2\rangle \exp \Gamma$. Since $\mathbb{\Gamma}_n$ is isomorphic to $\mathbb{M}_n$, we see that the digraph Γ is $\langle 2\rangle$-primitive $\iff$ the matrix $M(\Gamma)$ is $\langle 2\rangle$-primitive, and $\langle 2\rangle \exp \Gamma = \langle 2\rangle \exp M$. Hence, $\langle 2\rangle \exp \Gamma$ is equal to the smallest natural t, such that for any pair of vertices (i, j) in Γ there is the path of length t with the label "2" from i to j.

Denote $U(\Gamma)$ the digraph obtained from Γ by removing the all labels. For $0 \leqslant i, j < n$, $w_{i,j}(l)$ denotes the path of length l from the vertex i to the vertex j; $w_i^{[2]}$ – the shortest path from i to the nearest vertex $\xi(i)$, that is the startpoint of the arc $(\xi(i), s(i))$ with the label "2"; $d^{[2]} = \max\{len\, w_0^{[2]}, \dots, len\, w_{n-1}^{[2]}\}$. The vertices $\xi(i)$ and $s(i)$ are generally ambiguous.

Theorem 3 (The criterion of $\langle 2\rangle$-primitivity)**.** *The labelled digraph* $\Gamma \in \mathbb{\Gamma}_n$ *is* $\langle 2\rangle$*-primitive* $\iff$ Γ *contains the arc with the label "2" and* $U(\Gamma)$ *is primitive, thereby* $\exp U(\Gamma) \leqslant \langle 2\rangle \exp \Gamma \leqslant 1 + d^{[2]} + \exp U(\Gamma)$.

Proof. Necessity. Suppose, the labelled digraph Γ is $\langle 2\rangle$-primitive. For $t \in \mathbb{N}$, Γ contains the path from i to j of length t with the value 2 of the label. Then, Γ contains the arc labelled "2", and for any i, j, $U(\Gamma)$ contains the path of length t from i to j. Hence, the digraph $U(\Gamma)$ is primitive.

Sufficiency. Let $U(\Gamma)$ be primitive, and $t = \exp U(\Gamma)$. Then, for any $i, j = \{0, \dots, n-1\}$, Γ and $U(\Gamma)$ contains the paths from i to j of length $t, t+1, \dots$. Let we construct the path $w_{i,j}$ from i to j, such that:

$$w_{i,j} = w_i^{[2]} \bullet (\xi(i), s(i)) \bullet w_{s(i),j}(l_i),$$

where $l_i = t + d^{[2]} - len\, w_i^{[2]} \geqslant t, i, j \in \{0, \dots, n-1\}$. The path $w_{s(i),j}(l_i)$ exists because $t = \exp U(\Gamma)$. Then $len\, w_{i,j} = d^{[2]} + 1 + t > t$, hence, for any i, j in Γ

there is the path from i to j of length $d^{[2]} + 1 + t$ with the value 2 of the label. Hence, $\langle 2\rangle \exp \Gamma \leqslant 1 + d^{[2]} + t$. The upper bound is proved.

Taking into account the definition of the primitive digraph, if $\tau < \exp U(\Gamma)$, then for some $i, j \in \{0, \ldots, n-1\}$, $U(\Gamma)$ and Γ have no path of length τ from i to j. So, the lower bound is correct. □

Corollary 3. *If the labelled digraph Γ is $\langle 2\rangle$-primitive, then*

$$\langle 2\rangle \exp \Gamma \leqslant n + \exp U(\Gamma) \leqslant n^2 - n + 2.$$

Proof. The left inequality follows from the Theorem 3, because $d^{[2]} \leqslant n - 1$ in the $\langle 2\rangle$-primitive digraph. The right inequality is correct due to the universal Wielandt bound [3]. □

Theorem 4. *1. If $\langle 2\rangle$-primitive digraph Γ contains the circuit C of length $l > 1$, then*

$$\langle 2\rangle \exp \Gamma \leqslant d^{[2]} + 1 + n + l(n-2). \tag{8}$$

2. If the circuit C of length l passes through the arc with the label "2", then

$$\langle 2\rangle \exp \Gamma \leqslant n + l(n-1). \tag{9}$$

Proof. 1. Suppose that the digraph Γ^l is $\langle 2\rangle$-primitive and contains at least l loops. Then, Γ^l contains the path $w_{z,j}$ of length no more than $n-1$ from for any vertex z with a loop to any vertex j. Hence, Γ contains the path $u_{z,j}$ of length $l(n-1)$ from any vertex z of circuit C to any vertex j.

Denote $v_{i,z}$ the shortest path in Γ from i to the nearest vertex z of the circuit C. We see that $len\, v_{i,z} \leqslant n - l$, so Γ contains the path
$u_{i,z} = w_i^{[2]} \bullet (\xi(i), s(i)) \bullet w_{s(i),z}$ of length no more than $d^{[2]} + 1 + n - l$;
$u_{i,z}$ passes through the arc with the label "2". Therefore, for any vertices i and j the path $u_{i,z} \bullet u_{z,j}$ passes through the arc labelled "2" and has the length at most $d^{[2]} + 1 + n + l(n-2)$. Hence, the bound (8) is correct.

2. Suppose that the circuit C passes through the arc with the label "2". Let us attach the loop $\pi(z)$ to the beginning of the path $w_{z,i}$ in Γ^l. We get that Γ^l contains the path $\pi(z) \bullet w_{z,j}$ of length no more than n with the value 2 of the label. Then, Γ contains the path $u_{z,j}$ of length no more than ln labelled "2". Then, for any vertices i, j in Γ there is a path $v_{i,z} \bullet u_{z,j}$ of length $n + l(n-1)$ from i to j with the label "2". Hence, the bound (9) is correct. □

Corollary 4. *If $d^{[2]} \geqslant l$, the bound (8) is greater than (9).*

Example. Suppose that the labelled digraph Γ contains the Hamiltonian circuit $(0,\dots,n-1)$ and the circuit $(0,\dots,l-1)$, $3<l<n$. If the label "2" belongs to the only arc $(n-1,0)$, then $d^{[2]}=len\, w_0^{[2]}=n-1\geqslant l$, and the bound (8) is greater than the bound (9). If the label "2" belongs to the all arcs $(i,i+1)$, where i is odd and $i\leqslant 2$, then $d^{[2]}=len\, w_{n-1}^{[2]}\leqslant 2<l$, and the bound (8) is lower than the bound (9).

Denote $\pi^k(z)$ the loop in the vertex z passing k times, $k\geqslant 0, 0\leqslant z<n$.

Theorem 5. *1. If the $\langle 2\rangle$-primitive digraph Γ contains $p>0$ loops, then*

$$\langle 2\rangle \exp\Gamma\leqslant d^{[2]}+2n-p.$$

2. If the $\langle 2\rangle$-primitive digraph Γ contains $m>0$ loops with the label "2", then

$$\langle 2\rangle \exp\Gamma\leqslant 2n-m.$$

Proof. 1. Denote $w_{s(i),z}$ the shortest path of length τ from $s(i)$ to the nearest vertex z with a loop; $w_{z,j}$ – the shortest path of length θ from z to j. For $p>0$ and $i,j\in\{0,\dots,n-1\}$ construct the path $w_{i,j}$ passing through the arc with the label "2" and through the vertex z with a loop: $w_{i,j}=w_i^{[2]}\bullet(\xi(i),s(i))\bullet w_{s(i),z}\bullet\pi^k(z)\bullet w_{z,j}$, where $k\geqslant 0$. Then $\tau\leqslant n-p$, $\theta\leqslant n-1$, and $len\, w_{i,j}\leqslant d^{[2]}+2n-p+k, p>0$. Since i,j are arbitrary, and $k\geqslant 0$, then $\langle 2\rangle\exp\Gamma\leqslant d^{[2]}+2n-p$.

2. Denote by $w_{i,z}$ the path of length no more than $n-m$ from the vertex i to the nearest vertex z with the loop and label "2"; $w_{z,j}$– the path of length no more than $n-1$ from z to j (if $z=j$ then the path $w_{z,j}$ is empty). For $m>0$ and $i,j\in\{0,\dots,n-1\}$ construct the path $w_{i,j}$ passing through the loop with the label "2": $w_{i,j}=w_{i,z}\bullet\pi^k(z)\bullet w_{z,j}$. If $k>0$, then the path $w_{i,j}$ passes through the loop with the label "2", and $len\, w_{i,j}\leqslant 2n-m-1+k$. Since i,j are arbitrary, and $k>0$, than $\langle 2\rangle\exp\Gamma\leqslant 2n-m$. □

5 Applications

The proposed approach is applied to the estimation of $\langle 2\rangle$-exponents of the ternary matrices, which constructed for the round transformations of block encryption algorithms DES and GOST 28147-89 (the Diploma thesis at the Department of Cryptology and Cyber Security at National Research Nuclear

University "MEPhI", 2019). The obtained values coincided with the values of the exponents of mixing matrices for the round transformations.

References

[1] Fomichev V. M., Avezova Ya. A., Koreneva A. M., Kyazhin S. N., "Primitivity and Local Primitivity of Digraphs and Nonnegative Matrices", *Journal of Applied and Industrial Mathematics*, **12**:3 (2018), 453–469, DOI: 10.1134/S1990478917010045.

[2] Fomichev V. M., Melnikov D. A., *Kriptograficheskie metody zaschity informatsii [Cryptographic methods of information security]*, ed. Fomichev V. M., URAIT, Moscow, 2016 (Russian), 454 pp.

[3] Wielandt H., "Unzerlegbare, nicht negative Matrizen", *Mathematische Zeitschrift*, **52**:1 (1950), 642–648.

Quantum and Postquantum

Limonnitsa: Making Limonnik-3 Post-Quantum

Sergey Grebnev

Technical Committee for Standardization
«Cryptography and security mechanisms» (TC 026), Russia
grebnev_sv@tc26.ru

Abstract

We propose *Limonnitsa*, a quantum secure authenticated key exchange (AKE) scheme which brings together the standardized *Limonnik-3* AKE scheme and the supersingular elliptic curves isogeny cryptographic framework. We discuss Limonnitsa's basic cryptographic properties and preliminary choice of its basic parameters that conforms with another standardized cryptographic primitives.

Keywords: authenticated key exchange, isogenies, Limonnik-3, post-quantum cryptography, supersingular elliptic curves.

1 Introduction

An emerging threat of quantum computers leads cryptographers to review many of existing public key cryptographic systems. For example, cryptanalysis of the schemes based upon the factorization problem, such as RSA and Rabin, as well as discrete logarithm based schemes, including Diffie-Hellman-Merkle and ElGamal, is reduced to a polynomial-time quantum algorithm.

Thus, although the prospectives of the construction of a powerful enough (from a cryptanalyst's point of view) quantum computer are unclear, many researchers are concerned about creating "post-quantum" schemes which are to withstood both "classical" (that is, based upon the Turing-style computations) and "quantum" cryptanalysis. We mention the NIST proposal for the post-quantum family of cryptosystems which has brought anomalous amount of research into the post-quantum field.

In 2017, Russia officially accepted a family of AKE protocols designed by the author as recommendations for standardization (that is, a candidate to become a national standard). This family includes Echinacea-2, Echinacea-3 and Limonnik-3 protocols[1].

[1] *Эхинацея (Echinacea purpurea)* and *Лимонник (Schizandra chinensis)* are medicinal herbs extensively used in Russian complementary medicine.

Both the protocols are based upon the elliptic curve Diffie-Hellman scheme and are thus quantum-insecure. The Echinacea-3 protocol is built from the ISO-STS-MAC [9], using KEA+C [18] ideas. The Limonnik-3 protocol is built from the MTI/A0 protocol [19] with influences by [6] and [20].

Unlike Echinacea, Limonnik-3 does not require digital signatures, it may be viewed as the outputs of two elliptic curve Diffie-Hellman processes, each one mixing a static and an ephemeral key, hashed together to build a shared secret key. Thus, we choose this scheme for post-quantum conversion, replacing the Diffie-Hellman protocol by its post-quantum analogue.

Amongst the multiple post-quantum proposals, we have chosen SIDH, the supersingular elliptic curves isogeny-based Diffie-Hellman key exchange protocol [7] for the following reasons.

– Unlike most NIST competitors, the protocol allows for static keys (see, however, [12, 17] for discussion of several attacks against static keys), which are mandatory for an AKE scheme;

– the protocol, for a given security parameter, provides keys of moderate size;

– the protocol may be implemented quite efficiently with a well-studied mechanisms.

We proceed with a general description of the Limonnik-3 and basic ideas of supersingular elliptic curves cryptography.

2 Limonnik-3

We choose protocol parameters h_2, h_3 as two fixed distinct non-empty strings. The function $\pi : E(GF(p)) \to V^*$ represents the point's x-coordinate as a binary string, $\texttt{KDF}(\dots)$ is a key derivation function, for example, the one specified by [2]. $\texttt{MAC}_K(\dots)$ is a message authentication code defined in [4], $\texttt{enc}_K(\dots)$ is the «Kuznyechik» encryption [3] using the key K.

An optionally used information connected to the session (timestamps, IP addressess, previously shared secret strings etc) which may be used during key generation is denoted $\texttt{OI}$. Concatenation of strings a, b is denoted by $a \parallel b$.

A party A's identity is denoted by ID_A. We suppose that the communicating parties A and B are using two (possibly different) elliptic curves $E_A(GF(p_A)$ and $E_B(GF(p_B))$ defined over corresponding prime fields $GF(p_A), GF(p_B)$.

A party's curve has the following parameters important for the description:

– $m_A = |E_A|$;

– P_A is a point of large prime order q_A, $q_A | m_A$;

– $c_A = m_A/q_A$ is the cofactor.

Static key pairs (s_A, S_A) and (s_B, S_B) are defined as $S_A = s_A P_A$, $S_B = s_B P_B$, where $0 < s_A < q_A$, $0 < s_B < q_B$, and certified by $\mathtt{Cert}_A, \mathtt{Cert}_B$.

Limonnik-3

$A:$	$k_A \in_R [1, q_B - 1]$
$A \to B$	$\mathtt{ID}_A, \mathtt{Cert}_A, k_A P_B$
$B:$	$k_B \in_R [1, q_A - 1], Q = c_A k_B S_A, R = c_B s_B k_A P_B$
	$K \parallel M = \mathtt{KDF}(\pi(Q), \pi(R), \mathtt{ID}_A \parallel \mathtt{ID}_B [\parallel \mathtt{OI}])$
	$tag_B = \mathtt{MAC}_M(h_2, k_B P_A, k_A P_B, \mathtt{ID}_B, \mathtt{ID}_A)$
$B \to A$	$\mathtt{ID}_B, \mathtt{Cert}_B, k_B P_A, tag_B$
$A:$	$Q = c_A s_A k_B P_A, R = c_B k_A S_B$
	$K \parallel M = \mathtt{KDF}(\pi(Q), \pi(R), \mathtt{ID}_A \parallel \mathtt{ID}_B [\parallel \mathtt{OI}])$
	If $tag_B \neq \mathtt{MAC}_M(h_2, k_B P_A, k_A P_B, \mathtt{ID}_B, \mathtt{ID}_A)$, terminates the session with an error
	$tag_A = \mathtt{MAC}_M(h_3, k_A P_B, k_B P_A, \mathtt{ID}_A, \mathtt{ID}_B)$
$A \to B$	tag_A
$B:$	If $tag_A \neq \mathtt{MAC}_M(h_3, k_A P_B, k_B P_A, \mathtt{ID}_A, \mathtt{ID}_B)$, terminates the session with an error

We also assume that any party verifies validity of certificate received and correctness of elliptic curve points, terminating the session with an error if an invalid certificate or a "bad" point (i.e. not belonging to the given elliptic curve or having a small order) is provided by another party.

If the scheme succcessfully completes, the parties A and B are mutually authenticated and provided with an implicitly verified shared secret key $K = K_{AB} = K_{BA}$. Note that the key M is used only for the purposes of key confirmation and must be destroyed after the session is established, see [14].

2.1 Isogenies and cryptography in brief

Consider an elliptic curve $E(\mathcal{F})$ defined over a field $\mathcal{F}$, $\operatorname{char}\mathcal{F} \neq 2, 3$, $E_{a,b}(\mathcal{F}) : y^2 = x^3 + ax + b$.

Definition 1. *Let $E_{a,b}$, E_{a_1,b_1} – elliptic curves over K. A* rational map *$E_{a,b}$ to E_{a_1,b_1} – is a map*

$$\psi = \psi(x, y) = (f_1(x, y), f_2(x, y)),$$

where $f_1(x, y), f_2(x, y) \in \overline{\mathcal{F}}(E_{a,b})$, such that for any point $(x_0, y_0) \in E_{a,b}$ where the functions are defined, implies that $(f_1(x_0, y_0), f_2(x_0, y_0)) \in E_{a_1,b_1}(\mathcal{F})$.

Definition 2. *A rational map defined in every point of $E_{a,b}(\mathcal{F})$, is a* morphism.

Definition 3. *If ψ – is a morphism and $\psi(\mathcal{O}) = \mathcal{O}_1$, then ψ is an* isogeny. *If such a map exists, the curves are* isogenous.

Definition 4. *For any isogeny $\psi : E \to E'$ there exists an unique* dual *isogeny $\hat{\psi} : E' \to E$ such that $\hat{\psi} \circ \psi = [m]_E$ and $\psi \circ \hat{\psi} = [m]_{E'}$, where m is the* degree *of an isogeny ψ.*

Definition 5. *Considering three elliptic curves $E(\mathcal{F}), E'(\mathcal{F}), E''(\mathcal{F})$ and isogenies ϕ, ψ: $\phi : E \mapsto E'$, $\psi : E' \mapsto E''$, we define* composition *of isogenies $\psi\phi : E \mapsto E''$.*

We have that $\hat{\psi\phi} = \hat{\phi}\hat{\psi}$ and $\deg \psi\phi = \deg \psi \deg \phi$.
Let now $\operatorname{char} \mathcal{F} = p$.

Definition 6. *If $E[p^e] = \{\mathcal{O}\}$ for any $e = 1, 2, \ldots$, the curve E is* supersingular.

There are about $\lfloor p/12 \rfloor$ distinct supersingular curves defined over $GF(p^2)$, see [7] – that is quite enough for cryptographic applications.

2.2 Computation of isogenies

One can use Vélu's formulae [23] to compute isogenies φ with a given kernel (i.e. a subgroup $G \subset E$), $\varphi : E \mapsto E' = E/G$. Given curve coefficients a, b for E, and all of the x-coordinates x_i of the subgroup $G \subset E$, Vélu's formulae output a', b' for E', and the map

$$\varphi : E \to E' = E/G,$$

$$(x, y) \mapsto \left(\frac{f_1(x, y)}{g_1(x, y)}, \frac{f_2(x, y)}{g_2(x, y)} \right).$$

The complexity of computation of isogeny of degree l is $O(l)$ field operations. For isogenies of smooth degrees, however, the complexity may be lowered by decomposing it into a composition of isogenies of small degrees.

We recall that isomorphic curves have the same j-invariant. Since construction of an isomorphism is a simple task, the isogeny problem is actually the problem of finding isogenies between classes of isomorphic curves, every one of which is represented by its j-invariant.

3 Supersingular Isogeny Diffie-Hellman

We proceed with the description of the Supesingular Isogeny Diffie-Hellman scheme (SIDH), following [7].

We fix the public parameters: $p = l_A^{e_A} l_B^{e_B} \cdot f \pm 1$, where l_A, l_B are distinct small prime numbers (e.g., $l_A = 2$ and $l_B = 3$), $(l_A, f) = (l_B, f) = 1$, a supersingular elliptic curve $E_0(GF(p^2))$ and bases $\{P_A, Q_A\}$ and $\{P_B, Q_B\}$, which generate correspondingly $E_0[l_A^{e_A}]$ and $E_0[l_B^{e_B}]$, i.e. $\langle P_A, Q_A \rangle = E_0[l_A^{e_A}]$ and $\langle P_B, Q_B \rangle = E_0[l_B^{e_B}]$.

The party A chooses two random elements $m_A, n_A \in_R \mathbb{Z}/l_A^{e_A}\mathbb{Z}$, not both divisible by l_A, and constructs an isogeny $\varphi_A : E_0 \to E_A$ with the kernel $\mathcal{K}_A := \langle [m_A]P_A + [n_A]Q_A \rangle$. The party A also computes the image $\{\varphi_A(P_B), \varphi_A(Q_B)\}$ and sends these points to the party B together with E_A.

Simultaneously, the party B chooses two random elements $m_B, n_B \in_R \mathbb{Z}/l_B^{e_B}\mathbb{Z}$, not both divisible by l_B, and constructs an isogeny $\varphi_B : E_0 \to E_B$ with the kernel $\mathcal{K}_B := \langle [m_B]P_B + [n_B]Q_B \rangle$. The party B also computes the image $\{\varphi_B(P_B), \varphi_B(Q_B)\}$ and sends these points to the party A.

Having received the party B's set $E_B, \varphi_B(P_B), \varphi_B(Q_B)$, the party A constructs an isogeny $\varphi'_A : E_B \to E_{AB}$ with the kernel $\langle [m_A]\varphi_B(P_A) + [n_A]\varphi_B(Q_A) \rangle$; the party B operates in a similar way.

The shared key may be computed as the j-invariant of the curve

$$E_{AB} = \varphi'_B(\varphi_A(E_0)) = \varphi'_A(\varphi_B(E_0)) = \\ = E_0 / \langle [m_A]P_A + [n_A]Q_A, [m_B]P_B + [n_B]Q_B \rangle .$$

We have a following commutative diagram:

$$\begin{array}{ccc} E & \xrightarrow{\varphi} & E/\langle P\rangle \\ {\scriptstyle\psi}\downarrow & & \downarrow \\ E/\langle Q\rangle & \longrightarrow & E/\langle P,Q\rangle \end{array} \tag{1}$$

where φ, ψ are random walks in the graphs of isogenies of degrees equal to powers of l_A, l_B.

The protocol implements an analogue of the Diffie-Hellman scheme over this commutative diagram, where the party A chooses φ, and B chooses ψ.

4 Putting things together: Limonnitsa

In this section we describe an AKE scheme that is derived from Limonnik-3 by merging into it the ideas of supersingular elliptic curves isogenies crypto. The new protocol is named *Limonnitsa*[2].

So, we fix two (possibly distinct) sets of the public parameters for the parties:

- $p_A = 2^{e_{a2}}3^{e_{a3}} - 1$,
- $E_{A0}(GF(p_A^2)) : y^2 = x^3 + x$,
- linearly independent points $P_{A2}, Q_{A2} \in E_{A0}[2^{e_{a2}}]$ (that is, $|\langle P_{A2}, Q_{A2}\rangle| = 2^{2e_{a2}}$) and linearly independent points $P_{A3}, Q_{A3} \in E_{A0}[3^{e_{a3}}]$ (that is, $|\langle P_{A3}, Q_{A3}\rangle| = 3^{2e_{a2}}$)

For the party B, we have:

- $p_B = 2^{e_{b2}}3^{e_{b3}} - 1$,
- $E_{B0}(GF(p_B^2)) : y^2 = x^3 + x$,
- linearly independent points $P_{B2}, Q_{B2} \in E_{B0}[2^{e_{b2}}]$ (that is, $|\langle P_{B2}, Q_{B2}\rangle| = 2^{2e_{b2}}$) and linearly independent points $P_{B3}, Q_{B3} \in E_{B0}[3^{e_{a3}}]$ (that is, $|\langle P_{B3}, Q_{B3}\rangle| = 3^{2e_{a2}}$)

Now, the party A selects its secret static key as an integer s_A such that $0 < s_A < 2^{e_{a2}}$, constructs the isogeny $\varphi_A : E_A \to E_A/\langle P_{A2} + [s_A]Q_{A2}\rangle$, calculates

[2] *Лимонница (limonnitsa)* stands for a brimstone butterfly (*Gonepteryx rhamni*) in Russian.

$E_A = E_{A0}/\langle P_{A2} + [s_A]Q_{A2}\rangle$, $P_A = \varphi_A(P_{A3})$, $Q_A = \varphi_A(Q_{A3})$, sets its static public key to $\{E_A, P_A, Q_A\}$, and acquires a certificate $\texttt{Cert}_A$.

B selects its static key as an integer (s_B such that $0 < s_B < 2^{e_{b2}}$, constructs the isogeny $\varphi_B : E_B \to E_B/\langle P_{B2} + [s_B]Q_{B2}\rangle$, calculates $E_B = E_{B0}/\langle P_{B2} + [s_B]Q_{B2}\rangle$, $P_B = \varphi_B(P_{B3})$, $Q_B = \varphi_B(Q_{B3})$, sets is static public key as $\{E_B, P_B, Q_B\}$, and acquires a certificate $\texttt{Cert}_B$ as well.

We shall use SIKE [15] modification to the original scheme and generate kernels of the isogenies for public key calculation in the form $\langle P + [k]Q\rangle$, that is, we generate a single random value. As shown by Galbraith [11, 12], the corresponding computational problems are equivalent.

Limonnitsa

$A:$	$k_A \in_R [1, 3^{e_{b3}}], S_{AB} = P_{B3} + [k_A]Q_{B3}$,
	$\varphi_{AB} : E_B \to E_B/\langle S_{AB}\rangle$ – an isogeny with the kernel $\langle S_{AB}\rangle$
	$E_{AB} = E_{B0}/\langle S_{AB}\rangle$ (that is, $E_{AB} = \varphi_{AB}(E_{B0})$)
	$\mathcal{K}_A = \{E'_A, \varphi_{AB}(P_{B2}), \varphi_{AB}(Q_{B2})\}$ – A's ephemeral public key
$A \to B$	$\texttt{ID}_A, \texttt{Cert}_A, \mathcal{K}_A$
$B:$	$k_B \in_R [1, 3^{e_{a3}}], S_{BA} = P_{A3} + [k_B]Q_{A3}$,
	$\varphi_{BA} : E_A \to E'_B/\langle S_{BA}\rangle$ – an isogeny with the kernel $\langle S_{BA}\rangle$
	$E_{BA} = E_{A0}/\langle S_{BA}\rangle$ (that is, $E_{BA} = \varphi_{BA}(E_{A0})$)
	$\mathcal{K}_B = \{E'_B, \varphi_{BA}(P_{A2}), \varphi_{BA}(Q_{A2})\}$ – B's session public key
	$T_{AB} = P_A + [k_B]Q_A$
	$T'_{AB} = \varphi_{AB}(P_{B2}) + [s_B]\varphi_{AB}(Q_{B2})$
	$\psi_{AB} : E'_A \to E'_A/\langle T_{AB}\rangle$ – an isogeny with the kernel $\langle T_{AB}\rangle$
	$\psi'_{AB} : E_B \to E_B/\langle T'_{AB}\rangle$ – an isogeny with the kernel $\langle T'_{AB}\rangle$
	$E_{AB} = \psi_{AB}(E'_A); E'_{AB} = \psi'_{AB}(E_B)$
	$K \parallel M = \texttt{KDF}(j(E_{AB}) \parallel j(E'_{AB}) \parallel \texttt{ID}_A \parallel \texttt{ID}_B[\parallel \texttt{OI}])$
	$tag_B = \texttt{MAC}_M(h_2, \mathcal{K}_B, \mathcal{K}_A, \texttt{ID}_B, \texttt{ID}_A)$
$B \to A$	$\texttt{ID}_B, \texttt{Cert}_B, \mathcal{K}_B, tag_B$
$A:$	$T_{BA} = \varphi_{BA}(P_{A2}) + [s_A]\varphi_{BA}(Q_{A2})$
	$T'_{BA} = P_B + [k_A]Q_B$
	$\psi'_{BA} : E'_B \to E'_B/\langle T_{BA}\rangle$ – an isogeny with the kernel $\langle T_{BA}\rangle$
	$\psi_{BA} : E_A \to E_A/\langle T'_{BA}\rangle$ – an isogeny with the kernel $\langle T'_{BA}\rangle$
	$E'_{BA} = \psi'_{BA}(E'_B); E_{BA} = \psi'_{AB}(E_A)$
	$K \parallel M = \texttt{KDF}(j(E'_{BA}) \parallel j(E_{BA}) \parallel \texttt{ID}_A \parallel \texttt{ID}_B[\parallel \texttt{OI}])$
	If $tag_B \neq \texttt{MAC}_M(h_2, \mathcal{K}_B, \mathcal{K}_A, \texttt{ID}_B, \texttt{ID}_A)$,
	terminates the session with an error
	$tag_A = \texttt{MAC}_M(h_3, \mathcal{K}_A, \mathcal{K}_B, \texttt{ID}_A, \texttt{ID}_B)$
$A \to B$	tag_A
$B:$	If $tag_A \neq \texttt{MAC}_M(h_3, \mathcal{K}_A, \mathcal{K}_B, \texttt{ID}_A, \texttt{ID}_B)$,
	terminates the session with an error

Our protocol reminds Galbraith's variant of the NAXOS protocol from [10];

however, in our setting (provided that the parties' parameters may differ) the ephemeral-to-ephemeral shared key which is employed in NAXOS key generation cannot be produced.

The protocol is a combination of static-to-ephemeral sessions, and thus may be subject to an attack against static keys by [17]. In order to thwart the attack, a party must ensure that the public keys it receives are valid, i.e. elliptic curves are built as prescribed by the protocols, the generators are chosen at random and are of prescribed order and linearly independent. Several validation techniques are described in [12]. The paper [22] states that the key validation problem may be equivalent to the CSSI problem (see below); thus, we would rather use the following variant of a trick from [16].

Instead of choosing random ephemeral secret key k_A, the party A chooses a single random seed $r_A \in V^*$ and uses a pseudo-random function $\texttt{prf}$ to output $k_A = \texttt{prf}(r_A)$. Then, tag_A is calculated as

$$tag_A = \texttt{enc}_M(h_2, r_A, K_A, K_B, \texttt{ID}_A, \texttt{ID}_B).$$

The party B, having calculated the session key, recovers the seed r_A and repeats A's computations in order to verify that the keys were constructed as prescribed, otherwise, terminates the session. The parties B and A proceed vice versa.

Note that in this setting a party's ephemeral secret key is uncovered to another party, thus, it becomes the party's responsibility to generate unique value each time. Now many practical issues arise (for example, storing and searching through a database of any previously generated values in a secure manner may be too expensive). We propose to use a secure PRNG instead.

5 Analysis

The 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* $[m_1]R_1 + [n_1]S_1$*, where* m_1*,* n_1 *are chosen uniformly at random from the interval* $[1, l_1^{e_1}]$*, and are not both divisible by* l*. Given* E_1 *and images* $\phi_1(R_2), \phi_1(S_2)$*, find the generator of* $\langle [m_1]R_1 + [n_1]S_1 \rangle$*.*

Note that we choose the j-invariants of the elliptic curves resulting from two SIDH processes with different parameters and hash them together with a $\texttt{KDF}$ function to obtain a shared secret value. Then, a very naive deduction implies

that an adversary has to solve two distinct instances of CSSI, which should be twice as hard.

It is believed that the best classical algorithm to attack the problem has the complexity is $O(\sqrt[2]{l^{e_l}})$, where $l^{e_l} = \min(l_1^{e_1}, l_2^{e_2})$, while the claw-finding quantum algorithm [21] has the complexity $O(\sqrt[3]{l^{e_l}})$. Recent research [5, 16] show that the actual quantum complexity of breaking the isogeny problem is estimated by $O(\sqrt[4]{p})$ operations, but we choose to be a little on a safe side. We discuss the choice of parameters in the following sections.

Note that the protocol inherits implicit key confirmation, KCI- and UKS-immunity from Limonnik-3 [14]. The protocol provides forward security against A,B (but not A AND B, since if long-term keys of the both parties are compromised, all the sessions involving them both are compromised, too; this property arises from the basic structure of the MTI/A0 protocol [19].

6 Security arguments

Consider the following computational problem [7, 22].

Problem 2. Computational isogeny Diffie-Hellman, SSCDH*: let* $\varphi_A : E_0 \to E_A$ *– an isogeny with kernel* $\langle P_A + [n_A]Q_A\rangle$*, and* $\varphi_B : E_0 \to E_B$ *– an isogeny with kernel* $\langle P_B + [n_B]Q_B\rangle$*, where* n_A *is chosen uniformly randomly from* $\mathbb{Z}/l_A^{e_A}\mathbb{Z}$ *and* n_B *is chosen uniformly randomly from* $\mathbb{Z}/l_B^{e_B}\mathbb{Z}$*. Given* E_A, E_B *and the images* $\varphi_A(P_B), \varphi_A(Q_B), \varphi_B(P_A), \varphi_B(Q_A)$*, find the* j*-invariant of the curve* $E_0/\langle P_A + [n_A]Q_A, P_B + [n_B]Q_B\rangle$*.*

The decisional version of the problem may be stated as follows.

Problem 3. Decisional isogeny Diffie-Hellman, SSDDH*: Given a tuple sampled with probability* $1/2$ *from one of the following two distributions*

– $(E_A, E_B, \varphi_A(P_B), \varphi_A(Q_B), \varphi_B(P_A), \varphi_B(Q_A), E_{AB})$*, where* $(E_A, E_B, \varphi_A(P_B), \varphi_A(Q_B), \varphi_B(P_A), \varphi_B(Q_A)$ *– as before,*

$$E_{AB} \cong E_0/\langle P_A + Q_A, [m]P_B + [n]Q_B\rangle;$$

– $(E_A, E_B, \varphi_A(P_B), \varphi_A(Q_B), \varphi_B(P_A), \varphi_B(Q_A), E_C)$*, where* $(E_A, E_B, \varphi_A(P_B), \varphi_A(Q_B), \varphi_B(P_A), \varphi_B(Q_A)$ *– as before, and*

$$E_C \cong E_0/\langle P_A + [n']Q_A, P_B + [n']Q_B\rangle$$

where m', n' are chosen at random from from $\mathbb{Z}/l_B^{e_B}\mathbb{Z}$);

determine from which distribution the tuple is sampled.

We cannot state an analogue of the Diffie-Hellman problem (GDHP) for the supersingular isogeny case, since decisional problems here are equivalent to computational. Thus, security arguments for Limonnik-3, proven secure under the GDHP hardness assumption, cannot be directly transformed for Limonnitsa. However, as pointed out by Galbraith, we may consider a weaker adversary model.

We state now a weaker version of the security definition adapted from [6]. We allow an adversary M to perform any of the following queries.

- *Initiate* a session between any chosen parties.
- *Send* messages from a party to another, which is followed by a correct (prescribed by the protocol) response.
- *Execute* a correct session between any chosen parties.
- *Corrupt* a party (that is, to learn any secret keys, as well as all generated shared keys and any local state information).

Note that M cannot perform any Reveal queries.

Define as $\Lambda(n)$ the set of all Limonnitsa public parameters for a chosen security parameter n: that is, all primes of an appropriate form with bit-lentgh n, all possible supersingular elliptic curves defined over the corresponding primes.

Definition 7. *A key agreement protocol is said to be* weak-AKE-secure *if the following conditions hold:*

1. *If two honest parties complete matching sessions then, except with negligible probability, they both compute the same session key.*
2. *No polynomially bounded adversary M defined above can distinguish the session key of a fresh session from a randomly chosen session key with probability greater than $1/2$ plus a negligible fraction.*

Then the following theorem holds.

Theorem 1. *Let the SSDDH problem for Λ be computationally hard. Let* `KDF` *be modelled by a pseudorandom function, let* `MAC` *be secure against forgery attack. Then* Limonnitsa *is secure in the sense of Definition 7.*

Proof. The proof repeats the analogous result for Limonnik-3 [14] in a weaker security model. □

7 Choice of parameters

Consider primes of the form $p_A = 2^{e_2}3^{e_3} - 1$. In order to keep up the classic and quantum (see [8]) complexity with the standardized block cipher *Kuznyechik* [3], which has 128-bit block size and 256-bit keys, we choose the parameter p as the smallest prime of the form $p_A = 2^{e_2}3^{e_3} - 1$ such that $\log_2 p/6 \geqslant 128$ and the factors $2, 3$ are balanced: $e_2 \approx e_3/\log_2 3$. Thus we obtain the *Limonnitsa-prime* $p_\lambda = 2^{451}3^{284} - 1$; $\log_2 p_\lambda \approx 902$.

Note that SIKE NIST proposal [15], following NIST requirements and estimations of the quantum security of AES, provides a 964-bit prime for the same classical security level. Limonnitsa allows for distinct parameters of the parties; this means that, for example, a party with a SIKE public parameters may run a Limonnitsa session with a Limonnitsa-prime-based party. The only practical problem here may be mutual public parameter verification.

Elliptic curve operations may be implemented by various techniques; for example, Montgomery or Edwards forms of an elliptic curve may be used.

The protocol execution takes (almost) exactly twice the complexity of executing a SIDH protocol with analogous choice of parameters. Its feasibility for embedded systems may be a subject of discussion as well as that of SIDH/SIKE.

8 Conclusion

We have proposed a post-quantum variant of the officially adopted key exchange protocol. We have studied its basic cryptographic properties. We have shown that the protocol is both classical and quantum-secure and conforms to the cryptographic requirements.

However, implementation and efficiency issues of *Limonnitsa*, including parameters optimization for specific processors, as well as side-channel attack protection, are yet to investigate.

9 Acknowledgements

The author would like to thank anonymous reviewers for valuable comments which helped to improve the protocol.

References

[1] *R 1323565.1.004-2017. Standardization recommendations. Key agreement schemes based upon public-key methods*, Standartinform, Moscow, 2017, in Russian.

[2] *R 1323565.1.022-2018. Standardization recommendations. Key derivation functions*, Standartinform, Moscow, 2018, in Russian.

[3] *GOST R 34.12-2015. National standard of Russian Federation. Block ciphers*, Standartinform, Moscow, 2015.

[4] *GOST R 34.13-2015. National standard of Russian Federation. Block cipher modes*, Standartinform, Moscow, 2015, in Russian.

[5] Biasse J.-F., Jao D., Sankar A., "A quantum algorithm for computing isogenies between supersingular elliptic curves", *LNCS*, **8885**, 2014, 428–442.

[6] Chatterjee S., Menezes A., Ustaoglu B., "A generic variant of NIST's KAS2 key agreement scheme", *Proc. ACISP*, 2011, 353–370.

[7] De Feo L., Jao D., Plût J., "Towards Quantum-Resistant Cryptosystems From Supersingular Elliptic Curve Isogenies", *J. Mathematical Cryptology*, **8(3)** (2014), 209–247.

[8] Denisenko D., Marshalko G., Nikitenkova M., Rudskoy V., Shishkin V., "Estimation of Grover's algorithm implementation for searching GOST R 34.10-2015 block cipher keys", *Jour. Exp. Theor. Phys., to appear*, 2019.

[9] Diffie W., van Oorschot P., Wiener M., "Authentication and authenticated key exchanges", *Designs.*, **2**, 107–125.

[10] Galbraith S., "Authenticated key exchange for SIDH", *Cryptology ePrint Archive*, **2018/266**, 2018.

[11] Galbraith S., Petit P., Silva J., "Schemes Based On Supersingular Isogeny Problems", *Cryptology ePrint Archive*, **2016/1154**, 2016.

[12] Galbraith S., Petit P., Shani B., Yan Bo Ti, "On the Security of Supersingular Isogeny Cryptosystem", *Cryptology ePrint Archive*, **2016/859**, 2016.

[13] Grebnev S., "Security properties of certain authenticated key exchange protocols", Proc. CTCrypt'2014, 2014.

[14] Grebnev S., "Security properties of Limonnik-3", *Bezopasnost' Informacionnykh Tekhnologii*, **26**:2 (2019), (to appear), in Russian.

[15] 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.

[16] Jaques S., Schanck J.M., "Quantum cryptanalysis in the RAM model: Claw-finding attacks on SIKE", **2019/103** (2019).

[17] Kirkwood D., Lackey B.C., McVey J., Motley M., Solinas J.A., Tuller D., "Failure is not an option: Standardization issues for post-quantum key agreement", *Workshop on Cybersecurity in a Post-Quantum World*, 2015.

[18] Lauter K., Mityagin A., "Security analysis of KEA authenticated key exchange protocol", *LNCS*, PKC 2006, **3958**, 2006., 378–394.

[19] Matsumoto T., Takashima Y., Imai H., "On seeking smart public-key distribution systems", *Trans. IECE of Japan*, **E69(2)** (1986), 99–106.

[20] Matyukhin D., "On some properties of PKI-based key agreement schemes in the context of developing standardized solutions", *Review of Appllied and Industrial Mathematics*, **18** (2011), 793–794, in Russian.

[21] Seiichiro T., "Claw Finding Algorithms Using Quantum Walk", 2008, http://arxiv.org/abs/0708.2584.

[22] Urbanik D., Jao D., "SoK: The Problem Landscape of SIDH", *Cryptology ePrint Archive*, **2018/336**, 2018.

[23] Vélu J., "Isogenies entre courbes elliptiques", *C.R. Acad. Sc. Paris*, **Serie A., 273** (1971), 238–241.

Key distribution. Episode 1: Quantum menace

Grigory Marshalko and Vladimir Rudskoy

Technical Committee for Standardization
«Cryptography and security mechanisms» (TC 026), Russia
{marshalko_gb, rudskoy_vi}@tc26.ru

Abstract

We study the possibility of applying related key attacks on cryptographic devices which use quantum key distribution (QKD), in case on compromise of «quantum» part. We consider the simplest way of XORing quantum key and long-term key.

We review several known attacks on QKD systems in order to assess the probability of recovery of a quantum key by an attacker, which turns out to be close to 1 in many cases. This leads to increase of success probability when applying related key attack.

As a result we propose the usage of key derivation functions for key update.

Keywords: QKD, quantum key distribution, related key attack, Magma, Kuznyechik, attacks on QKD systems, block cipher, key derivation.

1 Introduction

Traditionally, it is desirable to obtain information-theoretic security proof for quantum key distribution protocols, which implies an assessment of the statistical distance between the uniform distribution and the distribution of key bits. The following three types of attacks are generally considered (in the order of increase of the eavesdropper capabilities):

– Individual attacks in which the eavesdropper makes independent measurements of photon states transmitted between legitimate users.

– Collective attacks in which it is assumed that the eavesdropper has quantum memory and has the ability to store measurement results for further processing.

– Coherent attacks in which the eavesdropper is assumed to have the broadest range of capabilities, including adaptive attacks (depending on the results of previous measurements).

Under this approach, encryption using the obtained keys should also be secure from the information-theoretical point of view. From a practical point of view, this leads to the need to use the Vernam cipher, which, however, does not allow to achieve an acceptable speed of information processing, since it is limited by the speed of key generation by quantum cryptographic key distribution systems (QKD systems).

As a compromise variant in the majority of existing systems, the use of QKD often consists in their integration into the existing cryptographic devices. In this case, the encryption key of the cryptographic device is periodically updated with the use of the key generated by the QKD system.

If the above mentioned Vernam cipher is used for updating the cryptographic device key (bitwise addition of the quantum key with the cryptographic device key), it is also possible to achieve in some cases information theoretic security in case of compromising the cryptographic device key or the quantum key. In each of these two cases, the compromised (known) key is encrypted using another key, which prevents the attacker from identifying the key used directly for encrypting the information.

Taking into account that up to now the assessment of practical security of the QKD systems remains questionable (first of all, regarding the attacks on the technical implementation of the quantum protocols), it is assumed that in case of compromising quantum keys, the mentioned approach will allow to keep the security of the information transmitted with the use of the cryptographic devices at the «initial» level.

At the same time, for some existing encryption algorithms the proposed approach can lead to the possibility of implementation of related keys attacks. In this regard, it is important to study the possibility of using such a method for Russian standardized encryption algorithms in case of successful implementation of an attack on the quantum component. We study the possibility of application of related key attacks on block ciphers, including those defined by the standard GOST R 34.12-2015, as well as assess their effectiveness depending on the parameters of attacks on the technical implementation of the QKD systems. We will consider attacks on modules implementing the most widely studied quantum BB84 and CV protocols.

2 Related key attacks on block ciphers

One of the known principles of cryptographic analysis is the so-called «Kerckhoffs's Law», which can be formulated as follows: «The cryptosystem should provide security even in a case when all information except a key (keys) is known to the attacker». Here, the expression «all information except for a key» means not only that the attacker does not know value of a key, but also that the attacker does not possess any indirect information on a key. More strictly the Kerckhoffs's law can be formulated in the form of an assumption that the encryption key is a realization of a uniformly distributed random variable. In practical means of cryptographic information protection the volume of the processed data with one key is often essentially limited for a number of reasons, thus for processing of big volumes of data periodic change of encryption keys is made. Hence the attacker has a set of ciphertexts encrypted with various keys, and the number of these keys can be rather large. In this case, we are dealing with a multikey attack model, and the Kerchhoff's principle can be reformulated as follows: the encryption keys are realizations of independent and uniformly distributed random variables on the key set.

A related key attack on encryption algorithms was proposed in [2]. In the related attack it is assumed that the attacker works in a multikey model, i.e. has data encrypted with different keys. At the same time, the values of the keys themselves are unknown to the attacker, but the attacker has information about some dependencies between these keys. Let's assume that the encryption is performed on the set of keys

$$(K_1, \ldots, K_m), K_i \in V_k,$$

where V_k is the key set. Let's call mapping $R : V_k^m \to \{0, 1\}$ as m-arity predicate, which is associated with a satisfiability set of this predicate

$$I_R \subseteq V_k^m,\ I_R = \{\bar{x} \in V_k^m | R(\bar{x}) = 1\}.$$

In general, the related key model assumes that the attacker has the information that the predicate for the set of keys $(K_1, \ldots, K_m)$ is satisfied, i.e. $R(K_1, \ldots, K_m) = 1$. And the predicate itself and the satisfiability set is chosen by the attacker or is known to him. A special case is the so-called functional

predicates:

$$R(K_1, \ldots, K_m) = \begin{cases} 1, & K_i = f_i(K_1),\ i = \overline{2, m}; \\ 0, & \text{otherwise}, \end{cases}$$

where $f_i : V_k \to V_k$ is a set of predefined or known functions, i.e. when the functional relations between the analyzed encryption keys is known (or chosen). In turn, a particular case of a functional relation between keys is the differential relation, which is often used in related keys differential attacks. In this case, it is assumed that

$$f_i(x) = x \oplus \Delta_i,$$

where $\oplus$ is a bitwise XOR in V_k. The values Δ_i are set by the attacker (or known to him).

Suppose that the encryption system uses a hybrid scheme for obtaining encryption keys K_i^H by adding a long-term key K^* with the keys K_i^Q distributed via QKD system, i.e. $K_i^H = K^* \oplus K_i^Q$. Then, if the QKD system has been compromised, the keys K_i^Q are known to the attacker. In this case, the encryption keys K_i^H are not known to the attacker because the long-term key is unknown. However, the attacker has information about the differences between the encryption keys:

$$K_i^H = K^* \oplus K_i^Q = K_j^H \oplus K_j^Q \oplus K_i^Q = K_j^H \oplus \delta_{ij},$$

where all

$$\delta_{ij} = K_i^Q \oplus K_j^Q$$

are known to the attacker. So we fall into the situation of a differential functional relation between the keys described above, which can lead to a significant reduction in the security of encryption algorithms.

3 Related key attacks for Magma

Related key attacks on GOST 28147-89 (Magma) was considered in [4,5]. In [4] the method of determining the key using the enhancement of the differential attack - the «boomerang» attack with related keys - was proposed.

The «boomerang» attack uses four plaintext/ciphertext pairs, where each of the ciphertexts is obtained by encryption on its own key. The basic attack of

[4] uses four encryption keys with the following relation

$$K_1, K_2 = K_1 \oplus \Delta_1, K_3 = K_1 \oplus \Delta_2, K_4 = K_1 \oplus \Delta_1 \oplus \Delta_2.$$

The attack recovers 31 bits of the first round key with time complexity of about 2^{12} encryptions and the same number of adaptive chosen plaintexts. The paper also suggests generalizations of the basic attack which recovers 192 bits of the key (6 round keys out of 8), while the remaining 64 bits are proposed to be determined by a brute force search. In this case, the attack complexity is about 2^{71} encryptions, and the data complexity is 2^{28} pairs of chosen plaintexts. A generalized attack requires a set of 14 related keys.

In [5] a combination of differential attack and boomerang method is proposed. The «boomerang» attack described in [4] is used to recover the first two round keys. For further recovery of round keys it is proposed to use the differential attack with related keys. The attack requires 12 related keys to be mounted. In the worst case scenario, the complexity of the attack is 2^{62} encryptions, and the data complexity does not exceed 2^{43} chosen text pairs.

4 Related key attacks for Kuznyechik

The analysis carried out in [7, 8] did not reveal any related key attacks for the full round Kuznyechik cipher, primarily due to its complex key schedule. Hence it is now believed that the related key attacks are not applicable to this algorithm.

5 Required related keys number and attack probability

In many cases, the application of related key attacks assumes that the relation between the keys used is set by the attacker. However, in the situation described above when compromising the QKD system, the relation between the keys is known to the attacker, but is not set by him. The value of

$$\delta_{ij} = K_i^Q \oplus K_j^Q$$

is the sum of the two keys generated from the QKD system, which are generated by a random bit generator. Thus, the values in general can be considered uniformly and independently distributed.

Then, when mounting a related key attack, the following problem arises. Suppose that we need to use m related keys $(K_1, \ldots, K_m)$ to perform the attack. The relation is defined by the differential functional predicate, i.e.

$$K_i = K_1 \oplus \Delta_i,\ i = \overline{2, m},$$

where the values Δ_i are fixed. We need to estimate the number of encryption keys M: K_j^H, $j = \overline{1, M}$ such that among them there is a subset with the necessary relation on the keys: $R(K_{j_1}^H, \ldots, K_{j_m}^H) = 1$ with the required probability.

Let's consider the simplest case, when an attack requires two keys: K_1 and $K_2 = K_1 \oplus \Delta$. The whole key set is divided into a set of key pairs with the difference Δ. Now the key set can be divided into two disjoint classes, such that if one of the keys of the pair lies in the first class, the other one lies in the second one. At the same time, the number of encryption keys K_j^H in each class will be the same on average due to the uniform distribution of encryption keys and equal cardinality of classes. Then the estimation of the probability of finding a key pair K_1 and $K_2 = K_1 \oplus \Delta$ among K_j^H is essentially the problem of estimating the probability of a collision in two samples.

It is known from the generalized «birthday paradox» that the probability of collision in two subsets of N elements, where the subset sizes are equal to $\tau_1\sqrt{N}$ and $\tau_2\sqrt{N}$ can be estimated as

$$1 - e^{-\tau_1\tau_2}$$

when $N \to \infty$. Then, assuming

$$\tau_1\sqrt{N} \approx \tau_2\sqrt{N} \approx \frac{M}{2},$$

we get that the probability of guessing the key pair K_j^H, satisfying the required relation is

$$P_2 = 1 - e^{-\frac{M^2}{4N}},$$

where $N = 2^k$ is the cardinality the key set.

Consider the situation that arises when using the «boomerang» attack with related keys. In this case, four related keys are used:

$$K_1,\ K_2 = K_1 \oplus \Delta_1,\ K_3 = K_1 \oplus \Delta_2, \text{ and } K_4 = K_1 \oplus \Delta_1 \oplus \Delta_2.$$

As before, the key set can be divided into 4 disjoint classes of equal size, and

the estimate of the probability of finding the four related keys can be obtained through the estimate of the probability of 4-multicollision. In [3] Wagner proposed an efficient algorithm for solving the problem, which gives us the correspondence between the desired probability and the amount of data need. As a result this allows us to correlate the number of related keys available to the attacker with the probability that there exists a subset with the required property, and hence, to evaluate the success rate of the attack.

In classical assumptions, when keys are assumed to be uniformly and independently distributed, the attacker acts under the assumption that the keys are related. Therefore, the probability of the attack includes a factor that corresponds to the probability of this assumption being fulfilled which is equal to

$$2^{-k(r-1)}$$

when using r keys. This factor makes the overall success rate of the attack almost zero.

In the case of a compromised QKD system, when the differences between the keys are known and uniformly distributed, and when the attack applicable to one particular relation predicate is known (as in the above attacks for the Magma cipher), the number of encryption keys required for the attack is quite large: to achieve success rate close to 1 for an attack with two related keys the attacker needs about

$$\sqrt{N} = \sqrt{|V_k|} = 2^{128}$$

encryption keys. For an attack with four related keys:

$$(|V_K|)^{\frac{3}{4}} = 2^{196}.$$

Hence if the attacker has a small number of encryption keys, the probability estimates will be almost the same as in the classical assumptions.

6 Practical attacks on quantum key distribution systems

In this section we will briefly review some known practical attacks exploiting implementation weaknesses of optical modules of QKD systems. We focus mostly on systems implementing widely studied BB84 and CV QKD systems. Our goal is to to assess the a posterior probability of the quantum key bits, or, in other words, the probability with which the eavesdropper knows the quantum

key, after the attack. It should be noted that in order to provide information-theoretic security of the quantum key privacy amplification techniques should be performed after error correction. This is usually done by means of 2-universal hash functions family [14]. That means, that quantum key, which is possibly eavesdropped, is loaded into the cryptographic device after some functional transformation. Since the procedures of error correction and privacy amplification of the key is determined, it is enough to estimate such probability P for the raw (before error correction) key.

6.1 Photon-number splitting attack

One of the main problems for QKD systems is to implement a single-photon source in practice. In majority of implementations a weak coherent pulses are used, which means that source emits multi-photon pulses with non-zero probability. In this case, Eva has the ability to split the photon beam in order to intercept one of the photons without affecting other and store it in quantum memory. After legitimate users announce their bases, Eva has the ability to measure the stored photon and get the encoding.

In [15] a thorough study is performed. The probability for eavesdropper to get a correct key depends on the transmission efficiency of a quantum channel η, mean photon number μ and proportion of the pulses containing one photon κ. The probability of correct key guessing for Eve as a function of disturbance D which is introduced for the information channel between the legitimate users is described as:

$$P(D) = \frac{1 - e^{-\mu}(1+\mu) + (1-\kappa)\mu e^{-\mu}[1/2 + \sqrt{D(1-D)}]}{1 - e^{-\mu}(1+\mu\kappa)}$$

The probability depending on the parameters could be up to 1, which means that the eavesdropper knows the whole key.

6.2 Detector laser damage

In [16] an attack based on high voltage laser damage of photodetectors is proposed. In the worst case the attacker by destroying detectors is able to get the full control over the process of quntum key generation. That means the probability P could be equal to 1 in this attack.

6.3 Trojan horse attack

In this type of attack, the eavesdropper irradiates the laser of the encoding module and receives information about the coding of the photon by analyzing the reflected signal. In [1] the practical application of the attack for two QKD systems is shown. The research suggests that it is possible to correctly determine the key bit with a probability of $P > 0.99$.

6.4 Bright illumination attack

This type of attack can be used against avalanche photo detectors [13]. During the attack, the photo detector is irradiated by a powerful beam of light, which leads to the transition of the detector from geiger mode to linear mode. As a result, an eavesdropper can carry out a meet-in-the-middles attack by measuring the photons sent by initiator and inducing, according to the measurement, the response of the corresponding photodetector on the receiver side. As a result, an eavesdropper receives full information about the key being transmitted $P = 1$.

6.5 Time shift attack

In order to minimize the effect of dark readings in a number of systems an activation of photon detectors on the signals of synchronization is implemented. The possibility of an attack arises from the fact that the detection efficiency profiles of photon detectors are not the same. In this case, an eavesdropper can measure photons, and then change the synchronization signal to activate the receiver's photo detector according to his measurements in order to control detection efficiency according to her measurements. In the worst case, an eavesdropper can get all the information about the key [10].

6.6 Wavelength attack

Many of QKD systems implement beam splitter on the receiver side. The imperfectness of the beam splitter's wavelength dependent optical property could be exploited by the eavesdropper. The study [17] suggests that the eavesdropper after measuring the photon resend it to the receiver at the wavelength depending on the measured state and the account optical properties of the beam splitter. In the worst case the eavesdropper could get all the information about the key.

7 On the possibility of related key attack and countermeasures

The review, presented in the previous clause, suggests that many of practically possible attacks on QKD make it possible for the eavesdropper to have the full knowledge of quantum key. At the same time the main problem with the implementation of related key attack from the attacker viewpoint is the impossibility of imposing desired keys to legitimate subscribers. Even when the eavesdropper is able to get full control over optical parts of the legitimate users, the process of encoding photons on sender side is beyond his control. As a result, after the application of the key correction and privacy amplification procedures, the eavesdropper will not be able to impose the keys with the specified ratios. As a result, even if the raw key is compromised, the eavesdropper will be able to expect the related keys to appear only with the specified probability.

At the same time, taking into account the fact that in the conditions under consideration the eavesdropper is able to detect the fact of appearance of related keys, which increases the reliability of the attack, and, in general, reduces the security of the cryptographic device, it seems reasonable to use methods key derivation that exclude the possibility of considered attacks.

The most effective countermeasure in this case is the use of the key derivation functions, for example, defined by the recommendation for standardisation [6, 18]. The key derivation functions $kdf(S, T, L, P \ldots)$ described in [6] consist of two stages. At the first stage, an intermediate key $K^{(1)} = kdf^{(1)}(T, S)$ is produced, which is obtained by hashing of the original secret key S using «salt». T - the vector which is supposed to be uniformly distributed on some set and, in general, is considered known to the attacker. At the second stage, the derivative keys $K^{(2)} = kdf^{(2)}(K^{(1)}, L, P, ...)$ are generated from the intermediate key using the cryptographic transformations, where L is the length of the key sequence produced, and P - additional information.

In order to provide the security of cryptographic device in case of the QKD system compromise, it is proposed to use the key derivation function, where the keys obtained as a result of the quantum protocol are used as salt K_i^Q and the long-term key of a cryptographic device K^* is used as the secret key.

As a result of this approach, even in the case of compromising the quantum keys K_i^Q, the security proofs of [9] about the computational indistinguishability of the keys produced by $K_i^H = kdf(K^*, K_i^Q, L, P \ldots)$ from the sequence of

independent, equal-probable random variables remain valid.

8 Conclusion

We studied the possibility of applying related key attacks to hybrid cryptographic devices which uses QKD for updating the long-term key in case of quantum channel compromise.

For the key update method, which consists in bitwise XOR of quantum and long-term key, it is shown the increase of success probability of the related key attack in comparison with the classical conditions.

A variant of the key update method based on standardised key derivation functions, which excludes the possibility of application of the considered attacks, is proposed.

References

[1] Stiller B., Khan I., Jain N., Jouguet P., Kunz-Jacques S., Diamanti E., Marquardt Ch., and Leuchs G., "Quantum hacking of continuous-variable quantum key distribution systems: real-time Trojan-horse attacks".

[2] Biham E., "New types of cryptanalytic attacks using related keys", *J. Cryptology*, **7**:4 (1994), 229-246.

[3] Wagner D., "A generalised birthday problem", Crypto'02, 2002, 288-303.

[4] Rudskoy V., *On zero practical significance of «Key recovery attack on full GOST block cipher with zero time and memory»*, 2011, http://eprint.iacr.org/2010/111.

[5] Pudovkina M.A. and Khoruzenko G.I., "Attacks on full block cipher GOST 28147-89 with 2 or 4 related keys", *Prikl. Diskr. Mat.*, **3** (2010).

[6] "Recommendation for standardisation R 1323565.1.022-2018. Information technologies. Cryptograhic data protection. Key derivation functions", In Russian.

[7] Alekseev E., Goncharenko K., and Marshalko G., "Provably secure counter mode with related key-based internal rekeying", CTCrypt 2018 Preproceedings, 2018.

[8] Ishchukova E.A., Krasovsckiy A.V., Polovko I.Yu., "Analysis of the cipher Kuznyechik by the related keys method", *Modern high technologies*, **5** (2018).

[9] Bellare, M., "New Proofs for NMAC and HMAC: Security without collision resistance", Advances in Cryptology - CRYPTO 2006, 26th Annual International Cryptology Conference, *Lecture Notes in Computer Science*, **4117** (2006), 602-619.

[10] B. Qi, C. Fung, H. Lo, et al., *Time-shift attack in practical quantum cryptosystems*, 2005, quant-ph/0512080.

[11] H. Li, S. Wang, J. Huang, et al., "Attacking a practical quantum-key-distribution system with wavelength-dependent beam-splitter and multiwavelength sources", *Physical Review A*, **84**:6 (2011).

[12] H. Weier, H. Krauss, M. Rau, et al., "Quantum eavesdropping without interception: an attack exploiting the dead time of single-photon detectors", *New Journal of Physics*, **13** (2011).

[13] L. Lydersen, J. Skaar and V. Makarov, "Tailored bright illumination attack on distributed-phase-reference protocols", *Journal of Modern Optics*, 2011.

[14] G. Gilbert, M. Hamrick, F.J. Thayer, *Privacy Amplification in Quantum Key Distribution: Pointwise Bound versus Average Bound*, 2001, https://arxiv.org/abs/quant-ph/0108013v1.

[15] M. Williamson, V. Vedral, "Eavesdropping on practical quantum cryptography", *J. Mod. Opt.*, **50**:13 (2003).

[16] Audun N. Bugge, Sebastien Sauge, Aina M. M. Ghazali, Johannes Skaar, Lars Lydersen, Vadim Makarov, "Laser damage helps the eavesdropper in quantum cryptography", *Phys. Rev. Lett.*, **112** (2014).

[17] H.-W. Li et al., "Attacking practical quantum key distribution system with wavelength dependent beam splitter and multi-wavelength sources", *Physical Review A*, **84** (2011).

[18] "Recommendation for standardisation R 50.1.113-2016. Information technologies. Cryptograhic data protection. Cryptographic algorithms to accompany the usage of digital signature and hash function", In Russian.

Optimization of S-boxes GOST R 34.12-2015 «Magma» Quantum Circuits Without Ancilla Qubits

Denis Denisenko and Marina Nikitenkova

Bauman Moscow State Technical University (BMSTU), Russia
denisenkodv@bmstu.ru, marina-nik-msc@yandex.ru

Abstract

The work is devoted to the study of ways to implement S-boxes in the form of quantum circuits with a minimum number of logical qubits and logical quantum gates, without using ancilla qubits. New quantum circuits that implement the S-boxes of the GOST R 34.12-2015 "Magma" on 4 logical qubits have obtained. We have concluded that for substitutions $s \in S(V_n)$ with a large number of cycles there are quantum circuits on n logical qubits that implement the substitution s with fewer logical quantum gates, compared with substitutions $g \in S(V_n)$ with a small number of cycles.

Keywords: S-box, quantum circuit, resource estimates.

1 Introduction

The theory of quantum computing has been developing since the end of the 20^{th} century. A number of formal quantum computing models are constructed in which some computational problems, for example, [1, 2, 3], are solved more efficiently than in the classical computational model [4]. Actual information about the current level of quantum technologies development in the field of quantum computing is presented in [5, 6].

In [7] we consider a method for implementing S-boxes GOST R 34.12-2015 and AES in the form of quantum circuits without using ancilla qubits, based on an algorithm for implementing an arbitrary unitary operator in the form of a quantum circuit by decomposing the corresponding unitary matrix into a product of two-level unitary matrices ([8, sec. 4.5]). Quantum circuits that implement S-boxes of GOST R 34.12-2015 and AES without ancilla qubits were constructed taking into account the optimization, based on the removal of sequences of quantum operations that equival to the identical transformation.

For the implementation of GOST R 34.12-2015 "Magma" S-boxes 4 logical qubits are enough, and for the implementation of GOST R 34.12-2015 "Kuznechik" and AES S-boxes 8 logical qubit are enough, while in [9, 10] the implementation of AES S-box requires 40 logical qubits. In [10] a method for constructing quantum circuit is described, which implements the AES S-box on 9 logical qubits, but it is argued that in comparison with 40 logical qubits case, this implementation requires approximately three times more quantum gates.

In this work for constructing quantum circuits (Fig. 1-8) we have used generalized $CNOT(C|t)$ gates ([8, 11]), in which qubit t is controlled by the set of qubits C. Generalized gates $CNOT(C|t)$ can be implemented without using ancilla qubits ([8, p. 184]), therefore, we will consider generalized gates $CNOT(C|t)$ as one self-independent logic gate, which is consistent with the techniques of assessing the quantum resources described in [12, 13].

2 Construction of quantum circuits implementing S-boxes without ancilla qubits

In this section, we present an algorithm for constructing quantum circuits that implement S-boxes without ancilla qubits, based on the decomposition of the substitution into independent cycles.

Table 1 presents the results of our implementation of Algorithm 1.

S-box	data' – sequences of elementary qubit states transformations
π_0	(14,15)(13,14)(11,14)(10,13)(9,14)(8,14)(7,9)(6,11)(12,0)(4,10)(3,6)(2,6)(1,4)
π_1	(10,14)(11,12)(9,14)(7,12)(1,8)(6,10)(5,10)(4,9)(6,0)
π_2	(1,3)(9,1)(13,9)(5,15)(7,13)(10,7)(6,10)(14,6)(2,5)(4,2)(8,14)(11,4)(0,11)
π_3	(14,15)(13,14)(12,15)(11,13)(9,12)(8,15)(7,15)(6,15)(5,13)(4,13)(3,8)(1,8)(12,0)
π_4	(5,1)(0,7)(2,5)(8,0)(14,2)(4,8)(13,4)(10,3)(11,14)(12,11)(15,12)
π_5	(12,13)(11,13)(10,11)(9,10)(8,11)(7,10)(5,15)(6,12)(4,9)(2,15)(3,6)(5,0)(1,13)
π_6	(13,15)(9,14)(12,13)(6,14)(5,9)(11,15)(10,11)(4,6)(3,5)(8,15)(1,14)(7,12)(8,0)
π_7	(7,3)(1,7)(0,1)(4,0)(8,4)(6,8)(11,6)(14,11)(2,14)(15,2)(9,15)(12,9)(13,12)

Table 1: Data from algorithm 1 for implementing S-boxes without ancilla qubits.

In Fig. 1-8 the quantum circuits are presented that implement the S-boxes of GOST R 34.12-2015 "Magma". Optimization of quantum circuits that implement S-boxes GOST R 34.12-2015 "Magma" was carried out by "brute force",

Algorithm 1

Input: Substitution $s \in S(V_n)$.

Output: Quantum circuit for $s \in S(V_n)$ without ancilla qubits.

1: Represent s as a product of independent cycles and remove fixed points. As a result, we obtain $k \in \mathbb{N}$ independent cycles, $s = s_1 s_2 \dots s_k$;

2: Consider each cycle s_i, $i \in \overline{1,k}$ as an independent substitution, find the decomposition of the unitary matrix U_{s_i} corresponding to the cycle $s_i, i \in \overline{1,k}$, into the product of two-level matrices (see [7, 8]):

$$U_{s_i} = V_1^i \cdot \ldots \cdot V_t^i;$$

3: By the found matrices $V_1^i \cdot \ldots \cdot V_t^i$ we could determine all possible pairs of states

$$data_i = \{(x_j^i, y_j^i) : V_j^i |x_j^i\rangle = |y_j^i\rangle, i \in \overline{1,k}, j \in \overline{1,t}, x_j^i \neq y_j^i\}.$$

Lists $data_i$ can be simply written according to the cycles of $s_i, i \in \overline{1,k}$, by restoring the transition table of s_i. Description of formation of lists $data_i$ through two-level matrixes is given in order to define $data_i$ strictly and unambiguously.

4: For each pair $(x_j^i, y_j^i) \in data_i$, $i \in \overline{1,k}, j \in \overline{1,t}$, define the list of bit numbers $numb_{(x_j^i, y_j^i)} \subset \{1, 2, \dots, n\}$, where x_j^i is different from y_j^i.

5: Denote $data = \bigcup_{i=1}^{k} data_i$. It is required to sort the elements of $data$ in such a way that

$$\sum_{w=1}^{|data|-1} |numb_{(x_w, y_w)} \cap numb_{(x_{w+1}, y_{w+1})}| \to max,$$

moreover, the transitions (x_j^i, y_j^i) obtained through the cycle s_i must preserve the relative order (otherwise, instead of the cycle s_i we will get some other cycle s'). This stage can be implemented using random search with restrictions. As a result of this sorting of $data$ we get $data'$.

6: To each element $(x', y') \in data'$ still corresponds some two-level matrix, i.e. each element of $data'$ could be easily implemented using some simple quantum circuit (see [7, 8]). The implementation of two-level matrices in the form of quantum circuit can be ambiguous. Let $d_{(x',y')}$ be the Hamming distance between the binary representations x' and y', then there exists exactly $d_{(x',y')}!$ various quantum circuits consisting from quantum gates CNOT and generalized CNOT(C|t), that implement the transition (x', y') (see [8]). Among them there are only $d_{(x',y')}$ quantum circuits that differ significantly, which are determined by the number t of the controlled qubit in quantum gates CNOT(C|t) that occurs during the implementation of two-level matrices.

7: We assume that the quantum circuit that implements the transitions of $data'$ could be optimized by independent parts. Then the search for an optimized quantum circuit for $s \in S(V_n)$ can be organized using the following procedure:

a: Initialize an array of $|data'|$ elements $memory = \{1, \ldots, 1\}$. In $memory[iter]$ we will write the number of implementation of the two-level matrix that implements the transition $(x'_{iter}, y'_{iter}) \in data'$, $iter \in \overline{1, |data'|}$. Set $iter = 1$.

b: Set the search depth, for example, $depth = 3$. Until $iter < |data'|$ do:

i: Search for a quantum circuit with minimum length that implements transitions $(x'_{iter}, y'_{iter}), \ldots, (x'_{iter+depth-1}, y'_{iter+depth-1})$ by iterating over all possible variants of quantum circuits that implements $(x'_{iter}, y'_{iter}), \ldots, (x'_{iter+depth-1}, y'_{iter+depth-1})$. Write the founded numbers of implementations of the two-level matrices to $memory[iter], \ldots, memory[iter + depth - 1]$.

ii: $iter = iter + depth$; The depth of the search is determined by the available computing power.

c: Repeat this procedure starting at $iter = 2$ and the same value of $depth$.

8: As a result, we obtain a quantum circuit that implements $s \in S(V_n)$ without using ancilla qubits, with the minimal number of quantum gates.

i.e. for Magma in Algorithm 1, we have set $depth = |data'| - 1$. Comparison of the number of quantum gates in quantum circuits, at [7] and pic. 1-8, that implementat the S-box without ancilla qubits is given in table 2.

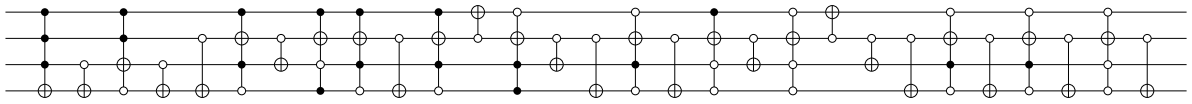


Figure 1: Quantum circuit for π_0 = (12, 4, 6, 2, 10, 5, 11, 9, 14, 8, 13, 7, 0, 3, 15, 1).

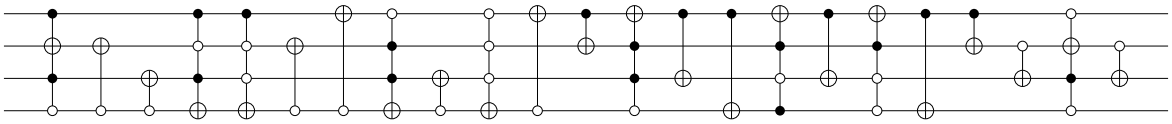


Figure 2: Quantum circuit for π_1 = (6, 8, 2, 3, 9, 10, 5, 12, 1, 14, 4, 7, 11, 13, 0, 15).

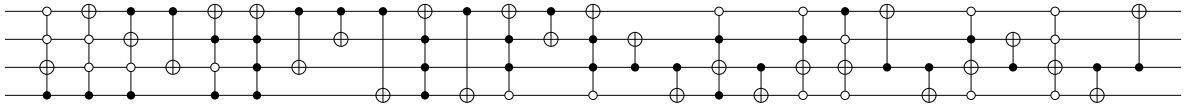


Figure 3: Quantum circuit for π_2 = (11, 3, 5, 8, 2, 15, 10, 13, 14, 1, 7, 4, 12, 9, 6, 0).

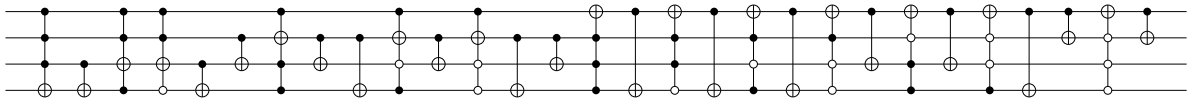


Figure 4: Quantum circuit for $\pi_3 = (12, 8, 2, 1, 13, 4, 15, 6, 7, 0, 10, 5, 3, 14, 9, 11)$.

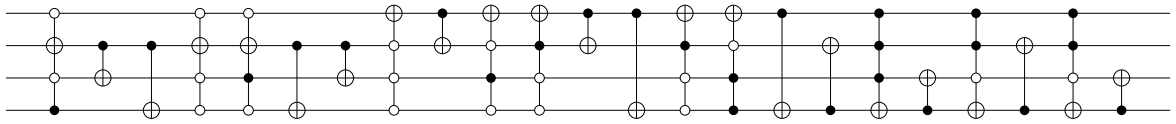


Figure 5: Quantum circuit for $\pi_4 = (7, 15, 5, 10, 8, 1, 6, 13, 0, 9, 3, 14, 11, 4, 2, 12)$.

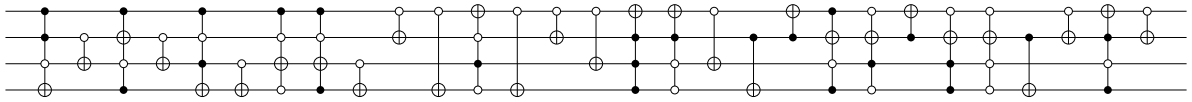


Figure 6: Quantum circuit for $\pi_5 = (5, 13, 15, 6, 9, 2, 12, 10, 11, 7, 8, 1, 4, 3, 14, 0)$.

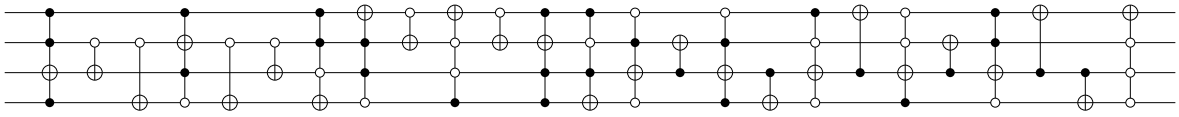


Figure 7: Quantum circuit for $\pi_6 = (8, 14, 2, 5, 6, 9, 1, 12, 15, 4, 11, 0, 13, 10, 3, 7)$.

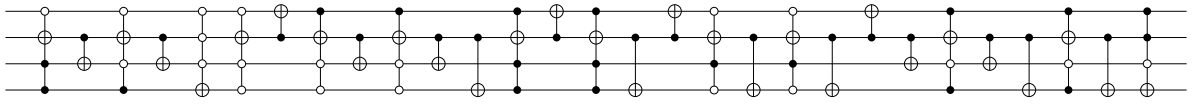


Figure 8: Quantum circuit for $\pi_7 = (1, 7, 14, 13, 0, 5, 8, 3, 4, 15, 10, 6, 9, 12, 11, 2)$.

Source code for verifying correctness of quantum circuits in the quantum simulator *Quipper* [14], example for π_4.

```
import Quipper
import QuipperLib.Simulation
import System.Random
import Quipper.Printing
import Quipper.QData
```

```
-------------------------
sub1 :: ([Qubit]) -> Circ ([Qubit])
sub1 (input) = do
  let [q0, q1, q2, q3] = input
  qnot_at q1 `controlled` (q0.==.0,q2.==.0,q3)
  qnot_at q2 `controlled` [q1]
  qnot_at q3 `controlled` [q1]
  qnot_at q1 `controlled` (q0.==.0,q2.==.0,q3.==.0)
  qnot_at q1 `controlled` (q0.==.0,q2,q3.==.0)
  qnot_at q3 `controlled` [q1]
  qnot_at q2 `controlled` [q1]
  qnot_at q0 `controlled` (q1.==.0,q2.==.0,q3.==.0)
  qnot_at q1 `controlled` [q0]
  qnot_at q0 `controlled` (q1.==.0,q2,q3.==.0)
  qnot_at q0 `controlled` (q1,q2.==.0,q3.==.0)
  qnot_at q1 `controlled` [q0]
  qnot_at q3 `controlled` [q0]
  qnot_at q0 `controlled` (q1,q2.==.0,q3.==.0)
  qnot_at q0 `controlled` (q1.==.0,q2,q3)
  qnot_at q3 `controlled` [q0]
  qnot_at q1 `controlled` [q3]
  qnot_at q3 `controlled` (q0,q1,q2)
  qnot_at q2 `controlled` [q3]
  qnot_at q3 `controlled` (q0,q1,q2.==.0)
  qnot_at q1 `controlled` [q3]
  qnot_at q3 `controlled` (q0,q1,q2.==.0)
  qnot_at q2 `controlled` [q3]
  return ([q0, q1, q2, q3])

test1_circuit :: IO ()
test1_circuit = do
  putStrLn "Quantum Cirquit in Adobe Reader..."
  print_generic  Preview (sub1) (replicate 4 qubit)

test1_exec :: IO ()
test1_exec = do
  putStrLn "Substitution functionality test:"
  print_generic GateCount sub1  (replicate 4 qubit)
  g <- newStdGen
  --Here are only 2 states, but you could check all 16:
  print $ run_generic g(0.0::Double) sub1 ([True,True,True,False])
  print $ run_generic g(0.0::Double) sub1 ([True,True,True,True])

--Run--
main = do
  test1_circuit
  test1_exec
```

S-box	Number of cycles in S-box	Total gates in quantum circuit [7]	Total gates in new quantum circuit
π_0	2	33	29
π_1	3	29	23
π_2	2	37	27
π_3	1	29	29
π_4	3	31	23
π_5	2	35	29
π_6	2	31	25
π_7	1	31	29

Table 2: Comparison of the number of quantum gates in quantum circuits, at [7] and Fig. 1-8, that implement the S-box without ancilla qubits.

3 Conclusion

We have obtained new quantum circuits for implementation GOST R 34.12-2015 "Magma" S-boxes on 4 logical qubits with fewer logical quantum gates than in [7]. The obtained results allow to draw a conclusion that the more cycles in substitution, the less the length of the quantum circuit implementing this substitution can be.

References

[1] Shor P.W., "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer", *SIAM Journal of computing*, **26**:5 (1997), 1484–1509, DOI:10.1137/S0097539795293172.

[2] Grover L.K., "A fast quantum mechanical algorithm for database search", *Proceedings, 28th Annual ACM Symposium on the Theory of Computing (STOC)*, 1996, 212–219, DOI:10.1145/237814.237866.

[3] Simon D.R., "On the power of quantum computation", *SIAM Journal of computing*, **26**:5 (1997), 1474–1483, DOI:10.1137/S0097539796298637.

[4] Gainutdinova A.F., "Comparative complexity of quantum and classical models of calculations.", Thesis for the degree of Candidate of Physical and Mathematical Sciences. Specialty 01.01.09 - discrete mathematics and cybernetics, 2004, http://www.dissercat.com/content/sravnitelnaya-slozhnost-kvantovykh-i-klassicheskikh-modelei-vychislenii.

[5] "Quantum Computing Report", https://quantumcomputingreport.com.

[6] National Academies of Sciences, Engineering, and Medicine, "Quantum Computing: Progress and Prospects", *The National Academies Press.*, 2018, DOI:10.17226/25196.

[7] Denisenko D.V., "Quantum circuits for S-box implementation without ancilla qubits", *ZhETF*, **155**:6 (2019), 999 pp.

[8] Nielsen M.A., Chuang I.L., *Quantum computation and quantum information*, Cambridge Univ. Press, 2010, http://csis.pace.edu/ctappert/cs837-18spring/QC-textbook.pdf.

[9] Kim P., Han D. Jeong K.C., "Time-space complexity of quantum search algorithms in symmetric cryptanalysis: applying to AES and SHA-2.", *Quantum Inf Process*, **17** (2018), 339 pp., https://doi.org/10.1007/s11128-018-2107-3.

[10] Grassl M., Langenberg B., Roetteler M., Steinwandt R., "Applying Grover's algorithm to AES: quantum resourse estimation", *Cryptology ePrint Archive, Report 2019/103*, 2015, arXiv:1512.04965v1.

[11] Younes A., Miller J., "Automated method for building CNOT based quantum circuits for boolean functions.", 2003, arXiv:quant-ph/0304099v1.

[12] Samuel Jaques and John M. Schanck, "Quantum cryptanalysis in the RAM model: Claw-finding attacks on SIKE", *Cryptology ePrint Archive*, **Report 2019/103** (2019), https://eprint.iacr.org/2019/103.

[13] National Institute of Standards and Technology, "Submission reuirements and evaluation criteria or the post-quantum cryptography standardization process", 2017, https://csrc.nist.gov/csrc/media/projects/post-quantum-cryptography/documents/call-for-proposals-final-dec-2016.pdf.

[14] "The Quipper Language", http://www.mathstat.dal.ca/ selinger/quipper/.

Appendix 1

The algorithm for constructing quantum circuits of an arbitrary unitary operator is described in [8], Section 4.5.

Definition 1. *Let $N = 2^n$, $n \in \mathbb{N}$, and $e_1, e_2, \ldots, e_N$ be the basis of the vector space $L_{\mathbb{C}^N}$ over field of complex numbers $\mathbb{C}$. The unitary matrices $U \in \mathbb{C}_{2^n,2^n}$, nontrivially acting on no more than two basis vectors $e_1, e_2, \ldots, e_N$, are called* two-level unitary matrices *(see [8], section 4.5.1).*

Let's construct a quantum circuit that implements

$$\pi_1 = (6, 8, 2, 3, 9, 10, 5, 12, 1, 14, 4, 7, 11, 13, 0, 15).$$

The substitution $\pi_1 \in S(V_4)$. Denote $y = \pi_1(x)$, $x, y \in V_4$. The states $|x\rangle, |y\rangle$ are vector-columns from $L_{\mathbb{C}^{2^4}}$, the action of the operator $U|x\rangle = |y\rangle$ is a multiplication of the column vector $|x\rangle$ by the matrix $U \in \mathbb{C}_{2^4,2^4}$.

1. The unitary matrix for π_1:

$$U_{\pi_1} = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}.$$

2. The matrix U_{π_1} can be represented as a product of two-level unitary matrices:

$$U_{\pi_1} = V_1 \cdot V_2 \cdot V_3 \cdot V_4 \cdot V_5 \cdot V_6 \cdot V_7 \cdot V_8 \cdot V_9.$$

The table 3 contains two-level matrices $V_1, \ldots, V_9$, participating in the decomposition U_{π_1}, states s and t, on which two-level matrices act nontrivially, and quantum circuits implementing two-level matrices $V_1, \ldots, V_9$. Matrices are written as a list of strings, each row is a vector $v_i \in V_{16}$, $||v_i|| = 1$, $i \in \overline{1, 16}$ and written in hexadecimal notation.

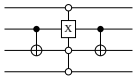
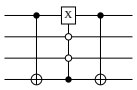
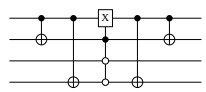
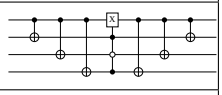
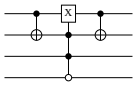
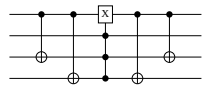
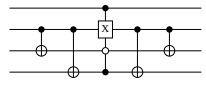
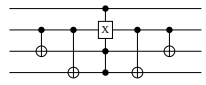
$V_1 = \{200, 4000, 2000, 1000, 800, 400, 8000, 100, 80, 40, 20, 10, 8, 4, 2, 1\}$	$s = \|0110\rangle$ $t = \|0000\rangle$	
$V_2 = \{8000, 80, 2000, 1000, 800, 400, 200, 100, 4000, 40, 20, 10, 8, 4, 2, 1\}$	$s = \|0001\rangle$ $t = \|1000\rangle$	
$V_3 = \{8000, 4000, 2000, 1000, 40, 400, 200, 100, 80, 800, 20, 10, 8, 4, 2, 1\}$	$s = \|0100\rangle$ $t = \|1001\rangle$	
$V_4 = \{8000, 4000, 2000, 1000, 800, 20, 200, 100, 80, 40, 400, 10, 8, 4, 2, 1\}$	$s = \|0101\rangle$ $t = \|1010\rangle$	
$V_5 = \{8000, 4000, 2000, 1000, 800, 400, 20, 100, 80, 40, 200, 10, 8, 4, 2, 1\}$	$s = \|0110\rangle$ $t = \|1010\rangle$	
$V_6 = \{8000, 4000, 2000, 1000, 800, 400, 200, 8, 80, 40, 20, 10, 100, 4, 2, 1\}$	$s = \|0111\rangle$ $t = \|1100\rangle$	
$V_7 = \{8000, 4000, 2000, 1000, 800, 400, 200, 100, 80, 2, 20, 10, 8, 4, 40, 1\}$	$s = \|1001\rangle$ $t = \|1110\rangle$	
$V_8 = \{8000, 4000, 2000, 1000, 800, 400, 200, 100, 80, 40, 2, 10, 8, 4, 20, 1\}$	$s = \|1010\rangle$ $t = \|1110\rangle$	
$V_9 = \{8000, 4000, 2000, 1000, 800, 400, 200, 100, 80, 40, 20, 8, 10, 4, 2, 1\}$	$s = \|1011\rangle$ $t = \|1100\rangle$	

Table 3: Representation of the matrix U_{π_1} as a product of two-level matrices $U_{\pi_1} = V_1 \cdot V_2 \cdot V_3 \cdot V_4 \cdot V_5 \cdot V_6 \cdot V_7 \cdot V_8 \cdot V_9$.

Since $|y\rangle = U|x\rangle$, $|y\rangle = V_1 \cdot \ldots \cdot (V_8 \cdot (V_9 \cdot |x\rangle))$, the quantum circuit of U_{π_1} is as follows:

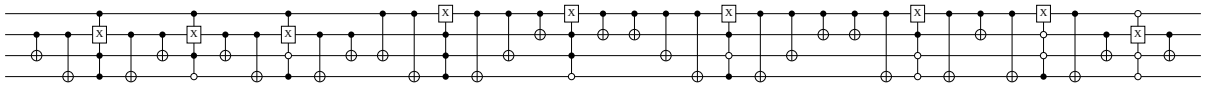


Figure 9: Quantum circuit $\pi_1 = (6, 8, 2, 3, 9, 10, 5, 12, 1, 14, 4, 7, 11, 13, 0, 15)$.

After optimization of the quantum circuit in the figure 9, we obtain the quantum circuit in the figure 2.

Symmetric cryptography

Related-key Attack on 5-Round Kuznyechik

Vitaly Kiryukhin

JSC «InfoTeCS», Russia
vitaly.kiryukhin@infotecs.ru

Abstract

The first related-key attack on 3-round (of 9) Kuznyechik with 2-round (of 8) key schedule was presented in CTCrypt'18. This article describes a related-key attack on 5-round cipher with the same key schedule. The presented one also has a practical complexity (2^{32} operations, 2^{30} memory, 2^{16} related keys) and verified in practice. We obtained result due to the simultaneous use of the integral properties of the cipher transformations and the key schedule.

Keywords: Kuznyechik, related-key attack, integral cryptanalysis.

1 Introduction

The setting of a related-key attack on cipher was introduced in [6]. Informally this model assumes that adversary has access to several encryptors with different unknown keys, but it knows a certain simple relationship (for example, bitwise xor) between these keys.

In some cases the related-key model is quite consistent with reality. A good example is an iterative hash function using block cipher as part of compression function. In this case, adversary has a possibility of manipulating the encryption keys. Some cryptographic protocols may use related keys by design. One such related-key protocol CTRR was proposed at CTCrypt'18 [2].

In the same publication [2], the first related-key attack on a reduced variant of block cipher Kuznyechik [1] was proposed. This approach exploits the ability of attacker to manipulate keys, and the similarity of the functions in encryption and the key schedule procedures.

In this paper we present a related-key attack on 5-round (of 9) Kuznyechik with 2-round (of 8) key schedule. Main result obtained due to the integral properties [4, 5] of encryption and the key schedule. We also used some approaches

from [3]. The simplified versions of Kuznyechik are described in the next section (equations (2) and (3)).

The presented attack was verified in practice with the help of C++ implementation. Source codes can be found at `https://gitlab.com/v.kir/rk-5R-kuznyechik`.

Comparative characteristics of attacks are presented in table 1.

Cipher rounds	Key schedule rounds	Operations	Keys	Memory	Source
3	2	2^{12}	2^{12}	$\sim$	[2]
5	2	2^{32}	2^{16}	2^{30}	Section 4

Table 1: Related-key attacks on Kuznyechik

2 Definitions

Let $\mathbb{F}_{2^8}$ be a finite field as defined in [1]. Each element of $\mathbb{F}_{2^8}$ can be interpreted as an integer or binary vector. Field elements are indicated by lowercase letters: a, b. Denote vector space of dimension $n \in \mathbb{N}$ over $\mathbb{F}_{2^8}$ by $\mathbb{F}_{2^8}^n$. Elements from $\mathbb{F}_{2^8}^n$ will be denoted by capital letters: A, B. Blocks of plaintext and ciphertext also belong to $\mathbb{F}_{2^8}^n$.

Denote bitwise xor operation by symbol $\oplus$. Let we have a sequence of blocks

$$B_0, ..., B_d \in \mathbb{F}_{2^8}^n, d \in \mathbb{N},$$

then we refer to sequence

$$\boldsymbol{\Delta B} = (B_0 \oplus B_1, B_0 \oplus B_2, ..., B_0 \oplus B_d) \in (\mathbb{F}_{2^8}^n)^d \tag{1}$$

as a difference. Throughout the article we always use $d = 2^8 - 1$. Differences are indicated by bold: $\boldsymbol{\kappa}$, $\boldsymbol{\Delta K}$.

The transformations over $\mathbb{F}_{2^8}^n$ (or sets of elements from $\mathbb{F}_{2^8}^n$) are denoted by Sans Serif font: f, S, L. Such characters may mean a bijective transformation of blocks ($\mathsf{f}(A)$, $A \in \mathbb{F}_{2^8}^n$) or non-bijective transformation of differences to the set of differences (for example, $\mathsf{S}(\boldsymbol{\kappa})$ is a set of differences, $\boldsymbol{\kappa} \in \left(\mathbb{F}_{2^8}^n\right)^d$). The notation LS indicates a composition of transformations, where S applies first.

The difference $\boldsymbol{\Delta} \in \left(\mathbb{F}_{2^8}^n\right)^d$ can also be interpreted as n «columns» of d bytes each: $\boldsymbol{\Delta} \in \left(\mathbb{F}_{2^8}^d\right)^n$. If i-th «column» ($i = 1, 2, \ldots, n$) $\boldsymbol{\alpha} \in \mathbb{F}_{2^8}^d$ contains all different non-zero bytes, we say that i-th position has an integral property

All (A). Similarly, if xor of all bytes is equal to zero, then i-th position of the difference has an integral property *Zero* ($\mathbf{0}$). Obviously, the property A implies the property $\mathbf{0}$. If at least one byte in such «column» is non-zero, we say that i-th position is active, otherwise inactive.

Kuznyechik

Kuznyechik [1] consists of a sequence of 9 rounds and a post-whitening key addition. Each round contains three operations:

X – modulo 2 addition of an input block with an iterative key;

S – parallel application of a fixed bijective substitution to each byte of the block;

L – linear transformation defined as an LFSR over $\mathbb{F}_{2^8}$.

The block size is 128 bits ($n = 16$ bytes), the size of key K is equal to 256 bits.

Key schedule uses round constants $C_i \in \mathbb{F}_{2^8}^n$, $i = 1, 2, \ldots, 32$.

Round keys $K_i \in \mathbb{F}_{2^8}^n$, $i = 1, 2, \ldots, 10$ are derived from a master key K as follows:

$$K = K_1||K_2,$$
$$(K_{2i+1}, K_{2i+2}) = \mathsf{F}[C_{8(i-1)+8}] \ldots \mathsf{F}[C_{8(i-1)+1}]\,(K_{2i-1}, K_{2i})\,,\ i = 1, 2, 3, 4,$$
$$\mathsf{F}[C]\,(A_1, A_2) = (\mathsf{LSX}[C](A_1) \oplus A_2, A_1)\,,\ C, A_1, A_2 \in \mathbb{F}_{2^8}^n.$$

We define 3-round Kuznyechik as in [2]. Each round of the key schedule has only 2 rounds of basic cipher's Feistel rounds.

$$\begin{gathered}\mathsf{E}_{K_1,K_2}(A) = \mathsf{X}[K_4]\mathsf{LSX}[K_3]\mathsf{LSX}[K_2]\mathsf{LSX}[K_1](A), \\ (K_3, K_4) = \mathsf{F}[C_2]\mathsf{F}[C_1]\,(K_1, K_2) \\ K_3 = K_1 \oplus \mathsf{LSX}[C_2](K_2 \oplus \mathsf{LSX}[C_1](K_1)), \\ K_4 = K_2 \oplus \mathsf{LSX}[C_1](K_1).\end{gathered} \tag{2}$$

5-round Kuznyechik is defined in a similar way:

$$\begin{gathered}\mathsf{E}_{K_1,K_2}(A) = \mathsf{X}[K_6]\mathsf{LSX}[K_5]\mathsf{LSX}[K_4]\mathsf{LSX}[K_3]\mathsf{LSX}[K_2]\mathsf{LSX}[K_1](A), \\ (K_3, K_4) = \mathsf{F}[C_2]\mathsf{F}[C_1]\,(K_1, K_2)\,, \\ (K_5, K_6) = \mathsf{F}[C_4]\mathsf{F}[C_3]\,(K_3, K_4)\,.\end{gathered} \tag{3}$$

Denote also the block before addition of the key K_i by P_i (for example $P_2 = \mathsf{LSX}[K_1](A)$).

3 Technical lemmas and concepts

The polytopic cryptanalysis was first introduced in [3]. We will use some techniques from this concept along with integral cryptanalysis [4].

In particular, we use the difference (1) as «d-difference» in [3]. Let's consider how cipher transformations change this difference.

It's easy to see, that adding a *same* round key does not change the difference. The attack presented in section 4 uses non-equal keys. In this case, the difference between the round keys is added to the difference between the intermediate states. Note that if both such differences $\boldsymbol{\Delta}$, $\boldsymbol{\kappa} \in \left(\mathbb{F}_{2^8}^n\right)^d$ have integral property $\mathbf{0}$, then $\boldsymbol{\Delta} \oplus \boldsymbol{\kappa}$ has the same property.

Suppose that the difference $\boldsymbol{\Delta} \in \left(\mathbb{F}_{2^8}^n\right)^d$ has only one active position, then after the S-transformation we have no more than 2^8 possible differences. Indeed, all inactive positions remain inactive. We have one non-zero «column» $\boldsymbol{\alpha} = (c_1, c_2, \ldots, c_d) \in \mathbb{F}_{2^8}^d$ and after substitution layer:

$$\mathsf{s}(\boldsymbol{\alpha}) = \{(\mathsf{s}(x \oplus c_1) \oplus \mathsf{s}(x), \mathsf{s}(x \oplus c_2) \oplus \mathsf{s}(x), \ldots, \mathsf{s}(x \oplus c_d) \oplus \mathsf{s}(x)),\ x \in \mathbb{F}_{2^8}\},$$

where $\mathsf{s} : \mathbb{F}_{2^8} \to \mathbb{F}_{2^8}$ is cipher Sbox. Obviously, the number of differences $\mathsf{s}(\boldsymbol{\alpha})$ does not exceed the number of x. In most cases, these numbers are equal. If all bytes in $\boldsymbol{\alpha}$ are different, all bytes in $\mathsf{s}(\boldsymbol{\alpha})$ are also different (the bijective Sbox preserve the integral property $\mathbf{A}$). If we know $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}' \in \mathsf{s}(\boldsymbol{\alpha})$, we can easily find the corresponding x.

The L-transformation bijectively maps one difference to another:

$$\boldsymbol{\Delta} = (\Delta_1, \Delta_2, \ldots, \Delta_d),\ \mathsf{L}(\boldsymbol{\Delta}) = (\mathsf{L}(\Delta_1), \mathsf{L}(\Delta_2), \ldots, \mathsf{L}(\Delta_d)).$$

If only one position in input difference is active then all positions in output difference are active (this is true if L is MDS matrix). Under the same conditions, if one position has the property $\mathbf{A}$, then all output positions will have this property. The integral property $\mathbf{0}$ is preserved by L-transformation:

$$\bigoplus_{i=1}^{d} \Delta_i = 0,\ \bigoplus_{i=1}^{d} \mathsf{L}(\Delta_i) = \mathsf{L}\left(\bigoplus_{i=1}^{d} \Delta_i\right) = 0.$$

We will use the so-called integral property [4, 5] of LSXLSX transformation.

Lemma 1. *Let one position in the difference* $\boldsymbol{\Delta} \in \left(\mathbb{F}_{2^8}^n\right)^d$ *has integral property* $\boldsymbol{A}$ *and all other positions are inactive (so-called* δ*-set). Then any difference from* $\mathsf{LSXLSX}(\boldsymbol{\Delta})$ *has the integral property* $\boldsymbol{0}$*.*

Proof. Adding a round key does not change the difference. Thus, we have $\mathsf{LSLS}(\boldsymbol{\Delta})$. After the first substitution layer, one position will have the property $\mathbf{A}$ and all others will remain inactive. The first linear transformation will make all bytes active. Each of them will have the property $\mathbf{A}$. The second S transformation will preserve $\mathbf{A}$ and consequently the property $\mathbf{0}$. Hence, after the last linear transformation we have the property $\mathbf{0}$ in each position of the difference. □

Equivalent representation of the last two rounds

The presented attack uses an equivalent representation of the last two rounds.

Let $A, B \in \mathbb{F}_{2^8}^n$ be a plaintext and ciphertext correspondingly. $K_1, \ldots, K_r, K_{r+1}$ are round keys, $K_i \in \mathbb{F}_{2^8}^n,\ i = 1, 2, \ldots, r+1$.

The original cipher has the form

$$B = \mathsf{X}[K_{r+1}]\mathsf{LSX}[K_r] \ldots \mathsf{X}[K_1](A) = \mathsf{E}_{r+1}(A).$$

Apply the inverse linear transformation to the known ciphertext

$$\mathsf{L}^{-1}(B) = \mathsf{L}^{-1}(\mathsf{X}[K_{r+1}]\mathsf{LSX}[K_r] \ldots \mathsf{X}[K_1](A)),$$
$$\mathsf{L}^{-1}(B) = \mathsf{L}^{-1}(K_{r+1}) \oplus \mathsf{SX}[K_r] \ldots \mathsf{X}[K_1](A).$$

We denote $B' = \mathsf{L}^{-1}(B)$, $K_i' = \mathsf{L}^{-1}(K_i)$, then the cipher has the form

$$B' = \mathsf{X}[K_{r+1}']\mathsf{SX}[K_r]\mathsf{LSX}[K_{r-1}] \ldots \mathsf{X}[K_1](A).$$

Similarly, for the penultimate round. Let's consider the transformation

$$\mathsf{X}[K_r]\mathsf{L}(A) = K_r \oplus \mathsf{L}(A) = \mathsf{L}\left(A \oplus \mathsf{L}^{-1}(K_r)\right) = \mathsf{LX}[K_r'](A).$$

Therefore, the cipher transformation can be represented by the formula

$$B' = \mathsf{X}[K_{r+1}']\mathsf{SLX}[K_r']\mathsf{SX}[K_{r-1}] \ldots \mathsf{X}[K_1](A).$$

4 Related-key attack

Let's represent 5-round Kuznyechik (3) in equivalent form

$$\begin{aligned}
\mathsf{E}_{K_1,K_2}(A) &= \mathsf{X}[K_6^{'}]\mathsf{SLX}[K_5^{'}]\mathsf{SX}[K_4]\mathsf{LSX}[K_3]\mathsf{LSX}[K_2]\mathsf{LSX}[K_1](A),\\
(K_3,K_4) &= \mathsf{F}[C_2]\mathsf{F}[C_1]\,(K_1,K_2)\,,\\
K_4 &= K_2 \oplus \mathsf{LSX}[C_1](K_1),\\
K_3 &= K_1 \oplus \mathsf{LSX}[C_2](K_4),\\
(K_5,K_6) &= \mathsf{F}[C_4]\mathsf{F}[C_3]\,(K_3,K_4)\,,\\
K_6 &= K_4 \oplus \mathsf{LSX}[C_3](K_3),\ K_6^{'} = \mathsf{L}^{-1}(K_6),\\
K_5 &= K_3 \oplus \mathsf{LSX}[C_4](K_6),\ K_5^{'} = \mathsf{L}^{-1}(K_5).
\end{aligned}$$

The attack consists of the following steps:

1. Adversary chooses 2^8 collections of related keys, 2^8 keys in each collection. One plaintext C_1 (first constant in the key schedule) will be used.
2. For one of these collections, the special easy verifiable property (integral distinguisher) is true.
3. The round keys K_6, K_5 are recovered by using integral and polytopic properties.

Let's describe these steps in more detail. We denote

$$\boldsymbol{\kappa} = \left(\begin{array}{ccccc}
0 & 0 & \ldots & 0 & 1\\
0 & 0 & \ldots & 0 & 2\\
 & & \ddots & & \\
\ldots & \ldots & & \ldots & \ldots\\
\underbrace{0} & \underbrace{0} & \ldots & \underbrace{0} & \underbrace{255}\\
\multicolumn{5}{c}{n=16}
\end{array}\right)$$

the difference between keys K_1. The set $\mathsf{LS}(\boldsymbol{\kappa})$ contains 2^8 differences. The collection of the related keys looks like

$$(K_1,K_2) \text{ and set } (K_1 \oplus \boldsymbol{\kappa}, K_2 \oplus \boldsymbol{\kappa}''), \text{ where } \boldsymbol{\kappa}'' \in \mathsf{LS}(\boldsymbol{\kappa}).$$

It is easy to see that each collection contains the «main» key and a set of 255 related keys. Adversary does not know the keys, but it know all relations ($\boldsymbol{\kappa}$

and $\boldsymbol{\kappa}'' \in \mathsf{LS}(\boldsymbol{\kappa})$) between them. Adversary encrypts only one plaintext C_1 and gets 2^8 ciphertexts for each collection of keys. In total we have $1 + 2^8 \cdot \left(2^8 - 1\right)$ different keys and different ciphertexts correspondingly. In the same collection we refer to the difference between i-th round keys K_i as $\boldsymbol{\Delta K_i}$, for example $\boldsymbol{\kappa} = \boldsymbol{\Delta K_1}$ and $\boldsymbol{\kappa}'' = \boldsymbol{\Delta K_2}$.

4.1 Integral property

Figure 1 shows the propagation of differences, which is true for only one collection of keys (for only one $\boldsymbol{\kappa}'' \in \mathsf{LS}(\boldsymbol{\kappa})$). Active Sboxes have a gray background. Integral properties are indicated in red bold (**A** – all bytes are different, **0** – bitwise xor of all bytes is zero). More detailed pictures are presented in Appendix B.

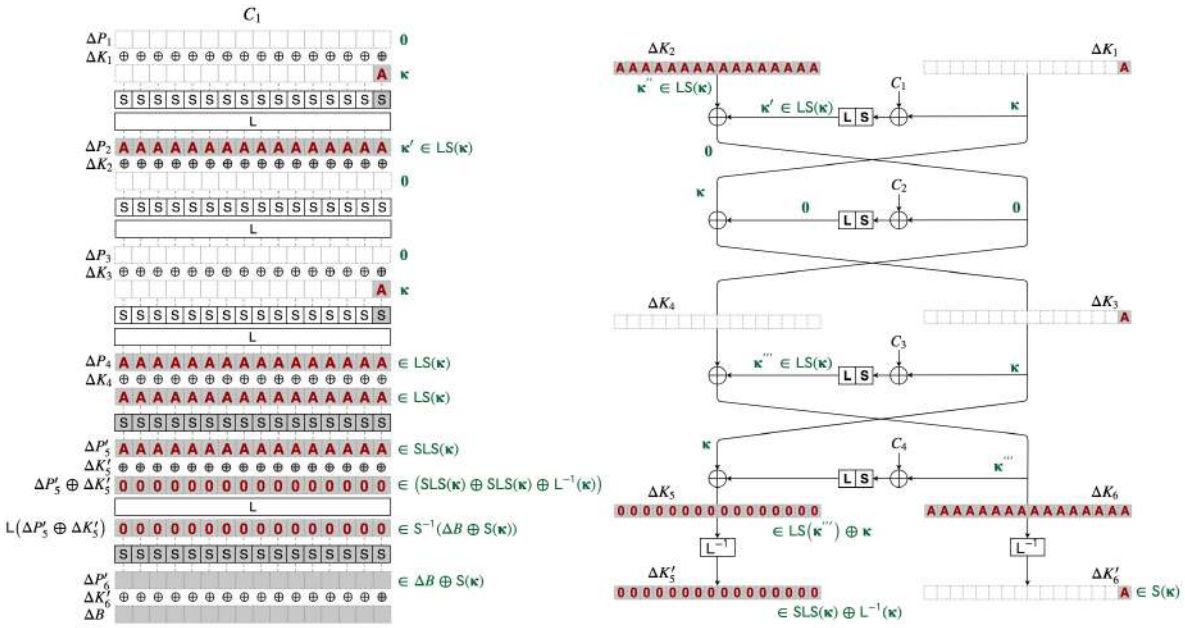


Figure 1: Related-key attack on 5-round Kuznyechik

Let's see the key schedule. After the first $\mathsf{LS}(C_1 \oplus K_1)$ transformation we have difference $\boldsymbol{\kappa}' \in \mathsf{LS}(\boldsymbol{\kappa})$. This difference is the same for all collections of keys, but only for one $\boldsymbol{\kappa}'' = \boldsymbol{\kappa}'$ is true. If so, we have $\boldsymbol{\Delta K_4} = \mathbf{0}$ and $\boldsymbol{\Delta K_3} = \boldsymbol{\kappa}$. The difference $\boldsymbol{\kappa}$ has one active byte. In the difference $\boldsymbol{\kappa}'' \in \mathsf{LS}(\boldsymbol{\kappa})$, all bytes are active and have an integral property **A**.

Similarly,

$$\boldsymbol{\Delta K_6} = \boldsymbol{\kappa'''} \in \mathsf{LS}(\boldsymbol{\kappa})$$

and

$$\boldsymbol{\Delta K_5} \in \mathsf{LS}(\boldsymbol{\kappa'''}) \oplus \boldsymbol{\kappa} = \{\boldsymbol{\delta} \oplus \boldsymbol{\kappa},\ \boldsymbol{\delta} \in \mathsf{LS}(\boldsymbol{\kappa'''})\}.$$

We use the equivalent representation of the last two rounds, therefore we consider difference between the keys $K_5^{'} = \mathsf{L}^{-1}(K_5)$ instead K_5, and $K_6^{'} = \mathsf{L}^{-1}(K_6)$ instead K_6. Thus we obtain that $\boldsymbol{\Delta K_6^{'}}$ belongs to $\mathsf{S}(\boldsymbol{\kappa})$ (correspondingly $\boldsymbol{\Delta K_5^{'}} \in \mathsf{SLS}(\boldsymbol{\kappa}) \oplus \mathsf{L}^{-1}(\boldsymbol{\kappa})$).

All bytes of $\boldsymbol{\Delta K_5^{'}}$ are active and have an integral property $\mathbf{0}$ (see lemma 1). The difference $\boldsymbol{\Delta K_6^{'}}$ has one active byte with the property $\mathbf{A}$.

Now let's consider the encryption functions. We use only one plaintext C_1, therefore the difference $\boldsymbol{\Delta P_1}$ is equal to zero. Note that the first round of encryption also has the form $\mathsf{LS}(C_1 \oplus K_1)$. Because of this, the difference between the blocks is also equal to $\boldsymbol{\kappa'}$.

If in the key schedule $\boldsymbol{\kappa'} = \boldsymbol{\kappa''} = \boldsymbol{\Delta K_2}$ then the difference between the blocks becomes zero and also $\boldsymbol{\Delta P_3} = \mathbf{0}$.

The addition of the third round key K_3 adds the non-zero difference $\boldsymbol{\kappa}$. We do not know the exact value of the difference $\boldsymbol{\Delta P_4}$, but we know that $\boldsymbol{\Delta P_4}$ belongs to the set $\mathsf{LS}(\boldsymbol{\kappa})$. Similarly, after the following substitution layer, we have $\boldsymbol{\Delta P_5^{'}} \in \mathsf{SLS}(\boldsymbol{\kappa})$. All bytes of $\boldsymbol{\Delta P_5^{'}}$ have an integral property $\mathbf{A}$ and consequently property $\mathbf{0}$. The second one is also true for $\boldsymbol{\Delta K_5^{'}}$. Therefore, their sum $\boldsymbol{\Delta P_5^{'}} \oplus \boldsymbol{\Delta K_5^{'}}$ has an integral property $\mathbf{0}$. The linear transformation preserves this one.

Obviously, we know the corresponding ciphertexts and the difference $\boldsymbol{\Delta B}$ between them. The difference $\boldsymbol{\Delta K_6^{'}} \in \mathsf{S}(\boldsymbol{\kappa})$ has one active byte.

Let's propagate the difference through S^{-1}. For each of 15 Sboxes we have 2^8 possible differences and for 16'th Sbox we get 2^{16} differences due to $\boldsymbol{\Delta K_6^{'}} \in \mathsf{S}(\boldsymbol{\kappa})$.

Let's check the integral property $\mathbf{0}$ for each obtained difference. If we correctly guessed $\boldsymbol{\kappa''}$, then there must be at least one such difference for each Sbox. Otherwise, if we do not guess it correctly, then there is at least one Sbox for which there is no such difference. Generally speaking, it is possible that a «false» collection of the related keys will satisfy this property. The probability of the

existence of the such «false» collection is approximately 0.23 (for more details see Appendix A). It does not lead to the failure of the attack. We will be able to distinguish this case through the next step.

We also expect that for each of 15 Sboxes about 2 differences have integral property $\mathbf{0}$. For the last Sbox about 2^8 differences have such property. Thus, the set $\mathsf{S}^{-1}(\boldsymbol{\Delta B} \oplus \mathsf{S}(\boldsymbol{\kappa}))$ will contain about $2^{15} \cdot 2^8 = 2^{23}$ possible differences, each of them has the property $\mathbf{0}$.

4.2 Recovering of the round keys

Let's consider the last linear transformation. We know that $\boldsymbol{\Delta P}_{\mathbf{5}}^{'} \in \mathsf{SLS}(\boldsymbol{\kappa})$ and $\boldsymbol{\Delta K}_{\mathbf{5}}^{'} \in \mathsf{SLS}(\boldsymbol{\kappa}) \oplus \mathsf{L}^{-1}(\boldsymbol{\kappa})$. The difference before the linear transformation is the sum

$$\boldsymbol{\Delta P}_{\mathbf{5}}^{'} \oplus \boldsymbol{\Delta K}_{\mathbf{5}}^{'} \in (\mathsf{SLS}(\boldsymbol{\kappa}) \oplus \mathsf{SLS}(\boldsymbol{\kappa}) \oplus \mathsf{L}^{-1}(\boldsymbol{\kappa})) = \\ \left\{\boldsymbol{\delta_1} \oplus \boldsymbol{\delta_2} \oplus \mathsf{L}^{-1}(\boldsymbol{\kappa}),\, \boldsymbol{\delta_1} \in \mathsf{SLS}(\boldsymbol{\kappa}),\, \boldsymbol{\delta_2} \in \mathsf{SLS}(\boldsymbol{\kappa})\right\}.$$

On the other hand we have the set of possible differences $\mathsf{S}^{-1}(\boldsymbol{\Delta B} \oplus \mathsf{S}(\boldsymbol{\kappa}))$ after the linear transformation.

The intersection of sets

$$(\mathsf{SLS}(\boldsymbol{\kappa}) \oplus \mathsf{SLS}(\boldsymbol{\kappa}) \oplus \mathsf{L}^{-1}(\boldsymbol{\kappa})) \cap \mathsf{L}^{-1}\mathsf{S}^{-1}(\boldsymbol{\Delta B} \oplus \mathsf{S}(\boldsymbol{\kappa}))$$

must contain at least one element. We use only one byte position to determine the inequality of elements from these two sets.

After checking the integral property in the set $\mathsf{L}^{-1}\mathsf{S}^{-1}(\boldsymbol{\Delta B} \oplus \mathsf{S}(\boldsymbol{\kappa}))$ there will be about 2^{23} possible differences.

Recall that the set $\mathsf{S}(\boldsymbol{\kappa})$ contains 2^8 elements. The linear transformation does not change the number of differences ($\mathsf{LS}(\kappa)$ contains 2^8 elements). After another substitution layer we have 2^{16} possible differences at each Sbox. The difference $\boldsymbol{\kappa}$ is known, therefore $\mathsf{L}^{-1}(\boldsymbol{\kappa})$ is also known. Consequently, the set $\mathsf{SLS}(\boldsymbol{\kappa}) \oplus \mathsf{SLS}(\boldsymbol{\kappa}) \oplus \mathsf{L}^{-1}(\boldsymbol{\kappa})$ contains

$$\frac{2^{16} \cdot (2^{16} - 1)}{2} + 1 < 2^{31}$$

possible differences at each Sbox.

Select the position of one of the block bytes. Recall also that each difference contains $2^8 - 1$ vectors and consequently difference in one position

contains $2^8 - 1$ bytes. We store in memory all possible differences from $\mathsf{SLS}(\boldsymbol{\kappa}) \oplus \mathsf{SLS}(\boldsymbol{\kappa}) \oplus \mathsf{L}^{-1}(\boldsymbol{\kappa})$ for selected position. Let's iterate through all differences $\boldsymbol{\gamma}$ in $\mathsf{L}^{-1}\mathsf{S}^{-1}(\boldsymbol{\Delta B} \oplus \mathsf{S}(\boldsymbol{\kappa}))$. If $\boldsymbol{\gamma}$ matches one of the stored differences then we assume that $\boldsymbol{\gamma} = \boldsymbol{\Delta P}_{\mathbf{5}}^{'} \oplus \boldsymbol{\Delta K}_{\mathbf{5}}^{'}$. We can expect that $\boldsymbol{\gamma}$ is the only such element even if we compare on eight bytes of a difference rather than $2^8 - 1$. Note that if the collection of the related keys is «false» ($\boldsymbol{\kappa}'' \neq \boldsymbol{\kappa}'$), the match will probably not be found.

At this step we know $\boldsymbol{\Delta P}_{\mathbf{5}}^{'} \oplus \boldsymbol{\Delta K}_{\mathbf{5}}^{'}$, $\mathsf{L}(\boldsymbol{\Delta P}_{\mathbf{5}}^{'} \oplus \boldsymbol{\Delta K}_{\mathbf{5}}^{'})$, $\boldsymbol{\Delta P}_{\mathbf{6}}^{'}$, $\boldsymbol{\Delta K}_{\mathbf{6}}^{'}$, $\boldsymbol{\Delta B}$. Block

$$Y : \mathsf{S}^{-1}(\boldsymbol{\Delta P}_{\mathbf{6}}^{'}) = \mathsf{L}(\boldsymbol{\Delta P}_{\mathbf{5}}^{'} \oplus \boldsymbol{\Delta K}_{\mathbf{5}}^{'})$$

can be easily found. Let B_0 be first ciphertext, then $K_6^{'} = B_0 \oplus Y$. The entire set of related keys $K_6^{'}$ can also be obtained by adding with $\boldsymbol{\Delta K}_{\mathbf{6}}^{'}$. Therefore, it is possible to decipher all 2^8 ciphertexts through the last round.

We know that $\boldsymbol{\Delta P_4} \in \mathsf{LS}(\boldsymbol{\kappa})$, $\boldsymbol{\Delta K}_{\mathbf{5}}^{'} \in \mathsf{S}(\boldsymbol{\Delta K}_{\mathbf{6}}^{'}) \oplus \mathsf{L}^{-1}(\boldsymbol{\kappa})$ and also $\boldsymbol{\Delta P}_{\mathbf{5}}^{'} \oplus \boldsymbol{\Delta K}_{\mathbf{5}}^{'}$. Let's iterate through possible $\boldsymbol{\tau} \in \mathsf{S}(\boldsymbol{\Delta K}_{\mathbf{6}}^{'}) \oplus \mathsf{L}^{-1}(\boldsymbol{\kappa})$ and propagate $\boldsymbol{\Delta P}_{\mathbf{5}}^{'} \oplus \boldsymbol{\Delta K}_{\mathbf{5}}^{'} \oplus \boldsymbol{\tau}$ through S^{-1}. If we guess $\boldsymbol{\tau} = \boldsymbol{\Delta K}_{\mathbf{5}}^{'}$, then $\mathsf{S}^{-1}(\boldsymbol{\Delta P}_{\mathbf{5}}^{'} \oplus \boldsymbol{\Delta K}_{\mathbf{5}}^{'} \oplus \boldsymbol{\tau}) = \mathsf{S}^{-1}(\boldsymbol{\Delta P}_{\mathbf{5}}^{'}) \in \mathsf{LS}(\boldsymbol{\kappa})$. Otherwise, we expect that $\mathsf{S}^{-1}(\boldsymbol{\Delta P}_{\mathbf{5}}^{'} \oplus \boldsymbol{\Delta K}_{\mathbf{5}}^{'} \oplus \boldsymbol{\tau}) \notin \mathsf{LS}(\boldsymbol{\kappa})$. In the matching process, each Sbox can be viewed independently of the others. After that we will know the differences $\boldsymbol{\Delta P_4}$, $\boldsymbol{\Delta P}_{\mathbf{5}}^{'}$, $\boldsymbol{\Delta K}_{\mathbf{5}}^{'}$. The ciphertexts after 5'th round are also known. Therefore, the keys $K_5^{'}$ can be found in the same way as $K_6^{'}$. Due to the reverse key schedule, the master key $K = K_1 || K_2$ can be easily obtained.

4.3 Complexity

As mentioned before, the attack requires $1 + 2^8 \cdot \left(2^8 - 1\right) < 2^{16}$ related keys and one chosen ciphertext.

The integral property for all 2^8 related key collections can be checked in about $2^8 \cdot (15 \cdot 2^8 + 2^{16}) \approx 2^{24}$ operations.

The most time-consuming stage is the construction of the set $\mathsf{SLS}(\kappa) \oplus \mathsf{SLS}(\kappa) \oplus \mathsf{L}^{-1}(\boldsymbol{\kappa})$. We construct this set for only one Sbox, and store only eight bytes for each difference. It requires about 2^{31} operations and $2^{31} \cdot 8 = 2^{34}$ bytes of memory. These constructed differences are stored in a hash table. The set $\mathsf{L}^{-1}\mathsf{S}^{-1}(\boldsymbol{\Delta B} \oplus \mathsf{S}(\boldsymbol{\kappa}))$ contains much fewer elements. Checking for a single element in a hash table requires constant time. Therefore, the complexity of constructing the hash table will be the most important. The difficulty of recovering the keys

K_5' is also small: $16 \cdot 2^8 \cdot 2^8 + 2^8 \approx 2^{20}$ operations.

The total complexity does not exceed 2^{32} memory access operations and 2^{30} memory (in sixteen-byte blocks). We also note that the attack is deterministic.

We modeled the attack with a non-optimized C++ implementation. The average attack time is about 5 minutes on a common PC. The amount of used memory did not exceed 17 GB.

5 Conclusion

In this paper we present the related-key attack on 5-round Kuznyehcik with 2-round key schedule. The attack has a practical complexity (2^{32} operations, 2^{30} memory, 2^{16} related keys) and has been verified with the help of C++ implementation. The experiments confirmed the correctness of the attack.

Source codes can be found at `https://gitlab.com/v.kir/rk-5R-kuznyechik`.

The main result was achieved by using the well-known integral property of LSX-transformations. We were able to use this property both in the cipher itself and in the key schedule.

We did not use any specific properties of the linear transformation and the Sbox. We think that through the use of such properties it is possible to obtain new results. Another possible way is the use of integral distinguishers for a greater number of rounds.

The presented attack also shows a significant security margin of the Kuznyechik's key schedule.

Acknowledgements

The author is grateful to Sergey Svetlov for help with experiments and verification, to Anton Naumenko and Igor Arbekov for support and valuable comments.

References

[1] *GOST R 34.12-2015. National standard of the Russian Federation. Information technology Cryptographic data security Block ciphers*, 2015, in Russian.

[2] E. Alekseev, K. Goncharenko, and Grigory Marshalko, "Provably Secure Counter Mode with Related Key-based Internal Re-keying", *Pre-proceedings*, 7th Workshop on Current Trends in Cryptology (CTCrypt 2018), 2018.

[3] T. Tiessen, "Polytopic Cryptanalysis", *LNCS*, EUROCRYPT 2016, **9665**, ed. Fischlin M., Coron J.S., Springer, Berlin, Heidelberg.

[4] Daemen J., Knudsen L., and Rijmen V., "The block cipher Square", *LNCS*, FSE 1997, **1267**, ed. Biham E., Springer, Berlin, Heidelberg, 1997.

[5] Barreto P. and Rijmen V., "The Khazad legacy-level block cipher", First open NESSIE Workshop, 2000, Submission to NESSIE.

[6] Biham E., "New types of cryptoanalytic attacks using related keys (extended abstract)", *LNCS*, EUROCRYPT 93, **765**, ed. Helleseth T., Springer, Berlin, Heidelberg, 1993.

A Probability aspects and experimental verification

«True» and «false» collections of the related keys

We know that there is at least one «true» collection. What is the probability that the integral property (section 4.1) will be correct for the «false» collection?

Assume that all ciphertexts are equally probable and independent of each other. We propagate the difference of each Sbox thorough nonlinear layer. For each of 15 Sboxes we'll have 2^8 possible differences and for 16'th Sbox we get 2^{16} differences. We also assume that the sum of the elements of any difference is uniformly distributed. Hence, the probability of the property **0** is equal to $p = \frac{1}{256}$ for each difference of any Sbox. Denote the probability of the opposite event by $q = 1 - p = \frac{255}{256}$.

Thus, we have:

$q^{2^8} = 0.367...$ – there is no difference that has the property **0** for one Sbox;

$1 - q^{2^8} = 0.632...$ – there is at least one such difference;

$(1 - q^{2^8})^{15} = 0.001...$ – there is at least one such difference for each of the 15 Sboxes.

The probability that one collection of the related keys has the integral property is

$$r = \left(1 - q^{2^8}\right)^{15} \cdot \left(1 - q^{2^{16}}\right) = 0.001...$$

We have 2^8 collections of keys and only one «true» collection. The probability that «false» collections do not exist is

$$(1 - r)^{255} = 0.765....$$

The opposite probability is

$$1 - (1 - r)^{255} = 0.234....$$

We performed $N = 5000$ experiments. The number of cases where collections exist is equal to 1179. The obtained value $\dfrac{1179}{5000} = 0.236$ is close to theoretical.

Number of possible differences

Let we have «true» collection of the related keys. We estimate the number of possible differences in the set $\mathsf{L}^{-1}\mathsf{S}^{-1}(\boldsymbol{\Delta B} \oplus \mathsf{S}(\boldsymbol{\kappa}))$.

Each Sbox gives at least one possible difference. The probability of the property $\mathbf{0}$ is equal to $p = \frac{1}{256}$ for each difference of any Sbox. We also have 2^8 possible differences for each of 15 Sboxes and 2^{16} for 16'th Sbox.

Thus, average number of elements in the set is equal to

$$\left(1 + \frac{1}{256} \cdot (2^8 - 1)\right)^{15} \cdot \left(1 + \frac{1}{256} \cdot (2^{16} - 1)\right) \approx 2^{23} \ll 2^{31}.$$

The average experimental value is $2^{22.7}$. The maximum value among all N experiments is 2^{29}.

Matching differences

The intersection of sets

$$(\mathsf{SLS}(\kappa) \oplus \mathsf{SLS}(\kappa) \oplus \mathsf{L}^{-1}(\boldsymbol{\kappa})) \cap \mathsf{L}^{-1}\mathsf{S}^{-1}(\boldsymbol{\Delta B} \oplus \mathsf{S}(\boldsymbol{\kappa}))$$

must contain at least one element. We use only one position to determine the inequality of elements from these two sets.

One position of the first set contains no more than 2^{31} differences. The number of elements of the second set is approximately 2^{23}. We also assume that the elements of these sets are random and equally probable.

Only the first 8 bytes (64 bits) of the difference are stored in memory. Then the average number of «false» matches can be estimated as

$$\frac{2^{31} \cdot 2^{23}}{2^{64}} = 2^{-10}.$$

A «false» match can be easily detected by an additional check. In $N = 5000$

experiments, we got only 7 cases of it.

Eight-byte numbers were chosen for ease of implementation.

B Detailed pictures

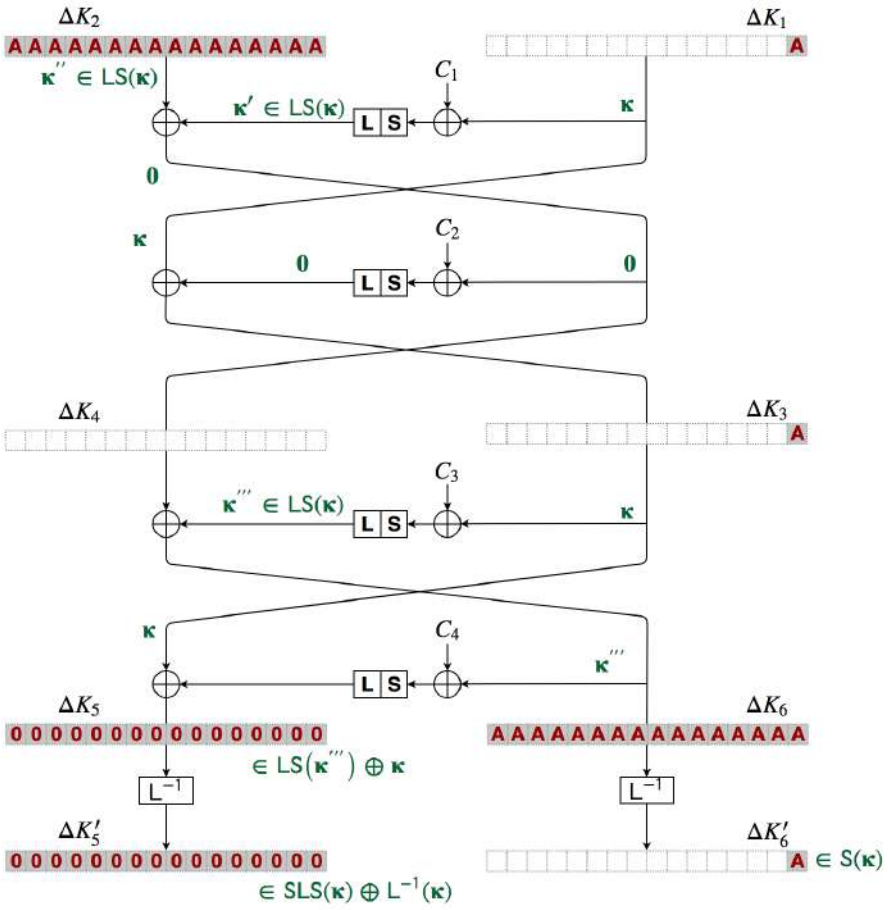


Figure 2: The difference propagation through the key schedule

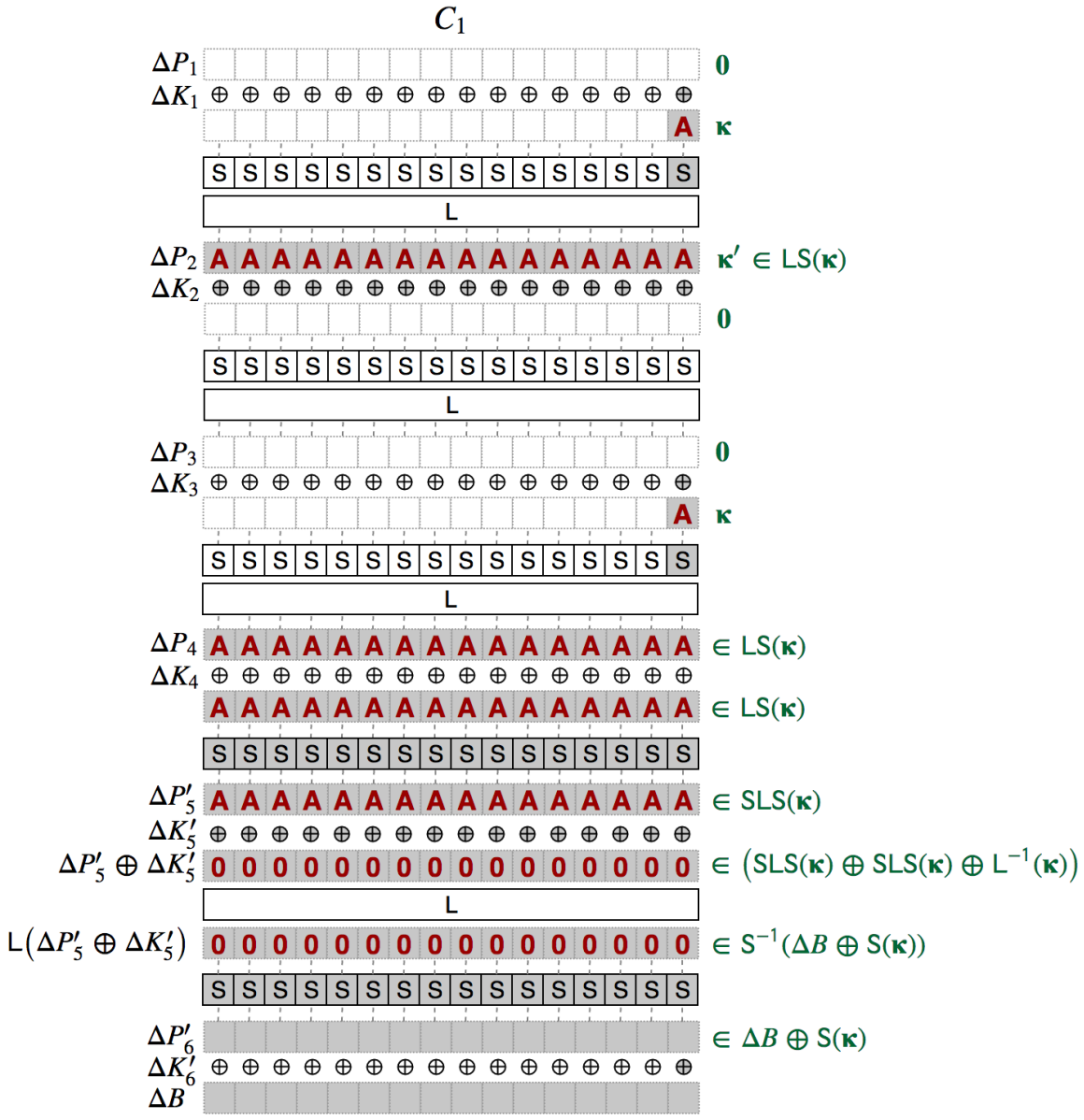


Figure 3: The difference propagation through the cipher

The Change in Linear and Differential Characteristics of Substitution Multiplied by Transposition

Andrey Menyachikhin

TVP Laboratories, Russia
and88@list.ru

Abstract

It is shown that the linearity and differential uniformity of the substitution multiplied by transposition can be calculated with time complexities $O\left(2^{2n}\right)$ and $O\left(2^{n}\right)$ respectively. Some heuristic algorithms of constructing s-boxes are optimized in this paper.

Keywords: s-box, substitution, heuristic optimization, linearity, linear spectrum, linear approximation table, differential uniformity, differential spectrum, difference distribution table.

1 Introduction

Constructing s-boxes with excellent cryptographic properties is one of the important problems in modern cryptography. One approach to solve this problem is based on heuristic optimization of some given s-box. The heuristic optimization methods include genetic algorithms (see [3, 9]), hill climbing methods (see [7, 8]), methods of gradient descent (see [6]), spectral-linear and spectral-differential methods (see [4, 5]).

The main problem of using heuristic methods is the high level of their time complexity. The δ_g-parameter and the p_g-parameter of s-box g are the most difficult to calculate. In this paper we introduce new techniques for calculating linearity and differential uniformity of the substitution $h \in S\left(V_n\right)$ such that $h = (x, y)\, g$ where $x, y \in V_n$, $g \in S\left(V_n\right)$.

The rest of the paper is organized as follows. Section 2 gives the basic definitions and notations. In section 3 we derive the main propositions and algorithms. The cryptographic applications of the results are discussed in Section 4. Finally, Section 5 presents our conclusions.

2 Definitions and Notations

Let $V_n(2) = V_n$ be n-dimensional vector space over the field $\mathbb{F}_2$. Suppose that $V_n^{\times} = V_n \backslash \{0\}$. Let $S(V_n)$ be the symmetric group on set of 2^n elements. The cardinality of a set A is usually denoted $|A|$.

We shall use the following operations and notations:

- exclusive-OR (or XOR) is denoted $\oplus$,
- logical AND is denoted $\wedge$,
- the scalar product of two elements $x = (x_{n-1}, ..., x_0)$ and $y = (y_{n-1}, ..., y_0)$ of $\mathbb{F}_2^n$ is denoted $\circ$ and is equal to $x \circ y = \oplus_{i=0}^{n-1} x_i \wedge y_i$

Now, we give some basic definitions.

Definition 1. *The* linearity *of s-box g is defined as the absolute value of the bias:*

$$\delta_g = \max_{\alpha,\beta \in V_n^{\times}} \delta_{\alpha,\beta}^g$$

where

$$\delta_{\alpha,\beta}^g = 2^{1-n} \cdot |\{x \in V_n \,|x \circ \alpha = g(x) \circ \beta\}|.$$

The Linear Approximation Table *(LAT) of s-box g is a $2^n \times 2^n$ matrix T_1 such that $T_1(\alpha, \beta) = \delta_{\alpha,\beta}^g$.*

S-boxes with small value of δ_g-parameter offer better resistance against linear attacks.

Definition 2. *The* differential uniformity *of s-box g is defined as*

$$p_g = \max_{\alpha,\beta \in V_n^{\times}} p_{\alpha,\beta}^g,$$

where

$$p_{\alpha,\beta}^g = 2^{-n} \cdot |\{x \in V_n \,|g(x \oplus \alpha) \oplus g(x) = \beta\}|.$$

The Difference Distribution Table *(DDT) of s-box g is a $2^n \times 2^n$ matrix T_2 such that $T_2(\alpha, \beta) = p_{\alpha,\beta}^g$.*

S-boxes using in cryptographic primitives must have a low p_g-parameter value to provide high resistance to differential cryptanalysis.

According to [4] we define the linear and the differential spectra of substitution g.

For $g \in S(V_n)$ and for elements $p \in P_{n-1}$ and $\delta \in P_{n-2}$, where

$$P_j = \left\{ \frac{i}{2^j} \left| i = 0, 1, ..., 2^j \right. \right\}, |P_j| = 2^j + 1, j \in \{n-2, n-1\};$$

we define the sets

$$D(g,p) = \left\{ (\alpha, \beta) \in V_n^{\times} \times V_n^{\times} \left| p_{\alpha,\beta}^g = p \right. \right\},$$

$$L(g,\delta) = \left\{ (\alpha, \beta) \in V_n^{\times} \times V_n^{\times} \left| \left| \delta_{\alpha,\beta}^g \right| = \delta \right. \right\}.$$

Definition 3. *The* differential spectrum *of s-box g is defined as*

$$D(g) = \{(p, \ |D(g,p)|) \, | p \in P_{n-1}\}, |D(g)| = 2^{n-1} + 1.$$

Definition 4. *The* linear spectrum *of s-box g is defined as*

$$L(g) = \{(\delta, |L(g,\delta)|) \, | \delta \in P_{n-2}\}, |L(g)| = 2^{n-2} + 1.$$

3 Main results

This section deals with the change in linear and differential characteristics of substitution multiplied by transposition. This issue has been studied in [10]. The authors showed that for $h = (x,y)\, g$ such that $g, h : V_n \to V_n$ we get:

$$\delta_g - 2^{2-n} \leqslant \delta_h \leqslant \delta_g + 2^{2-n},$$

$$p_g - 2^{2-n} \leqslant p_h \leqslant p_g + 2^{2-n}.$$

In [7] the similar properties of boolean functions are used to optimize the hill climbing methods.

In this section we describe two new algorithms for calculating linearity and differential uniformity of substitution $h \in S(V_n)$ such that $h = (x,y)\, g$. We also formally prove the correctness of the new algorithms and study their time complexity.

3.1 Efficient computation of the elements in linear approximation table

The first subsection deals with the relationship between elements of linear approximation tables for the substitutions $g, h \in S(V_n)$ such that $h = (x, y)\, g$.

Algorithm 1.

Input. Substitution $g \in S(V_n)$; the elements $x, y \in V_n$; the LAT $T_1(g)$; the linear spectrum $D(g)$.

Step 1. For each element $i = 0, ..., n-1$ do the following items:

- calculate elements $\alpha = x \oplus y$ and $\beta = g(x) \oplus g(y)$;
- if $\alpha \circ i > 0$ then add i to the list I_1;
- if $\beta \circ i > 0$ then add i to the list I_2.

Step 2. For each ordered pair $(\alpha, \beta) \in I_1 \times I_2$ do the following items:

- calculate $\left|L\left(g, \left|\delta^g_{\alpha,\beta}\right|\right)\right| = \left|L\left(g, \left|\delta^g_{\alpha,\beta}\right|\right)\right| - 1$;
- calculate value $\delta^g_{\alpha,\beta} = \delta^g_{\alpha,\beta} + (-1)^{\alpha \circ x \oplus \beta \circ g(x) \oplus 1} \cdot 2^{2-n}$;
- calculate $\left|L\left(g, \left|\delta^g_{\alpha,\beta}\right|\right)\right| = \left|L\left(g, \left|\delta^g_{\alpha,\beta}\right|\right)\right| + 1$.

Step 3. The algorithm stops after calculating $h = (x, y)\, g$ and $D(h) = D(g)$.

Output. Substitution $h \in S(V_n)$ such that $h = (x, y)\, g$; the linear spectrum $L(h)$.

The correctness of the algorithm 1 is presented in the first proposition.

Proposition 1. *For substitutions $g, h \in S(V_n)$ such that $h = (x, y)\, g$ we have*

$$\delta^h_{\alpha,\beta} - \delta^g_{\alpha,\beta} = \begin{cases} 0, \text{ if either } (x \oplus y) \circ \alpha = 0 \text{ or } (g(x) \oplus g(y)) \circ \beta = 0 \\ (-1)^{\alpha \circ x \oplus \beta \circ g(x) \oplus 1} \cdot 2^{2-n} \text{ in the converse case} \end{cases}.$$

Proof. Consider the following sequence of equations

$$\delta^h_{\alpha,\beta} - \delta^g_{\alpha,\beta} = 2P\{z \circ \alpha = h(z) \circ \beta\} - 2P\{z \circ \alpha = g(z) \circ \beta\} =$$

$$= (|\{z \,|z \circ \alpha = h(z) \circ \beta\}| - |\{z \,|z \circ \alpha = g(z) \circ \beta\}|) \cdot 2^{1-n} =$$

$$= \left(\sum_{z \in V_n} (1 - z \circ \alpha \oplus h(z) \circ \beta) - \sum_{z \in V_n} (1 - z \circ \alpha \oplus g(z) \circ \beta) \right) \cdot 2^{1-n} =$$

$$= \big(x \circ \alpha \oplus g(x) \circ \beta - x \circ \alpha \oplus g(y) \circ \beta +$$

$$+ y \circ \alpha \oplus g(y) \circ \beta - y \circ \alpha \oplus g(x) \circ \beta \big) \cdot 2^{1-n}.$$

It is easily shown that if $x \circ \alpha = y \circ \alpha$ or $g(x) \circ \beta = g(y) \circ \beta$ then

$$\delta^h_{\alpha,\beta} - \delta^g_{\alpha,\beta} = 0.$$

Suppose that $\left\{ \begin{array}{l} x \circ \alpha \neq y \circ \alpha \\ g(x) \circ \beta \neq g(y) \circ \beta \end{array} \right. \left(\left\{ \begin{array}{l} y \circ \alpha = (x \circ \alpha) \oplus 1 \\ g(y) \circ \beta = (g(x) \circ \beta) \oplus 1 \end{array} \right. \right)$.
Note that in this case we have

$$\left(\underbrace{x \circ \alpha \oplus g(x) \circ \beta - x \circ \alpha \oplus g(x) \circ \beta \oplus 1}_{(-1)^{x \circ \alpha \oplus g(x) \circ \beta \oplus 1}} + \right.$$

$$\left. + \underbrace{x \circ \alpha \oplus g(x) \circ \beta - x \circ \alpha \oplus g(x) \circ \beta \oplus 1}_{(-1)^{x \circ \alpha \oplus g(x) \circ \beta \oplus 1}} \right) \cdot 2^{1-n} =$$

$$= (-1)^{x \circ \alpha \oplus g(x) \circ \beta \oplus 1} \cdot 2^{2-n}.$$

This completes the proof. □

Let us denote by t_1 the complexity of an algorithm 1.

Proposition 2. *As $n \to \infty$ we obviously have*

$$t_1 = O\left(2^{2n}\right).$$

Proof. We see that $|I_1| = |I_2| = 2^{n-1}$. The complexity of algorithm is the dot product of the following values.

– the number of the iterations of step 2 is $|I_1| \cdot |I_2| = 2^{2n-2}$;

– constant number of operations on the step 2 of the algorithm.

□

Remark 1. *The algorithm 1 is nearly n times faster than the classical algorithm of calculating the linearity.*

Example 1. *Consider the following substitutions* $g, h \in S(V_3)$ *such that* $h = (1,2)\, g$*:*

$$g = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 2 & 4 & 1 & 0 & 3 & 5 & 6 & 7 \end{pmatrix}, h = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 2 & 1 & 4 & 0 & 3 & 5 & 6 & 7 \end{pmatrix}.$$

The results obtained by algorithm 1 are presented in Table 1 (all changing elements of the matrix are in bold).

In particular, using preposition 1, we obtain

1. *Since* $(1 \oplus 2) \circ 3 = 0$ *we see that for any* $\beta \in V_3$ *it follows that*

$$\delta^h_{3,\beta} - \delta^g_{3,\beta} = 0;$$

2. $\delta^h_{1,1} - \delta^g_{1,1} = (-1)^{(1\circ 1)\oplus(1\circ 4)\oplus 1} \cdot 2^{2-3} = 1/2$;

3. $\delta^h_{2,3} - \delta^g_{2,3} = (-1)^{(2\circ 1)\oplus(3\circ 4)\oplus 1} \cdot 2^{2-3} = -1/2$.

Table 1.

Substitution g								*Substitution h*							
2 4 1 0 3 5 6 7								2 1 4 0 3 5 6 7							
LAT of g								*LAT of h*							
1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	**0**	$-\frac{1}{2}$	$-\mathbf{\frac{1}{2}}$	$\mathbf{\frac{1}{2}}$	$-\frac{1}{2}$	**0**	0	0	$\mathbf{\frac{1}{2}}$	$-\frac{1}{2}$	**0**	**0**	$-\frac{1}{2}$	$-\mathbf{\frac{1}{2}}$	0
0	**0**	0	**0**	**0**	0	**−1**	0	0	$-\mathbf{\frac{1}{2}}$	0	$-\mathbf{\frac{1}{2}}$	$\mathbf{\frac{1}{2}}$	0	$-\mathbf{\frac{1}{2}}$	0
0	0	$-\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0	0	$-\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	0
0	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0	$-\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0	$-\frac{1}{2}$
0	$-\mathbf{\frac{1}{2}}$	0	$-\mathbf{\frac{1}{2}}$	**0**	$\frac{1}{2}$	**0**	$-\frac{1}{2}$	0	**0**	0	**0**	$-\mathbf{\frac{1}{2}}$	$\frac{1}{2}$	$-\mathbf{\frac{1}{2}}$	$-\frac{1}{2}$
0	$\mathbf{\frac{1}{2}}$	$-\frac{1}{2}$	**0**	$-\mathbf{\frac{1}{2}}$	0	**0**	$-\frac{1}{2}$	0	**0**	$-\frac{1}{2}$	$-\mathbf{\frac{1}{2}}$	**0**	0	$\mathbf{\frac{1}{2}}$	$-\frac{1}{2}$
0	$\frac{1}{2}$	0	$-\frac{1}{2}$	0	$\frac{1}{2}$	0	$\frac{1}{2}$	0	$\frac{1}{2}$	0	$-\frac{1}{2}$	0	$\frac{1}{2}$	0	$\frac{1}{2}$
δ		0		$\frac{1}{2}$		1		δ		0		$\frac{1}{2}$		1	
$\lvert L(g,\delta)\rvert$		24		24		1		$\lvert L(g,\delta)\rvert$		21		28		0	

3.2 Efficient computation of the elements in difference distribution table

The second subsection deals with the relationship between elements of difference distribution tables for the substitutions $g, h \in S(V_n)$ such that $h = (x, y)\, g$.

Algorithm 2.

Input. Substitution $g \in S(V_n)$; the elements $x, y \in V_n$; the DDT $T_2(g)$; the differential spectrum $D(g)$.

Step 1. For each element $\alpha = 1, ..., 2^n - 1$ such that $\alpha \neq x \oplus y$ do the following items:
calculate elements
$\beta_0 = g(x) \oplus g(x \oplus \alpha)$ and $\beta_2 = g(y) \oplus g(y \oplus \alpha)$;
Let us consider 2 cases.
Case 1: assume that $\beta_0 = \beta_2$; then
- calculate element $\beta_1 = g(y) \oplus g(x \oplus \alpha)$;
- for each element $i = 0, 1$ calculate values:

$$\left|D\left(g, p^g_{\alpha,\beta_i}\right)\right| = \left|D\left(g, p^g_{\alpha,\beta_i}\right)\right| - 1,$$

$$\left|D\left(g, p^g_{\alpha,\beta_i} + 4 \cdot (-1)^{i+1}\right)\right| = \left|D\left(g, p^g_{\alpha,\beta_i} + 4 \cdot (-1)^{i+1}\right)\right| + 1.$$

Case 2: suppose that $\beta_0 \neq \beta_2$; then
- calculate elements
$\beta_1 = g(y) \oplus g(x \oplus \alpha)$ è $\beta_3 = g(x) \oplus g(y \oplus \alpha)$;
- for each element $i = 0, ..., 3$ calculate values:

$$\left|D\left(g, p^g_{\alpha,\beta_i}\right)\right| = \left|D\left(g, p^g_{\alpha,\beta_i}\right)\right| - 1,$$

$$\left|D\left(g, p^g_{\alpha,\beta_i} + 2 \cdot (-1)^{i+1}\right)\right| = \left|D\left(g, p^g_{\alpha,\beta_i} + 2 \cdot (-1)^{i+1}\right)\right| + 1.$$

Step 2. The algorithm stops after calculating $h = (x, y)\, g$ and $D(h) = D(g)$.

Output. Substitution $h \in S(V_n)$ such that $h = (x, y)\, g$; the differential spectrum $D(h)$.

Let us denote the indicator function

$$I_\beta(x) = \begin{cases} 1, & \text{if } \beta = x \\ 0, & \text{if } \beta \neq x \end{cases}, \text{where } \beta, x \in V_n.$$

The correctness of the algorithm 2 is shown in the following proposition.

Proposition 3. *For substitutions $g, h \in S(V_n)$ such that $h = (x, y)\, g$ we have*

$$p^h_{\alpha,\beta} - p^g_{\alpha,\beta} = \begin{cases} 0, \textit{ if } \alpha = x \oplus y \\ \left(I_\beta(x_1) + I_\beta(x_2) - I_\beta(x_3) - I_\beta(x_4)\right) \cdot 2^{1-n}, \textit{ otherwise} \end{cases},$$

where $x_1 = g(x \oplus \alpha) \oplus g(y)$, $x_2 = g(y \oplus \alpha) \oplus g(x)$, $x_3 = g(x \oplus \alpha) \oplus g(x)$,

$$x_4 = g(y \oplus \alpha) \oplus g(y).$$

Proof. Let us consider the following transformation sequence

$$p^h_{\alpha,\beta} - p^g_{\alpha,\beta} = P\{h(z \oplus \alpha) \oplus h(z) = \beta\} - P\{g(z \oplus \alpha) \oplus g(z) = \beta\} =$$

$$= \left(\left|\{z \,|h(z \oplus \alpha) \oplus h(z) = \beta\}\right| - \left|\{z \,|g(z \oplus \alpha) \oplus g(z) = \beta\}\right|\right) \cdot 2^{-n} =$$

$$= \left(\sum_{z \in V_n} I_\beta(h(z \oplus \alpha) \oplus h(z)) - \sum_{z \in V_n} I_\beta(g(z \oplus \alpha) \oplus g(z))\right) \cdot 2^{-n} =$$

$$= 2^{1-n} \cdot \sum_{z \in \{x,y\}} \left(I_\beta(h(z \oplus \alpha) \oplus h(z)) - I_\beta(g(z \oplus \alpha) \oplus g(z))\right)$$

It can easily be checked that if $\alpha = x \oplus y$ then $p^h_{\alpha,\beta} - p^g_{\alpha,\beta} = 0$.
Assume that $\alpha \neq x \oplus y$, then we get

$$\Big(I_\beta(g(x \oplus \alpha) \oplus g(y)) + I_\beta(g(y \oplus \alpha) \oplus g(x)) -$$

$$- I_\beta(g(x \oplus \alpha) \oplus g(x)) - I_\beta(g(y \oplus \alpha) \oplus g(y))\Big) \cdot 2^{1-n} =$$

$$= \big(I_\beta(x_1) + I_\beta(x_2) - I_\beta(x_3) - I_\beta(x_4)\big) \cdot 2^{1-n}.$$

This concludes the proof. □

Let us denote by t_2 the complexity of an algorithm 2.

Proposition 4. *As $n \to \infty$ we obviously have*

$$t_2 = O(2^n).$$

Proof. The complexity of algorithm is the dot product of the following values:

1. the number of the iterations of step 1 is $2^n - 1$;
2. constant number of operations on the steps of the algorithm.

□

Remark 2. *The algorithm 2 is nearly 2^n times faster than the classical algorithm of computing the differential uniformity.*

Example 2. *Consider the following substitutions $g, h \in S(V_3)$ such that $h = (0,1)\,g$*

$$g = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 5 & 6 & 1 & 7 & 2 & 4 & 3 & 0 \end{pmatrix}, h = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 6 & 5 & 1 & 7 & 2 & 4 & 3 & 0 \end{pmatrix}.$$

The results obtained by algorithm 2 are presented in Table 2 (all changing elements of the matrix are in bold).

In particular taking into account preposition 3 we can conclude

1. *Since $\alpha = (0 \oplus 1) = 1$ we see that for any $\beta \in V_3$ we obtain*

$$p^h_{1,\beta} - p^g_{1,\beta} = 0;$$

2. $p^h_{4,4} - p^g_{4,4} = (I_4(4) + I_4(1) - I_4(7) - I_4(2)) \cdot 2^{-2} = 1/4;$

3. $p^h_{7,5} - p^g_{7,5} = (I_5(6) + I_5(6) - I_5(5) - I_5(5)) \cdot 2^{-2} = -2/4.$

Table 2.

Substitution g								*Substitution h*							
5 6 1 7 2 4 3 0								6 5 1 7 2 4 3 0							
DDT of g								*DDT of h*							
1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	$\frac{2}{4}$	0	0	$\frac{2}{4}$	0	0	0	0	$\frac{2}{4}$	0	0	$\frac{2}{4}$	0
0	$\mathbf{\frac{2}{4}}$	**0**	0	$\mathbf{\frac{2}{4}}$	0	0	**0**	0	$\mathbf{\frac{1}{4}}$	$\mathbf{\frac{1}{4}}$	0	$\mathbf{\frac{1}{4}}$	0	0	$\mathbf{\frac{1}{4}}$
0	**0**	$\mathbf{\frac{2}{4}}$	0	**0**	0	0	$\mathbf{\frac{2}{4}}$	0	$\mathbf{\frac{1}{4}}$	$\mathbf{\frac{1}{4}}$	0	$\mathbf{\frac{1}{4}}$	0	0	$\mathbf{\frac{1}{4}}$
0	**0**	$\mathbf{\frac{2}{4}}$	0	**0**	0	0	$\mathbf{\frac{2}{4}}$	0	$\mathbf{\frac{1}{4}}$	$\mathbf{\frac{1}{4}}$	0	$\mathbf{\frac{1}{4}}$	0	0	$\mathbf{\frac{1}{4}}$
0	$\mathbf{\frac{2}{4}}$	**0**	0	$\mathbf{\frac{2}{4}}$	0	0	**0**	0	$\mathbf{\frac{1}{4}}$	$\mathbf{\frac{1}{4}}$	0	$\mathbf{\frac{1}{4}}$	0	0	$\mathbf{\frac{1}{4}}$
0	0	0	$\frac{2}{4}$	0	**0**	$\mathbf{\frac{2}{4}}$	0	0	0	0	$\frac{2}{4}$	0	$\mathbf{\frac{2}{4}}$	**0**	0
0	0	0	0	0	**1**	**0**	0	0	0	0	0	0	$\mathbf{\frac{2}{4}}$	$\mathbf{\frac{2}{4}}$	0
p	0	$\frac{1}{4}$	$\frac{2}{4}$	$\frac{3}{4}$	1			p	0	$\frac{1}{4}$	$\frac{2}{4}$	$\frac{3}{4}$	1		
$\lvert D(g,p)\rvert$	36	0	12	0	1			$\lvert D(g,p)\rvert$	25	16	6	0	0		

4 Cryptographic applications

The results of this paper can be applied to optimize some heuristic methods of constructing s-boxes. It is well-known that the most heuristic methods are based on swap operations (for example, see [1], [2], [3], [4], [5], [6], [9]). We can optimize some of this methods using algorithms of this paper. Let's show this for the spectral-linear and spectral-differential methods of generating s-boxes (see [4], [5]).

Let t_{sl} be the computational complexity of algorithm 2 described in [4].

Proposition 5. *As $n \to \infty$ we have the following*

$$t_{sl} = O\left(2^{7n}\right).$$

Proof. In the paper [4], it is proved that

$$t_{sl} = O\left(n \cdot 2^{7n}\right).$$

The complexity of algorithm is the product of the following values:

1. the complexity of step 2 is $c_1 \cdot 2^{4n} \cdot n$, $c_1 = \text{const}$;
2. the maximum number of iterations of the step 2 is $c_2 \cdot 2^{3n}$, $c_2 = \text{const}$.

Using algorithm 1 of this paper we obtain the following complexity bounds of step 2 $c_1 \cdot 2^{4n}$. This completes the proof of proposition. □

The reader will easily prove the following proposition.

Let t_{sd} be the computational complexity of algorithm 1 described in [4].

Proposition 6. *As $n \to \infty$ we obviously have*

$$t_{sd} = O\left(n \cdot 2^{6n}\right).$$

Suppose t_{new} is the average execution time of the modified algorithm, t_{old} is the average execution time of its original version. For n ($n = 5, ..., 8$) table 3 includes the value $\frac{t_{old}}{t_{new}}$ of spectral-linear and spectral-differential methods. In particular from table 1 we obtain the following:

1. if $n = 7$ then modified algorithm is nearly 4 times faster than its original version (see Algorithm 2 in[4]);

2. if $n = 8$ then modified algorithm is nearly 28 times faster than the old one (see Algorithm 1 in [4]).

Table 3.

n	5	6	7	8
Spectral-linear method (see Algorithm 2 in[4])	2	3	4	5
Spectral-differential method (see Algorithm 1 in [4])	5	8	16	28

5 Conclusions

The results of our paper can be summarized as follows.

The δ_h-parameter (the linear spectrum $L(h)$, the LAT $T_1(h)$) of an s-box $h \in S(V_n)$ such that $h = (x, y)\, g$ can be computed with time complexity $O\left(2^{2n}\right)$. This is effected by using the algorithm 1 which described in this paper.

The p_h-parameter (the differential spectrum $D(h)$, the DDT $T_2(h)$) of an s-box $h \in S(V_n)$ such that $h = (x, y)\, g$ can be calculated with time complexity $O(2^n)$. This is effected by using the algorithm 2 which described in this paper.

We optimized some heuristic methods of generating s-boxes. The optimized methods can be applied for generating of big-size s-boxes.

References

[1] Isa H., Jamil N., and Z'aba M., "Hybrid Heuristic Methods in Constructing Cryptographically Strong S-boxes", *International Journal of Cryptology Research*, **6**:1 (2016).

[2] Ivanov G., Nikolov N., and Nikova S., "Cryptographically Strong S-Boxes generated by modified immune Algorithm", *Cryptography and Information Security in the Balkans, Lecture Notes in Computer Science*, **9540** (2015), 31–42.

[3] Ivanov G., Nikolov N., and Nikova S., "Reversed genetic algorithms for generation of bijective S-boxes with good cryptographic properties cryptographic parameters", *Cryptography and Communications*, **8**:2 (2016), 247–276.

[4] Menyachihin A.V., "Spectral-linear and spectral-differential methods for generating S-boxes having almost optimal cryptographic parameters", *Mathematical aspects of cryptography*, **8**:2 (2017), 97–116.

[5] Menyachihin A.V., "The method for generating s-boxes by using elements of linear and differential spectra and device implementing it", *RU 2633132,*, **8** (2016).

[6] Kazymyrov O.V., Kazymyrova V.N., Oliynykov R.V., "A method for generation of high-nonlinear S-boxes based on gradient descent", *Mathematical aspects of cryptography*, **5**:2 (2014), 71–78.

[7] Millan W., Clark A. and Dawson E., "Boolean Functions Design Using Hill Climbing Methods", *Lecture Notes in Computer Science*, **1587** (1999), 1–11.

[8] Millan W., "How to improve the nonlinearity of bijective s-boxes", *In Australian Conference on Information Security and Privacy*, **1438** (1998), 181–192.

[9] Tesar P., "A New Method for Generating High Non-linearity S-Boxes", *Radioengineering*, **19**:1 (2010), 23–26.

[10] Yu Y., Wang M., Li Y., "Constructing differentially 4 uniform permutation from know ones", *Chinese Journal of Electronics*, **22**:2 (2013), 495–499.

Linear and Differential Cryptanalysis: Another Viewpoint

Fedor Malyshev[1] and Andrey Trishin[2]

[1] Steklov Mathematical Institute of RAS, Russia
[2] Certification Research Center LLC, Russia
malyshevfm@mi.ras.ru, trishin17@yandex.ru

Abstract

Theorems on the exact values of linear and differential characteristics are proved based on the separation of the cipher functional scheme into nonlinear part and linear medium. The example of universal functional scheme demonstrates a significant range of possible errors using the current way to estimate the characteristics of probabilistic relations. It is stressed the difference in obtaining complexity estimates in linear and differential cryptanalysis in comparison with some other types of cryptanalytic techniques and the importance of proper way to implement experiments in order to verify the estimation values of the relations characteristics to their true values. The point of view of finding relations under the condition of a fixed cipher key is insisted. The duality of the linear and differential cryptanalysis based on the concept of the linear medium is exposed and formulated mathematically strictly. The degrees of diffusion in linear medium are defined which maximization is one of the basic principles of ciphers' design. By that, the qualitative property of high diffusion of the cipher formulated by K. Shannon is formalized.

Keywords: linear cryptanalysis, differential cryptanalysis, linear medium, block ciphers.

1 Introduction

The point of view on linear and differential cryptanalysis presented here was formed independently of their numerous representations in cryptographic literature. These methods should be called local linear and local differential cryptanalysis more correctly.

This paper is devoted to a technique for constructing a probabilistic linear and differential relations of functions $F : V_N \to V_M$ defined by a functional schemes. Here $V_N = GF(2)^N$, $V_M = GF(2)^M$ are arithmetic vector spaces over the field $GF(2)$.

Let V_N^* denotes the set of all Boolean column vectors of length N. Suppose a is a random vector with the uniform probabilistic distribution on V_N. The non-strict equation $aL' \simeq bL''$, $a \in V_N$, $b = F(a) \in V_M$, is called a probabilistic linear relation defined by column vectors $L' \in V_N^*$, $L'' \in V_M^*$ if the measure of its strictness $\delta_{L',L''} = \delta^F_{L',L''} = 2\mathbf{P}\{aL' = bL''\} - 1$ is defined. The value $\delta_{L',L''}$ is called a linear characteristic of a probabilistic linear relation.

Let $D' \in V_N$, $D'' \in V_M$ are any fixed vectors, $a^{(1)}, a^{(2)} \in V_N$, $b^{(1)}, b^{(2)} \in V_M$ are any vectors such that $b^{(1)} = F(a^{(1)})$, $b^{(2)} = F(a^{(2)})$. A non-strict implication "if $a^{(1)}+a^{(2)} = D'$, then $b^{(1)}+b^{(2)} = D''$" is called a probabilistic differential relation and denoted by (D', D''). A measure of strictness of this implication $p_{D',D''} = p^F_{D',D''} = \mathbf{P}\{F(a + D') + F(a) = D''\}$ is called a differential characteristic of a differential relation.

Probabilistic linear and differential relations are used in attacks on cryptographic keys of the ciphers usually defined by functional schemes. In the next section we'll give some definitions related to functional schemes.

2 The linear medium of a functional scheme

The function $F : V_N \to V_M$, $V_N \ni a \mapsto b = F(a) \in V_M$, defined by any cipher with any fixed key is called a cipher transformation. It is specified using the functional scheme $\mathcal{F}$, i.e. the sequence of linear and nonlinear mappings. This sequence may be considered as the computer program without cycles.

Let the nonlinear mappings $f_i : V_{n_i} \to V_{m_i}$, $x_i \mapsto y_i = f_i(x_i)$, $i = 1, \dots, k$, be performed in this program by definite order. By $x_i \in V_{n_i}$ we denote the argument of the function f_i. This argument is expressed using $a \in V_N$ as the result of applying some previous operations. The linear operations are used between nonlinear ones. Any linear operation may be expressed using additions modulo 2 and reproduction nodes with several outputs $x \mapsto (x, \dots, x)$.

We won't consider linear operations of the cipher separately. It suffices to say that for any Boolean $m_i \times n_j$ matrices c_{ij}, $i = 0, 1, \dots, k$, $j = 1, \dots, k, k+1$, $m_0 = N$, $n_{k+1} = M$, we can write $x_j = ac_{0j} + \sum_{i=1}^{j-1} y_i c_{ij}$, $j = 1, \dots, k$, $b = ac_{0,k+1} + \sum_{i=1}^{k} y_i c_{i,k+1}$. Let $c_{ij} = 0$ if $i \geqslant j$. Then it's possible to construct a matrix C of size $\left(N + \sum_{i=1}^{k} m_i\right) \times \left(\sum_{i=1}^{k} n_i + M\right)$ with blocks c_{ij} such that

$$(a, y_1, \dots, y_k)C = (x_1, \dots, x_k, b). \tag{1}$$

The matrix C is said to be a matrix of linear medium of the functional scheme $\mathcal{F}$. It integrates all linear operations of the cipher transformation. The linear medium is denoted by the same letter C and is also a functional scheme. It is obtained from the functional scheme $\mathcal{F}$ by deletion of all functional elements f_i, $i = 1, \dots, k$. As a result all n_i Boolean inputs of any functional element f_i becomes the outputs of the linear medium C and all m_i Boolean outputs of this element becomes the inputs of the linear medium C. The linear medium C determines a linear mapping $C : V_{N+\sum_{i=1}^{k} m_i} \to V_{\sum_{i=1}^{k} n_i+M}$ in accordance with (1).

Thus the functional scheme $\mathcal{F}$ is represented by the linear medium C in which the nonlinear elements $f_i : V_{n_i} \to V_{m_i}$, $i = 1, \dots, k$, are embedded. As a result we are able to study separately properties of the linear medium C of the cipher transformation F and properties of its nonlinear part consisting of separate functions f_i, $i = 1, \dots, k$.

The representation of the cipher transformation by a functional scheme is ambiguous. The cryptanalyst himself chooses a certain functional scheme determining the cipher transformation F. If he chooses local elements f_i, $i = 1, \dots, k$, too large, then the analysis of each element will be difficult, but this will reduce their number and simplify the linear medium. On the other hand, if he chooses elements f_i too small up to Sheffer functions, it will increase their number and will complicate the linear medium.

3 Chains of conformal local linear relations

The main aim of linear cryptanalysis is to construct a pair of column vectors $L' \in V_N^*$, $L'' \in V_M^* \backslash \{0\}$ such that absolute value of linear characteristic $\delta_{L',L''} = \delta_{L',L''}^F = 2\mathbf{P}\{aL' = bL''\} - 1$ of linear relation $aL' \simeq bL''$ is as high as possible. Here $b = F(a)$ and the vector $a \in V_N$ has the uniform probabilistic distribution on V_N.

The probabilistic linear relation $aL' \simeq bL''$ or, what is the same thing, $aL' + bL'' \simeq 0$ is obtained by formal addition modulo 2 of local probabilistic linear relations $x_i l_i' + y_i l_i'' \simeq 0$, $l_i' \in V_{n_i}$, $l_i'' \in V_{m_i}$, $y_i = f_i(x_i)$, $i = 1, \dots, k$, characterized by linear characteristics $\delta_i = \delta_{l_i',l_i''}^{f_i} = 2\mathbf{P}\{x_i l_i' = y_i l_i''\} - 1$, where each vector x_i has the uniform probabilistic distribution on V_{n_i}.

The set $\mathfrak{L} = ((l_i', l_i''), i = 1, \dots, k)$ should be conformal in the sense that for some $L' \in V_N^*$, $L'' \in V_M^*$ we have the equality $\sum_{i=1}^{k}(x_i l_i' + y_i l_i'') = aL' + bL''$, or

the following chain of equalities

$$0=\sum_{i=1}^{k}(x_il'_i+y_il''_i)+aL'+bL''=xl'+yl''+aL'+bL''=(a,y)\begin{pmatrix}L'\\l''\end{pmatrix}+$$
$$+(x,b)\begin{pmatrix}l'\\L''\end{pmatrix}=(a,y)\begin{pmatrix}L'\\l''\end{pmatrix}+(a,y)C\begin{pmatrix}l'\\L''\end{pmatrix}=(a,y)\left[\begin{pmatrix}L'\\l''\end{pmatrix}+C\begin{pmatrix}l'\\L''\end{pmatrix}\right],$$

with respect to variables x_i, y_i, $i=1,\dots,k$ (as if under independent a,y). This variables are related with vectors a and b only by linear relation (1) excluding equations $y_i=f_i(x_i)$. In the equalities above $x=(x_1,\dots,x_k)$, $y=(y_1,\dots,y_k)$ and $l'\in V^*_{\sum_{i=1}^k n_i}$ is a concatenation of vectors l'_i, similarly $l''\in V^*_{\sum_{i=1}^k m_i}$ is a concatenation of vectors l''_i, $i=1,\dots,k$. Thus the set $\mathfrak{L}=((l'_i,l''_i),\ i=1,\dots,k)$ is conformal iff

$$C\begin{pmatrix}l'\\L''\end{pmatrix}=\begin{pmatrix}L'\\l''\end{pmatrix}, \tag{2}$$

for some $L'=L'_{\mathfrak{L}}\in V^*_N$ and $L''=L''_{\mathfrak{L}}\in V^*_M$.

Let us consider a product $\tilde{\delta}=\delta_1\cdot\ldots\cdot\delta_k$ as a rough approximation (approximate value) for the linear characteristic $\delta_{L',L''}$. This **is motivated** by the next reasoning.

Assume that if the random input $a\in V_N$ is uniformly distributed then random variables $x_i\in V_{n_i}$ for all $i=1,\dots,k$, are uniformly distributed. Suppose that the Boolean random variables $\eta_i=x_il'_i+y_il''_i$, $i=1,\dots,k$ are statistically independent. Then $\delta_{L',L''}=\tilde{\delta}$.

Let $\mathfrak{W}(L',L'')$ be the set of all solutions of the system (2) with respect to $\mathfrak{L}=((l'_i,l''_i),\ i=1,\dots,k)$ given $L'\in V^*_N$, $L''\in V^*_M$ (the vectors L',L'' are called boundary conditions). Let $\mathfrak{W}=\bigcup_{L'\in V^*_N,L''\in V^*_M}\mathfrak{W}(L',L'')$ and $\mathfrak{W}^{(0)}=\{\mathfrak{L}\in\mathfrak{W}\,|\,l''_i=0\Rightarrow l'_i=0,\ i=1,\dots,k\}$. If $\mathfrak{L}\in\mathfrak{W}\backslash\mathfrak{W}^{(0)}$, then $\tilde{\delta}_{\mathfrak{L}}=0$, where $\tilde{\delta}_{\mathfrak{L}}=\prod_{i=1}^k\delta^{f_i}_{l'_i,l''_i}$. To construct the required linear relation we should find a set $\mathfrak{L}\in\mathfrak{W}^{(0)}$ for which the product $|\tilde{\delta}_{\mathfrak{L}}|$ is as high as possible. The factors equal to one are preferable in the product $\tilde{\delta}_{\mathfrak{L}}$. This factors can appear if $l''_i=0$. Suppose $\mathfrak{L}\in\mathfrak{W}^{(0)}$. Let $\theta_{\mathfrak{L}}=|\{i\in\{1,\dots,k\}\,|\,l''_i\neq 0\}|$ be the number of factors in product $\tilde{\delta}_{\mathfrak{L}}$ possibly not equal to one. To maximize $|\tilde{\delta}_{\mathfrak{L}}|$ sometimes it is necessary to find a number $\theta_C=\min_{\mathfrak{L}\in\mathfrak{W}^{(0)}\backslash\{0\}}\theta_{\mathfrak{L}}$ called a degree of diffusion of the linear medium C with respect to linear cryptanalysis. The sets $\mathfrak{L}\in\mathfrak{W}^{(0)}$ are called a chains of conformal local linear relations if only nontrivial (when $l''_i\neq 0$) local linear relations $x_il'_i\simeq y_il''_i$, $i=1,\dots,k$, are considered.

The technique described in this section can be applied for searching multidimensional (s-dimentional) linear relations [1]. In this case it is necessary to consider matrices with s columns instead of column vectors L', L'', l'_i, l''_i, $i = 1, \dots, k$.

4 Chains of conformal local differential relations

If we consider equations (1) for two inputs $a^{(1)}$, $a^{(2)} \in V_N$ of the functional scheme and subtract one from the other, we will get

$$(D', d''_1, \dots, d''_k)C = (d'_1, \dots, d'_k, D''). \tag{3}$$

where $D' = a^{(1)} + a^{(2)}$, $D'' = b^{(1)} + b^{(2)}$, $d''_i = y_i^{(1)} + y_i^{(2)}$, $d'_i = x_i^{(1)} + x_i^{(2)}$, $i = 1, \dots, k$. Here $b^{(1)} = F(a^{(1)})$, $b^{(2)} = F(a^{(2)})$ and $x_i^{(1)}$, $x_i^{(2)}$ are the function f_i argument values, if $a^{(1)}$ and $a^{(2)}$ are the inputs of the functional scheme, $y_i^{(1)} = f_i(x_i^{(1)})$, $y_i^{(2)} = f_i(x_i^{(2)})$, $i = 1, \dots, k$.

The main aim of differential cryptanalysis is to construct a pair of vectors $D' \in V_N \backslash \{0\}$, $D'' \in V_M$ such that the differential characteristic $p_{D',D''} = p^F_{D',D''} = \mathbf{P}\{F(a + D') + F(a) = D''\}$ of non-strict implication $a^{(1)} + a^{(2)} = D' \Rightarrow b^{(1)} + b^{(2)} = D''$ is as high as possible.

Let $D' \in V_N$, $D'' \in V_M$, $d'_i \in V_{n_i}$, $d''_i \in V_{m_i}$, $i = 1, \dots, k$, is an arbitrary set of vectors, satisfying the conformity condition (3). Since C is a block upper triangular matrix, then from implications $x_i^{(1)} + x_i^{(2)} = d'_i \Rightarrow y_i^{(1)} + y_i^{(2)} = d''_i$ for all $i = 1, \dots, k$, it follows that $a^{(1)} + a^{(2)} = D' \Rightarrow b^{(1)} + b^{(2)} = D''$.

Let us have the set of local differences $\mathfrak{D} = ((d'_i, d''_i),\ i = 1, \dots, k)$, $d'_i \in V_{n_i}$, $d''_i \in V_{m_i}$ satisfying the condition (3) for some boundary vectors $D' = D'_{\mathfrak{D}} \in V_N$, $D'' = D''_{\mathfrak{D}} \in V_M$. Consider a product $\tilde{p}_{\mathfrak{D}} = \prod_{i=1}^{k} p^{f_i}_{d'_i,d''_i}$ of differential characteristics $p^{f_i}_{d'_i,d''_i} = \mathbf{P}\left\{f_i(x_i^{(1)} + d'_i) + f_i(x_i^{(1)}) = d''_i\right\}$ of local differential relations (d'_i, d''_i), where each vector $x_i^{(1)}$ has the uniform probabilistic distribution on V_{n_i}.

The product $\tilde{p}_{\mathfrak{D}}$ is a rough approximation for the differential characteristic $p^F_{D',D''}$ of the differential relation $(D', D'') \in V_N \times V_M$. This **is motivated** by the next reasoning. Let $a^{(2)} = a^{(1)} + D'$. Suppose the uniformity of distributions of random variables $x_i^{(1)} \in V_{n_i}$, $i = 1, \dots, k$, follows from the uniformity of distribution of the random input $a^{(1)} \in V_N$. Also suppose that the events $y_i^{(1)} +$

$y_i^{(2)} = d''_i$, $i = 1, \dots, k$, are independent. Then, using (3), we have

$$p_{D',D''} \geqslant \mathbf{P}\{y_i^{(1)} + y_i^{(2)} = d''_i, i = 1, \dots, k\} =$$
$$= \prod_{i=1}^{k} \mathbf{P}\{f_i(x_i^{(1)}) + f_i(x_i^{(2)}) = d''_i\} =$$
$$= \prod_{i=1}^{k} \mathbf{P}\left\{f_i\left(x_i^{(1)}\right) + f_i\left(x_i^{(1)} + d'_i\right) = d''_i\right\} = \tilde{p}_{\mathfrak{D}}.$$

Let $W(D', D'')$ be the set of all solutions of the system (3) with respect to $\mathfrak{D} = ((d'_i, d''_i), i = 1, \dots, k)$ given boundary differences $D' \in V_N$, $D'' \in V_M$. Let $W = \bigcup_{D' \in V_N, D'' \in V_M} W(D', D'')$ and $W^{(0)} = \{\mathfrak{D} \in W | \, d'_i = 0 \Rightarrow d''_i = 0, i = 1, \dots, k\}$. If $\mathfrak{D} \in W \backslash W^{(0)}$, then $\tilde{p}_{\mathfrak{D}} = 0$. To construct the required differential relation we should find a set $\mathfrak{D} \in W^{(0)}$ for which the product $\tilde{p}_{\mathfrak{D}} = \prod_{i=1}^{k} p_{d'_i, d''_i}^{f_i}$ is as high as possible. The factors equal to one are preferable in this product. This factors can appear if $d'_i = 0$. For any $\mathfrak{D} \in W^{(0)}$ let $\theta'_{\mathfrak{D}} = |\{i \in \{1, \dots, k\} | \, d'_i \neq 0\}|$ be the number of factors in product $\tilde{p}_{\mathfrak{D}}$ possibly not equal to one. To maximize $\tilde{p}_{\mathfrak{D}}$ sometimes it is necessary to find a number $\theta'_C = \min_{\mathfrak{D} \in \mathrm{W}^{(0)} \backslash \{0\}} \theta'_{\mathfrak{D}}$ called a degree of diffusion of the linear medium C with respect to differential cryptanalysis. The sets $\mathfrak{D} \in W^{(0)}$ are called a chains of conformal local differential relations if only nontrivial ($d'_i \neq 0$) local differential relations among (d'_i, d''_i), $i = 1, \dots, k$, are considered.

5 Theorems about exact values of linear and differential characteristics

Theorem 1. *If $L' \in V_N^*$, $L'' \in V_M^*$, then $\delta_{L',L''} = \sum_{\mathfrak{L} \in \mathfrak{W}(L',L'')} \tilde{\delta}_{\mathfrak{L}}$.*

Theorem 2. *If $D' \in V_N$, $D'' \in V_M$, then*

$$p_{D',D''} =$$
$$= \sum_{\mathfrak{D} \in W(D',D'')} \tilde{p}_{\mathfrak{D}} + \frac{1}{2^M} \sum_{(L',L'') \in V_N^* \times V_M^*} (-1)^{D'L' + D''L''} \sum_{\substack{\mathfrak{L}_1, \mathfrak{L}_2 \in \mathfrak{W}(L',L''): \\ \mathfrak{L}_1 \neq \mathfrak{L}_2}} \tilde{\delta}_{\mathfrak{L}_1} \tilde{\delta}_{\mathfrak{L}_2}.$$

The theorem 1 have been proved in special cases (see [12],[14]). The theorem 2 follows from the theorem 1 and the fact from [12] about the links between

linear and differential characteristics of Boolean mappings. In the light of this theorems, the search of linear and differential relations (see sections 3 and 4) is performed by maximizing the absolute values of individual summands $|\tilde{\delta}_{\mathfrak{L}}|$ and $\tilde{p}_{\mathfrak{D}}$ in formulas for exact values $\delta_{L',L''}$ and $p_{D',D''}$. But this maximizations are carried out for all $\mathfrak{L} \in \mathfrak{W}^{(0)} \backslash \{0\}$ and for all $\mathfrak{D} \in W^{(0)} \backslash \{0\}$. This fact is an additional motivation for choice $\tilde{\delta}_{\mathfrak{L}}$ and $\tilde{p}_{\mathfrak{D}}$ as rough approximations for $\delta_{L',L''}$ and $p_{D',D''}$.

Also due to theorems 1 and 2 we can see the disadvantages of the technique of linear and differential cryptanalysis described above. Here are the main disadvantages:

1. We are searching for relations that we can find, not the best ones.

2. General results about the exactness of the approximations for linear and differential characteristics are not available.

3. If we focus on problems of finding $\tilde{\delta} = \max_{\mathfrak{L} \in \mathfrak{W}^{(0)} \backslash \{0\}} |\tilde{\delta}_{\mathfrak{L}}|$ and $\tilde{p} = \max_{\mathfrak{D} \in W^{(0)} \backslash \{0\}} \tilde{p}_{\mathfrak{D}}$, then we have to leave the domains containing the values $\delta = \max_{L' \in V_N^*, L'' \in V_M^* \backslash \{0\}} |\delta_{L',L''}|$ and $p = \max_{D' \in V_N \backslash \{0\}, D'' \in V_M} p_{D',D''}$.

Let's explain the last. The problems of finding $\tilde{\delta}$ and $\tilde{p}$ are related to minimization of characteristics $\theta_{\mathfrak{L}}$ and $\theta'_{\mathfrak{D}}$. If we decrease $\theta_{\mathfrak{L}}$ and $\theta'_{\mathfrak{D}}$, then the cardinalities $|\mathfrak{W}(L', L'')|$, $|W(D', D'')|$ (see theorems 1 and 2) will reduce and the values $|\delta_{L'_{\mathfrak{L}}, L''_{\mathfrak{L}}}|$ and $p_{D'_{\mathfrak{D}}, D''_{\mathfrak{D}}}$ will probably decrease.

We have to put up with these disadvantages because the problems of maximazing the values $|\delta_{L',L''}|$ and $p_{D',D''}$ is much more difficult then the problems of maximazing the values $|\tilde{\delta}_{\mathfrak{L}}|$ and $\tilde{p}_{\mathfrak{D}}$ as it can be seen from theorems 1 and 2.

Of course, the assumptions about the uniformity of random variables $x_i \in V_{n_i}$, $i = 1, \ldots, k$, and on the independence of events in motivations formulated above are not satisfied as a rule. Nevertheless, **SOMETIMES** we have $\tilde{\delta} \approx \delta$, $\tilde{p} \approx p$, for example, $3^{-1} \leqslant \tilde{\delta}/|\delta| \leqslant 3$, $10^{-1} \leqslant \tilde{p}/p \leqslant 10$. We can verify this with the help of empirical estimations δ^*, p^* and appropriate confidence intervals. If we exceed characteristic threshold of measurement accuracy, then this can lead to the study of phenomena not related to the essence of the measured value. The characteristic **SOMETIMES** refers to a sets of all possible cipher transformations and chains of conformal local probabilistic relations obtained by the cryptanalyst.

If the values $|\tilde{\delta}_{\mathfrak{L}}|$, $\tilde{p}_{\mathfrak{D}}$ are small, then the verification of the approximate equalities $\tilde{\delta}_{\mathfrak{L}} \approx \delta_{L'_{\mathfrak{L}}, L''_{\mathfrak{L}}}$, $\tilde{p}_{\mathfrak{D}} \approx p_{D'_{\mathfrak{D}}, D''_{\mathfrak{D}}}$ is complicated due to limited computing resources. In this case we have to consider a close analogues of the investigated

cipher. According to the authors, the best analogues are those ciphers that have the same linear medium, even with the same number of iterations. In this case the equalities in theorems 1 and 2 have the same form. A verification must be carried out as many situations as possible. For this purpose for the fixed n_i and m_i the functional elements $\tilde{f}_i : V_{n_i} \to V_{m_i}$, $i = 1, \dots, k$, should be varied so as to increase $|\tilde{\delta}_{\mathfrak{L}}|$, $\tilde{p}_{\mathfrak{D}}$. And for any $i \in \{1, \dots, k\}$ even the inequalities $\left|\delta^{\tilde{f}_i}_{l'_i, l''_i}\right| < \left|\delta^{f_i}_{l'_i, l''_i}\right|$, $p^{\tilde{f}_i}_{d'_i, d''_i} < p^{f_i}_{d'_i, d''_i}$ can be achieved. The functional elements f_i, $i \in \{1, \dots, k\}$, for which $l'_i = 0$, $l''_i = 0$ or $d'_i = 0$, $d''_i = 0$ may also vary.

The characteristics $\tilde{\delta}$, $\tilde{p}$ have advantages and disadvantages, but they are independent characteristics of the cipher more important then δ, p. And it doesn't even matter that they are close to or far from δ, p. Usually, the characteristics δ, p are in the imagination of cryptanalyst. They cannot be evaluated for the real ciphers in contrast to $\tilde{\delta}$, $\tilde{p}$.

A large number of indirect data points to the proximity of δ^2 and p. For this reason the linear cryptanalysis is preferable to the differential cryptanalysis (in some cases). For example, we need about $1/\delta^2$ known plaintexts to attack the cipher with linear cryptanalysis and about $1/p$ chosen plaintexts or chosen ciphertexts to attack it with differential cryptanalysis, more precisely, about $1/p$ pairs $(a^{(1)}, b^{(1)})$, $(a^{(2)}, b^{(2)})$ such that $a^{(1)} + a^{(2)} = D'$ or $b^{(1)} + b^{(2)} = D''$.

6 The duality of differential and linear cryptanalysis

From the comparison of sections 3 and 4 it follows that the problems of finding linear and differential relations are identical. It is not accidental [2]. The equality (2) follows from $\left(l_1'^T, \dots, l_k'^T, L''^T\right) C^T = \left(L'^T, l_1''^T, \dots, l_k''^T\right)$ by transposition or from equality $\left(L''^T, l_k'^T, \dots, l_1'^T\right) C^* = \left(l_k''^T, \dots, l_1''^T, L'^T\right)$, where the matrix $C^* = \|c^*_{ij}\|$, $i = 0, 1, \dots, k$, $j = 1, \dots, k, k+1$, is obtained from the matrix C^T by centrally symmetric permutation of blocks: $c^*_{ij} = c^T_{k+1-j, k+1-i}$.

Thus, the problem of finding probabilistic differential relations of the cipher F defined by the functional scheme $\mathcal{F}$ with the linear medium C and local nonlinear functions $f_i : V_{n_i} \to V_{m_i}$, $i = 1, \dots, k$, is equivalent to the problem of finding probabilistic linear relations for a different (for a dual) functional scheme $\mathcal{F}^*$ with the linear medium C^* and local nonlinear functions such that $\delta^{f^*_i}_{l', l''} = p^{f_{k+1-i}}_{l''^T, l'^T}$, $l' \in V^*_{m_{k+1-i}}$, $l'' \in V^*_{n_{k+1-i}}$, $i = 1, \dots, k$.

Similarly, the problem of finding probabilistic linear relations of the cipher F is equivalent to the problem of finding probabilistic differential relations for a functional scheme $\mathcal{F}^*$, for which $p^{f_i^*}_{d',d''} = \delta^{f_{k+1-i}}_{d''^T,d'^T}$, $d' \in V_{m_{k+1-i}}$, $d'' \in V_{n_{k+1-i}}$, $i = 1, \dots, k$.

The functions $f_i^* : V_{m_i} \to V_{n_i}$, $i = 1, \dots, k$, with such characteristics may not exist, but it does not matter. The mappings f_i are not used to find probabilistic linear and differential relations. Only the matrices $\left\| |\delta^{f_i}_{l',l''}| \right\|_{l' \in V^*_{n_i}, l'' \in V^*_{m_i}}$ and $\left\| p^{f_i}_{d',d''} \right\|_{d' \in V_{n_i}, d'' \in V_{m_i}}$ are needed for this.

7 Universal functional scheme

Consider two functional schemes $\mathfrak{f}_\varepsilon$, $\varepsilon \in GF(2)$, with parameters $N = 2$, $M = 1$, $k = 2$, defined by equations $y_1 = x_0x_1 + (x_0 + \varepsilon)(x_1 + 1 + \varepsilon) = (1 + \varepsilon)x_0 + \varepsilon x_1$. For functional elements $(x_0 + \varepsilon)(x_1 + 1 + \varepsilon)$ the absolute values of linear and differential characteristics are equal for all relations and are not depend on $\varepsilon \in GF(2)$.

A universal functional scheme $\mathfrak{F}$ for functions $V_N \to V_M$, $(a_1, \dots, a_N) \mapsto (b_1, \dots, b_M)$, includes $M2^N$ schemes $\mathfrak{f}_\varepsilon$, 2^N schemes for each b_i, $i = 1, \dots, M$. For any scheme $\mathfrak{f}_\varepsilon$ we put $x_0 = a_1 + a_1 = 0$, and denote by x_1 any one of 2^N conjunction of Boolean variables $a_1, \dots, a_N$. Any b_i is a sum modulo 2 of outputs of 2^N schemes $\mathfrak{f}_\varepsilon$. We may obtain any function $V_N \to V_M$ by selecting the value ε for each of $M2^N$ schemes. Each function corresponds to its own scheme. Each of the 2^{M2^N} functional schemes has the same linear medium. The absolute values of linear and differential characteristics are equal for the corresponing functional elements. If we use this functional schemes and follow the recommendations from sections 3 and 4, then for all functions $V_N \to V_M$ we obtain the same "best" probabilistic linear relation and the same "best" differential relation. This is an extreme example of the first disadvantage from section 5. The universal functional scheme allows us to obtain exotic examples of relations both between $\tilde{\delta}$ and δ and between $\tilde{p}$ and p.

8 Mission of cryptographic keys in linear and differential cryptanalysis

Most authors obtain probabilistic linear and differential relations and estimate their characteristics directly for functions $\Phi : V_N \times V_K \to V_M$, $(a, X) \mapsto b = \Phi(a, X)$, where $X \in V_K$ is a cryptographic key. They often use various kinds of probability-theoretic models depending on X (besides the probability distributions on $V_N \times V_K$).

The authors of this work are deeply convinced of the following. The probabilistic relations used to determine subkey Z that apply before (or after) the mapping Φ, should to be constructed for the cipher transformation $F : a \mapsto b = \Phi(a, X)$ in accordance with sections 3 and 4 for each key separately. Sometimes the relations can be obtained for entire classes of keys X with the help of the same chains. This relations will be characterized by the same approximations $\tilde{\delta}_{\mathfrak{L}}$, $\tilde{p}_{\mathfrak{D}}$. Our point of view is difficult to challenge, for example, for a cipher such that on every round all components of current block are permuted by some permutation depending on the key X. The characteristics of linear and differential cryptanalysis used to determine subkey Z are the random values with respect to random $X \in V_K$. These random values should be averaged.

Cryptographers who applies linear and differential cryptanalisys are concentrated near two poles. One construct chains $\mathfrak{L}$ and $\mathfrak{D}$ of conformal local linear and differential relations. They maximize the values $\tilde{\delta}_{\mathfrak{L}}$ and $\tilde{p}_{\mathfrak{D}}$ by painstaking search and may not know true ratio between $\tilde{\delta}_{\mathfrak{L}}$, $\delta_{L'_{\mathfrak{L}},L''_{\mathfrak{L}}}$ and between $\tilde{p}_{\mathfrak{D}}$, $p_{D'_{\mathfrak{D}},D''_{\mathfrak{D}}}$. Other cryptographers are trying to prove theoretically the proximity of $\tilde{\delta}$, $\tilde{p}$ and δ, p. Sometimes they do it in general case. They use different probability-theoretic models and do not care if the model is relevant to the analyzed cipher. Such stadies may lead to false conclusions.

9 The importance of linear and differential cryptanalysis for cipher design.

We may consider the problem of minimizing parameters $\tilde{\delta}$ and $\tilde{p}$ in cipher design as a formalization of a rule that always exists to make ciphers nonlinear as much as possible. This is achieved by decreasing the absolute values of linear and differential characteristics and by increasing the degrees of diffusion θ and θ' of the linear medium. The latter explains the transition from SP-networks

(in which Shannon's idea of confusion and diffusion were realized in the second half of the last century [3]) to XSL-ciphers. A new cipher design technique has appeared. This technique is based on guaranteeing high diffusion degrees θ and θ' of cipher's linear medium. Cipher construction starts with the construction of a general structure, i.e. a linear medium. The nonlinear mappings are not specified in the preliminary stage of cipher design. They are considered in general form. The diffusion degrees of linear medium are evaluated before the specification of nonlinear mappings. The higher diffusion degrees, the easier it is to guarantee the security of a cipher against linear, differential and other techniques of cryptanalysis due to careful selection of nonlinear mappings included in the functional scheme. The degrees of diffusion θ and θ' of the linear medium (along with the number of rounds and the cardinality of the key set) are unusual characteristics that are easy to estimate and at the same time allow to make conclusions about the security of the cipher.

The diffusion degrees θ and θ' of XSL-cipher with two rounds are the coefficients of diffusion ρ_Λ and ρ'_Λ for separate nonsingular linear transformation $\Lambda \in GL(n, 2)$ [2]. Similar linear medium's diffusion degrees related to s-dimentional linear cryptanalysis are also important. They are more exact diffusion characteristics of permutation matrices used in SP-networks then θ and θ' [4].

Another representation of the methods of constructing probabilistic linear and differential relations without separation of the cipher's linear medium can be found, for example, in [5]–[13].

References

[1] Erokhin A. V., Malyshev F. M., Trishin A. E., "Multidimentional linear method and diffusion characteristics of linear medium of ciphering transform", *Mat. Vopr. Krypt.*, **8**:4 (2017), 29–62.

[2] Malyshev F. M., "The duality of differential and linear methods in cryptography", *Mat. Vopr. Krypt.*, **5**:3 (2014), 35–47.

[3] Massey J. L., "An introduction to contemporary cryptology", *Pro-ceedings of the IEEE*, **76**:5 (1988), 533–549.

[4] Malyshev F. M., Trifonov D. I., "Diffusion properties of XSLP-ciphers", *Mat. Vopr. Krypt.*, **7**:3 (2016), 47–60.

[5] Biham E., Shamir A., "Differential cryptanalysis of DES-like crypto-systems", *LNCS*, Crypto'90, **537**, 1991, 2–21.

[6] Biham E., Shamir A., "Differential cryptanalysis of DES-like crypto-systems", *J. Cryptology*, **4**, 1991, 3–72.

[7] Matsui M., "Linear cryptanalysis method for DES Cipher", *LNCS*, EUROCRYPT'93, **765**, 1994, 386–397.

[8] Matsui M., "The first experimental cryptanalysis of the Data Encryption Standard", *LNCS*, Crypto'94, **839**, 1994, 1–11.

[9] Biham E., "On Matsui's linear cryptanalysis", *LNCS*, EUROCRYPT'94, **950**, 1995, 341–355.

[10] Matsui M., "On correlation between the order of S-boxes and the strength of DES", *LNCS*, EUROCRYPT'94, **950**, 1995, 366–375.

[11] Nyberg K., "Linear approximation of block ciphers", *LNCS*, EUROCRYPT'94, **950**, 1995, 439–444.

[12] Daemen J., Govaerts R., Vandewalle J., "Correlation matrices", *LNCS*, FSE'94, **1008**, 1995, 275–285.

[13] Borst J., Preneel B., Vandewalle J., "Linear cryptanalysis of RC5 and RC6", *LNCS*, FSE'99, **1636**, 1999, 16–30.

[14] Daemen J., Rijmen V., "The Design of Rijndael: AES – The Advanced Encryption Standard", 2002.

The CTR Mode with Encrypted Nonces and Its Extension to AE

Sergey Agievich

Research Institute for Applied Problems of Mathematics and Informatics
Belarusian State University, Belarus
agievich@bsu.by

Abstract

In the modified CTR (Counter) mode known as CTR2, nonces are encrypted before constructing sequences of counters from them. This way we have only probabilistic guarantees for non-overlapping of the sequences. We show that these guarantees, and therefore the security guarantees of CTR2, are strong enough in two standard scenarios: random nonces and non-repeating nonces. We also show how to extend CTR2 to an authenticated encryption mode which we call CHE (Counter-Hash-Encrypt). To extend, we use one invocation of polynomial hashing and one additional block encryption.

Keywords: CTR mode, authenticated encryption, block cipher, polynomial hashing, gamma overlapping.

1 Preliminaries

Let E be a block cipher with block size n and key space $\mathcal{K}$. It is a multiset consisting of permutations $E_K \in \mathrm{Perm}(n)$ which are indexed by secret keys $K \in \mathcal{K}$.

Here $\mathrm{Perm}(n)$ is the set of all permutations over $\{0,1\}^n$. Elements of $\{0,1\}^n$ are called blocks. Let $N = 2^n$ denote their number.

We also denote by $\{0,1\}^*$ the set of all binary words of finite length. For a word $u \in \{0,1\}^*$, let $|u|$ be its length. If u, v are words of the same length, then $u \oplus v$ is their bitwise modulo 2 sum (XOR). For a permutation $\pi \in \mathrm{Perm}(n)$, let π^i be its ith compositional power (π^0 is the identity permutation). Denote by $m^{[i]}$ the ith factorial power of a positive integer m: $m^{[i]} = m(m-1)\dots(m-i+1)$.

To extend the action of E from $\{0,1\}^n$ to $\{0,1\}^*$, encryption modes are used. One of the most popular is CTR. In this mode, a unique nonce $S \in \{0,1\}^n$ is repeatedly transformed by a public permutation `next`. The resulting sequence

$$C_1 = S,\ C_2 = \texttt{next}(C_1),\ C_3 = \texttt{next}(C_2), \ldots$$

is encrypted using $E_K \in E$ to get the blocks

$$\Gamma_1 = E_K(C_1),\ \Gamma_2 = E_K(C_2), \ldots .$$

To encrypt a plaintext $X \in \{0,1\}^*$, the first $\lceil |X|/n \rceil$ blocks are used. They are concatenated and then truncated to $|X|$ bits. The resulting word $\Gamma \in \{0,1\}^{|X|}$ is XORed with X to produce a ciphertext

$$Y = X \oplus \Gamma.$$

In the Soviet standard GOST 28147 [7], the word Γ is called a *gamma*. That is why the notations. The blocks $C_1, C_2, \ldots$ are usually called *counters*. That is why CTR (Counter).

Suppose that in two encryption sessions, gammas Γ and Γ' overlap. Then an adversary who has intercepted a plaintext-ciphertext pair (X, Y) in one session can restore $\Gamma = X \oplus Y$ and then partially reconstruct X' from $Y' = X' \oplus \Gamma'$ in the parallel session. Thereby, a gamma overlapping is considered a compromise of the CTR encryption.

To avoid overlapping, a permutation `next` is chosen to have long disjoint cycles in its cycle decomposition. The nonces S of different sessions are picked from different cycles or a new nonce continues the cycle (actually, the sequence of counters) from the previous session. This approach, implemented in the standards [6, 8, 9], ensures that all counters in all sessions are unique. In other words, there are no collisions between counters and gamma overlapping is certainly impossible.

Unfortunately, such strict guarantees of no collisions / non-overlapping force the nonce management to be rather complicated. One has to use a safe monotonous timer to generate nonces or a rewritable memory to store them between sessions. Both options can be difficult to implement on some cryptographic devices. The third option, random generation of nonces, does not match the approach, at least it is not allowed in the mentioned standards.

Another approach, probabilistic guarantees of gamma non-overlapping, was

proposed in GOST 28147 and repeated in [15], where a nonce S is first encrypted and then transformed by `next`:

$$C_1 = \texttt{next}(E_K(S)),$$

not $C_1 = S$. (To be completely accurate, GOST's `next` is not a permutation: it acts bijectively on only a $2^{n/2}(2^{n/2}-1)$-element subset of $\{0,1\}^n$, $n = 64$.) The similar scheme

$$C_1 = E_K(S)$$

was considered later by P. Rogaway in [14], where the corresponding encryption mode is called CTR2. We extend this name to the GOST case. It is natural because the main point there is nonce encryption, the optional invocation of `next` is not critical.

Nonce encryption has obvious drawbacks. First, it slightly decreases the overall effectiveness of the mode. Second, it throws C_1 at an unpredictable point of `next`'s cycle that may cause a collision with other counters.

On the other hand, the probability of collisions is controllable small under reasonable restrictions on the amount of data processed with a single key. We confirm this fact in Section 2 in terms of a game called "Battleship on a circle". A control over collisions allows us to prove the security of CTR2 in the CPA (Chosen Plaintext Attack) settings. This is done in Section 3. In a nutshell, we embed well-known or easily derived combinatorial estimates within the context of Provable Security. We examine two techniques for the nonce generation: random nonces and non-repeating nonces. Note that we do not require that the nonce management deterministically ensures uniqueness of all counters in all sessions and thus allow it to be more flexible.

An additional argument in favor of nonce encryption is that it provides an easy extension of the conventional CTR encryption to authenticated encryption (AE). In Section 4, we show how to build this extension using polynomial hashing and one additional invocation of E_K. We call the resulting scheme CHE, meaning the cascade Counter-Hash-Encrypt. It is actually one of two AE schemes briefly described in [1]. There the security of only authentication, not encryption, is considered. In this paper, we fill the gap. We also provide a detailed description of CHE.

Usually, in AE schemes based on polynomial hashing (perhaps the most famous of them is GCM [10]), a data-driven polynomial is evaluated at a secret

point which depends only on K. In some cases (including GCM), this point can be recovered with the subsequent compromise of all encryption sessions as soon as a nonce S is used twice. A distinctive feature of CHE is that the secret point depends on S. Due to this fact, a repetition of nonces in multiple encryption sessions compromises only these sessions without affecting others. Thus, CHE provides reasonable security guarantees against nonce-misusing. To the best of our knowledge, stronger guarantees, the so-called full nonce-misuse resistance where only completely identical sessions compromise each other, are only achieved through two passes over data what is difficult to maintain in many scenarios.

Further we assume that `next` is a full cycle or almost full cycle permutation. In other words, if M is the maximum cycle length of `next`, then $M \approx N$. Usually, $M = N$ which is achieved by interpreting blocks of $\{0,1\}^n$ as integers modulo N and incrementing these integers in `next`. Another option for `next` is to interpret $\{0,1\}^n$ as the binary field F of N elements. Let α be a primitive element of F and β be an arbitrary element. Then the permutation

$$\texttt{next}\colon \lambda \mapsto \alpha\lambda + \beta$$

decomposes into a cycle of length $M = N - 1$ and a loop at $\beta/(1-\alpha)$. We use this `next` in Section 4.

Finally, it should be mentioned that encrypting a nonce S we make the counters $C_1, C_2, \ldots$ secret. An adversary cannot reconstruct any input-output pair of E_K even after intercepting all the session data (S, X, Y). Blocking direct access to E_K complicates attacks to recover K, especially statistical and algebraic attacks which usually strongly depend on the complexity of the simplest accessible cryptographic component.

2 Battleship on a circle

"Battleship on a circle" is played by Navy and an adversary. A game field is a circle on which M points numbered from 0 to $M-1$ are placed. Navy deploys ships on the circle concealing their locations. A ship of displacement r_i (a positive integer) occupies r_i consecutive points. In total, q ships of overall diplacement r ($q \leqslant r \leqslant M$) are deployed. The adversary makes q shots on the ships.

Detailed rules of the game (see Figure 1 for example):

1. The adversary splits r into a sum $r_1 + r_2 + \ldots + r_q$ of positive integers and reports $r_1, r_2, \ldots, r_q$ to Navy.

2. Navy deploys ships at random points on the circle. The bow of the ith ship is placed at point $C_{i,1}$ and the whole ship occupies the segment $C_{i,1}, C_{i,1} + 1, \ldots, C_{i,1} + r_i - 1$ (additive operations are modulo M). Collisions of ships, that is, intersections of their segments may occur. In the case of a collision, Navy loses and capitulates. Let the event $\mathcal{D}_1$ mean no collisions.

3. If Navy has not capitulated, then the adversary makes q shots at different points $S_1, \ldots, S_q$ on the circle. If at least one shot hits a ship, then the adversary wins. If all the shots miss, which is fixed by the event $\mathcal{D}_2$, then Navy wins.

Further we consider two variants of the game: G_1 and G_2.

In G_1, the ship bows $C_{i,1}$ are chosen uniformly at random independently of each other. The shot points S_i are also chosen uniformly at random with the only restriction that they must be different. In other words, $(S_1, \ldots, S_q)$ is a random q-permutation of M numbers. There are $M^{[q]}$ ways to choose it.

In G_2, the bows also form a random q-permutation. Shot points are arbitrary distinct.

Let us immediately explain that the games G_1 and G_2 simulate attacks on CTR2 with random and non-repeating nonces respectively. Ships correspond to sequences of counters. The lengths of the sequences can be chosen by an adversary who needs only to keep the total length, that is, the total amount of plaintext-ciphertext data. A collision of ships trivially means a gamma overlapping. More subtle are shots. A hit means that a nonce coincides with one of the internal counters. We will explain further details in the next section.

We are interested in the probability that Navy wins: $\mathsf{P}\{\mathcal{D}_1\mathcal{D}_2\} = \mathsf{P}\{\mathcal{D}_2 \mid \mathcal{D}_1\}\,\mathsf{P}\{\mathcal{D}_1\}$.

Lemma 1. *In the games G_1 and G_2,*

$$\mathsf{P}\{\mathcal{D}_1\mathcal{D}_2\} \geqslant 1 - \frac{4qr - q^2 - 2r + q}{M}.$$

Proof. Let us start with G_1. Navy can deploy the fleet without collisions as follows:

1) put the bow of the first ship at any of M point on the circle;

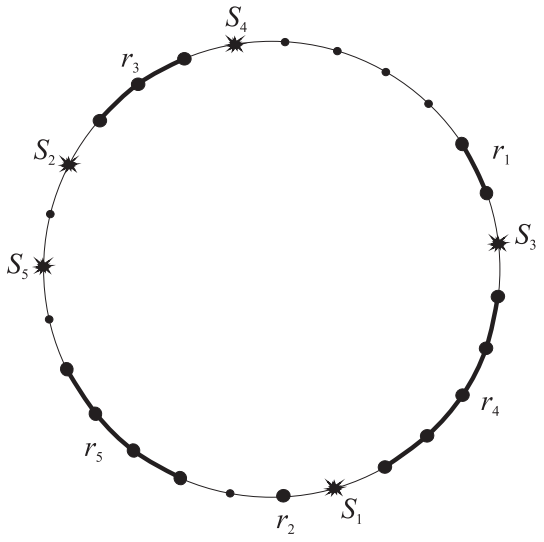


Figure 1: Battleship on a circle (Navy wins)

2) permute remaining ships in one of $(q-1)!$ ways;

3) choose q non-negative intervals between successive ships starting from the first one. The tuple of intervals is a weak q-composition of $M-r$ and, therefore, can be chosen in $\binom{M-r+q-1}{q-1}$ ways.

We repeat here the arguments of V. Nosov reported in [2]. The arguments yield:

$$\mathsf{P}\left\{\mathcal{D}_1\right\} = \frac{M(q-1)!\binom{M-r+q-1}{q-1}}{M^q} = \frac{(M-r+q-1)^{[q-1]}}{M^{q-1}} = \\ = \prod_{i=r-q+1}^{r-1}\left(1-\frac{i}{M}\right) \geqslant 1 - \sum_{i=r-q+1}^{r-1}\frac{i}{M} = 1 - \frac{(2r-q)(q-1)}{2M}.$$

Let $\bar{\mathcal{D}}_{2,i}$ be the event that the shot S_i is successful. We have

$$\mathsf{P}\left\{\bar{\mathcal{D}}_{2,i} \mid \mathcal{D}_1\right\} = \frac{r}{M}$$

and, therefore,

$$\mathsf{P}\left\{\mathcal{D}_2 \mid \mathcal{D}_1\right\} = 1 - \mathsf{P}\left\{\bar{\mathcal{D}}_{2,1} \cup \ldots \cup \bar{\mathcal{D}}_{2,q} \mid \mathcal{D}_1\right\} \geqslant \\ \geqslant 1 - \sum_{i=1}^{q}\mathsf{P}\left\{\bar{\mathcal{D}}_{2,i} \mid \mathcal{D}_1\right\} = 1 - \frac{qr}{M}.$$

In result,

$$\mathsf{P}\{\mathcal{D}_1\}\,\mathsf{P}\{\mathcal{D}_2 \mid \mathcal{D}_1\} \geqslant$$
$$\geqslant \left(1 - \frac{(2r-q)(q-1)}{2M}\right)\left(1 - \frac{qr}{M}\right) \geqslant 1 - \frac{4qr - q^2 - 2r + q}{2M},$$

which was to be proven.

When passing from G_1 to G_2, the probability $\mathsf{P}\{\mathcal{D}_1\}$ does not decrease and we can use the bound just derived on this probability. We can also reuse the bound on $\mathsf{P}\{\mathcal{D}_2 \mid \mathcal{D}_1\}$ and get the same overall bound as for G_1. □

An interesting question is what is the best strategy for an adversary in G_2. The partial answer is that with $qr \ll M$ the bound of Lemma 1 is almost reached when the adversary chooses $r_1 = r - q + 1$, $r_2 = \ldots = r_q = 1$ and shoots the circle with step r_1 starting from a random point. This tactic leads to the fact that to satisfy $\mathcal{D}_1\mathcal{D}_2$ the bow of the first ship must not occupy a continious segment of length $r_1 q$. The second ship must avoid $r_1 + q$ points, the third ship must avoid $r_1 + q + 1$ points and so on. In result,

$$\mathsf{P}\{\mathcal{D}_1\mathcal{D}_2\} = \left(1 - \frac{r_1 q}{M}\right)\prod_{i=1}^{q-1}\left(1 - \frac{r_1 + q + i - 1}{M - i}\right) \approx$$
$$\approx 1 - \frac{r_1 q}{M} - \sum_{i=1}^{q-1}\frac{r_1 + q + i - 1}{M}.$$

The right part coincides with the bound of the lemma. Approximately the same probability will be achieved, if the adversary chooses $r_1 = \ldots = r_{q-1} = \lfloor r/q \rfloor$ and shoots again with step r_1.

Finally, let us point out a fact that will be used in Section 4. Suppose that the rules of the game are relaxed and an adversary is allowed to choose zero r_i or, in other words, reduce the number of ships while maintaining their total displacement r. Analyzing the proof of the lemma, we conclude that the probability $\mathsf{P}\{\mathcal{D}_1\}$ increases with this reduction and, therefore, the bound on $\mathsf{P}\{\mathcal{D}_1\mathcal{D}_2\}$ becomes better. Of course, this bound will be even better, if the adversary reduces the total diplacement r.

3 Security of CTR2

To approve the security of CTR2, we use the standard notions sketched below (see [13] for further details and references).

1. An adversary (probabilistic algorithm) A gains access to an encryption oracle O. The adversary interacts with O using the following interface. It chooses a plaintext $X \in \{0,1\}^*$ and a nonce $S \in \{0,1\}^n$, sends the oracle the pair (X, S) and receives a ciphertext $Y \in \{0,1\}^{|X|}$. The adversary must use this interface following one of the two contracts: the nonces S are either chosen uniformly independently at random (the *random nonces* contract) or they are arbitrary distinct (the *non-repeating nonces* contract). Empty plaintexts are not allowed in both contracts.

2. The oracle can be implemented in two ways. In the first (real) implementation, O actually performs the CTR2 encryption using a permutation E_K chosen at random from E. This implementation is denoted by $\mathrm{CTR2}[E_K]$. In the second (ideal) implementation, O picks Y uniformly at random from $\{0,1\}^{|X|}$. This implementation is denoted by ρ.

3. The adversary sends O arbitrary queries, receives and analyzes corresponding answers. Its task is to distinguish the real implementation from ideal. The adversary returns 1 (real) or 0 (ideal). Let A^O be the output of A.

4. The distinguishing capabilities of A are characterized by the advantage

$$\mathbf{Adv}^{\texttt{ind-cpa}}_{\mathrm{CTR2}[E]}(A) = \left|\mathbf{P}\left\{A^{\mathrm{CTR2}[E_K]} = 1\right\} - \mathbf{P}\left\{A^{\rho} = 1\right\}\right|.$$

   The probabilities here are over the random tape of A and over the random choice of K and ρ. If $\mathbf{Adv}^{\texttt{ind-cpa}}_{\mathrm{CTR2}[E]}(A)$ is small, then the two implementations are hard to distingush, which means the security of CTR2 based on E relative to A. The used abbreviation `ind-cpa` covers the notion of indistinguishability and CPA settings: the adversary has access to the encryption oracle, but not the decryption one.

Let us make a standard simplification replacing E_K, a random representative of E, with π, a random representative of $\mathrm{Perm}(n)$. This replacement turns $\mathbf{Adv}^{\texttt{ind-cpa}}_{\mathrm{CTR2}[E]}(A)$ into the advantage

$$\mathbf{Adv}^{\texttt{ind-cpa}}_{\mathrm{CTR2}[\mathrm{Perm}(n)]}(A) = \left|\mathbf{P}\left\{A^{\mathrm{CTR2}[\pi]} = 1\right\} - \mathbf{P}\left\{A^{\rho} = 1\right\}\right|.$$

The replacement is motivated by the general assumption that permutations of a secure E are hard to distinguish from random ones. The replacement is accompanied by a penalty (another advantage) which characterizes indistinguishability between random representativies of E and $\mathrm{Perm}(n)$. This penalty is formal in nature (it is never estimated), we do not specify it here for simplicity.

For given non-negative integers q and r, $q \leqslant r$, we are interested in estimating

$$\max_A \mathbf{Adv}^{\mathtt{ind\text{-}cpa}}_{\mathrm{CTR2}[\mathrm{Perm}(n)]}(A),$$

where the maximum is taken over all adversaries that make q queries to O and the total length of plaintexts X in these queries is equal to r. The length is specified in blocks, possibly incomplete last. Incomplete blocks of different plaintexts are counted separately.

The advantage of a reasonable A cannot increase if some full block is cut to incomplete. Therefore, we can assume without loss of the maximum advantage that all plaintexts and ciphertexts consist of full blocks.

Let us write again how CTR2 works, that is, how plaintexts $X_1, \ldots, X_q$ and nonces $S_1, \ldots, S_q$ are transformed into ciphertexts

$$Y_i = \mathrm{CTR2}[\pi](X_i, S_i), \quad i = 1, \ldots, q.$$

Let X_i consist of blocks $X_{i,1}, \ldots, X_{i,r_i}$, $i = 1, \ldots, q$. where $r_i > 0$ and $r_1 + \ldots + r_q = r$. The corresponding blocks of the ciphertext Y_i are

$$Y_{i,j} = X_{i,j} \oplus \pi(C_{i,j}),$$

where

$$C_{i,1} = \mathtt{next}^c(\pi(S_i)), \quad C_{i,2} = \mathtt{next}(C_{i,1}), \quad \ldots, \quad C_{i,r_i} = \mathtt{next}(C_{i,r_i-1}).$$

Here c is an integer parameter of the mode. It equals 0 (the original CTR2) or 1 (GOST). In this section, the choice of c is inessential. However, in the next section we use $c = 1$.

Lemma 2. *Let N be a positive integer and q, r be non-negative integers such that $q + r \leqslant N$. Then*

$$\frac{1}{(N-q)^{[r]}} \geqslant \frac{1}{N^r}\left(1 + \frac{r(2q+r-1)}{2N}\right).$$

Proof. Consider three fractions: $1/(N+2q+r-1)$, $1/(N-q-i)$ and $1/(N-q-r+1+i)$, $0 \leqslant i \leqslant r-1$. The sum of their denominators is $3N$. Therefore, the product of the denominators does not exceed N^3, the product of the fractions is not less than $1/N^3$, and

$$\frac{1}{N-q-i} \cdot \frac{1}{N-q-r+1+i} \geqslant \frac{N+2q+r-1}{N^3} = \frac{1}{N^2}\left(1+\frac{2q+r-1}{N}\right).$$

Hence,

$$\left(\frac{1}{(N-q)^{[r]}}\right)^2 = \\ = \prod_{i=0}^{r-1}\left(\frac{1}{N-q-i} \cdot \frac{1}{N-q-r+1+i}\right) \geqslant \frac{1}{N^{2r}}\left(1+\frac{2q+r-1}{N}\right)^r,$$

from which the result follows. □

Theorem 1. *Let M, the maxumum cycle length of* `next`*, be at least $N-1$. Let an adversary A make at most q queries (X,S) with either random or non-repeating S. Let r be the total number of X's blocks in these queries. Then*

$$\mathbf{Adv}^{\textit{ind-cpa}}_{\textit{CTR2}[\textit{Perm}(n)]}(A) \leqslant \frac{r(r-1)}{2N} + \varepsilon,$$

where

$$\varepsilon = \\ = \max\left(0, \frac{r(r+2q-1)(4qr-q^2-2r+3q+2)}{4N^2} - \frac{(r-q)^2+r-3q-2}{2N}\right).$$

Proof. The bound obviously holds for $q+r > N$ (in this case $r > N/2$). Assume further that $q+r \leqslant N$, so that Lemma 2 can be applied.

Consider arbitrary nonempty plaintexts $X_1, \ldots, X_q$, r full blocks in total, random or arbitrary non-repeating $S_1, \ldots, S_q$, and random $\pi, Y_1, \ldots, Y_q$. When we say random, we mean that implied objects are chosen uniformly at random from prescribed domains, each object independently of others.

Let the event $\mathcal{B}$ means that all r blocks $\Gamma_{i,j} = X_{i,j} \oplus Y_{i,j}$ are distinct. For

the complementary event $\bar{\mathcal{B}}$, it holds that

$$\mathsf{P}\left\{\bar{\mathcal{B}}\right\} \leqslant \frac{r(r-1)}{2N}.$$

Introduce the probability

$$p = \mathsf{P}\left\{\mathrm{CTR2}[\pi](X_i, S_i) = Y_i \colon i = 1, \ldots, q \mid \mathcal{B}\right\}$$

and apply Patarin's "coefficients H" technique (see [12] and also [4, 5, 11]). According to this technique, if an inequality $p \geqslant (1-\varepsilon)/N^r$ with some $\varepsilon \geqslant 0$ holds, then the required advantage is upper bounded by the sum $\mathsf{P}\left\{\bar{\mathcal{B}}\right\} + \varepsilon$. It is remains to prove that ε from the statement of the theorem indeed satisfies the inequality.

Consider the following events, each new one provided that previous events occur.

The event $\mathcal{C}$: all blocks $\pi(S_i)$ fall into the largest cycle of `next`. The probability $p_C = \mathsf{P}\left\{\mathcal{C}\right\}$ equals either M^q/N^q in the case of random nonces or $M^{[q]}/N^{[q]}$ in the case of non-repeating nonces. In both cases,

$$p_C \geqslant \frac{M}{N}\left(1 - \frac{q}{N}\right).$$

Indeed,

$$\frac{M^q}{N^q} \geqslant \frac{M^{[q]}}{N^{[q]}} = \frac{M}{N} \cdot \frac{(M-1)^{[q-1]}}{(N-1)^{[q-1]}} = \\ = \frac{M}{N} \cdot \frac{M-q+1}{N-1} \geqslant \frac{M}{N} \cdot \frac{N-q}{N-1} \geqslant \frac{M}{N}\left(1 - \frac{q}{N}\right).$$

The event $\mathcal{D}$: all counters $C_{i,j}$ (they are all on the largest cycle according to $\mathcal{C}$) differ from each other and from nonces S_k. The probability of this event is already estimated in Lemma 1 of the previous section:

$$p_D = \mathsf{P}\left\{\mathcal{D} \mid \mathcal{C}\right\} \geqslant 1 - \frac{4qr - q^2 - 2r + q}{2M}.$$

We indeed satisfy the rules of the game described there, if we imagine that the initial counters $C_{i,1}$ are placed on the cycle randomly and after that, in the case of no collisions, the random permutation π either "generates" random distinct $S_i = \pi^{-1}(\mathtt{next}^{-c}(C_{i,1}))$ or implicitly transfers the given distinct S_i

into $\mathtt{next}^{-c}(C_{i,1})$. It may be that some S_i lies outside the cycle. In this case, the probability p_D only increases with respect to the probability treated in Lemma 1 and the bound of the lemma remains valid.

Consider the probability $p_{CD} = \mathsf{P}\{\mathcal{CD}\} = p_C p_D$. Dealing with the case $M = N - 1$, we get

$$p_{CD} \geqslant \left(1 - \frac{q}{N}\right)\left(\frac{M}{N} - \frac{4qr - q^2 - 2r + q}{2N}\right) \geqslant 1 - \frac{4qr - q^2 - 2r + 3q + 2}{2N}.$$

Obviously, this bound also holds for $M = N$.

The event $\mathcal{E}$: π maps $C_{i,j}$ to $\Gamma_{i,j}$. The previous events means that all $\Gamma_{i,j}$ are distinct, all $C_{i,j}$ are distinct, all S_i are distinct, $C_{i,j}$ differ from S_k, and q images of π at points S_i are already known. So there are $(N - q)!$ ways to determine remaining images of π and exactly $(N - q - r)!$ of them are in favor of $\mathcal{E}$. Therefore,

$$p_E = \mathsf{P}\{\mathcal{E} \mid \mathcal{BCD}\} = \frac{1}{(N-q)^{[r]}} \geqslant \frac{1}{N^r}\left(1 + \frac{r(r + 2q - 1)}{2N}\right).$$

Here we use Lemma 2.

In result,

$$p \geqslant \mathsf{P}\{\mathcal{CDE} \mid \mathcal{B}\} = p_{CD}p_E \geqslant$$
$$\geqslant \frac{1}{N^r}\left(1 - \frac{4qr - q^2 - 2r + 3q + 2}{2N}\right)\left(1 + \frac{r(2q + r - 1)}{2N}\right),$$

from which the expresion for ε follows. □

It is easy to verify that ε increases as a function of q for $q \leqslant r$. Substituting $q = r$ into the expressions of the theorem and slightly simplifying them, we obtain the following bound, uniform in q:

$$\mathbf{Adv}^{\mathtt{ind\text{-}cpa}}_{\mathrm{CTR2[Perm}(n)]}(A) \leqslant \frac{r^2 + r + 2}{2N} + \frac{r^2(9r^2 + 5)}{4N^2}.$$

For comparison, a similar advantage in the CTR mode is upper bounded by $r^2/(2N)$ (see [13]). Informally, the transition from CTR to CTR2 is accompanied by a penalty, the main contribution to which is made by the term $9r^4/(4N^2)$. This penalty is insignificant in the region $r^2 \ll N$, which is used in practice.

Note that the bound r^2/N on the CTR2 advantage is reported (without proof) in [14] for the case $M = N$.

4 CHE and its security

In this section, we extend CTR2 to the authentication encryption mode called CHE (Counter+Hash+Encrypt). The extended functionality of CHE is data authentication. CHE follows the Encrypt-then-MAC paradigm (first encrypt, then authenticate) which seems to be better than the MAC-then-Encrypt alternative (see [3]). Not only encrypted data is authenticated, but also associated data that is transmitted in the plain form. Thus, CHE belongs to the AEAD (Authentication Encryption with Associated Data) class of the AE schemes.

Let us interpret blocks of $\{0,1\}^n$ as elements of the finite field F of order N. Suppose that the usual correspondence between F and $\{0,1\}^n$ is used, when the addition in F is $\oplus$. Let

$$\texttt{next}(\lambda) = \alpha * \lambda \oplus \beta,$$

where α is a primitive element of F, β is a nonzero element. Hereinafter we make the multiplication sign explicit. As we have already noted, the maximum cycle length of `next` is $N-1$. Moreover, the powers $\texttt{next}^i$, $i = 1, 2, \ldots, N-2$, considered as polynomials over F all have nonzero constant terms.

The CHE mode is determined by the algorithms described below. Their inputs and outputs are: a plaintext $X \in \{0,1\}^*$, associated data $I \in \{0,1\}^*$, a key $K \in \mathcal{K}$, a nonce $S \in \{0,1\}^*$, a ciphertext $Y \in \{0,1\}^{|X|}$, an authentication tag $T \in \{0,1\}^n$. An arbitrary nonzero $T_0 \in F$ is used. The operation $\overset{n}{\leftarrow}$ means splitting a binary word into n-bit blocks preceded by padding to the block size. The reverse operation $\underset{m}{\leftarrow}$ means assembling a word from several blocks followed by truncation to m bits.

Algorithm Wrap	**Algorithm Unwrap**
Input: X, I, K, S. **Output**: Y, T. **Steps**: 1. $H \leftarrow E_K(S)$, $C \leftarrow H$, $T \leftarrow T_0$. 2. $(I_1, \ldots, I_{r'}) \xleftarrow{n} I$. 3. For $i = 1, 2, \ldots, r'$: (a) $T \leftarrow (T \oplus I_i) * H$. 4. $(X_1, \ldots, X_r) \xleftarrow{n} X$. 5. For $i = 1, 2, \ldots, r$: (a) $C \leftarrow \mathtt{next}(C)$; (b) $Y_i \leftarrow X_i \oplus E_K(C)$; (c) $T \leftarrow (T \oplus Y_i) * H$. 6. $Y \xleftarrow[\lvert X\rvert]{} (Y_1, \ldots, Y_r)$. 7. Encode $\lvert I\rvert$ and $\lvert X\rvert$ by $W \in \{0,1\}^n$. 8. $T \leftarrow (T \oplus W) * H$. 9. $T \leftarrow E_K(T)$. 10. Return (Y, T).	**Input**: Y, I, K, S, T. **Output**: X or $\perp$ (authentication error). **Steps**: 1. $H \leftarrow E_K(S)$, $C \leftarrow H$, $T' \leftarrow T_0$. 2. $(I_1, \ldots, I_{r'}) \xleftarrow{n} I$. 3. For $i = 1, 2, \ldots, r'$: (a) $T' \leftarrow (T' \oplus I_i) * H$. 4. $(Y_1, \ldots, Y_r) \xleftarrow{n} Y$. 5. For $i = 1, 2, \ldots, r$: (a) $T' \leftarrow (T' \oplus Y_i) * H$; (b) $C \leftarrow \mathtt{next}(C)$; (c) $X_i \leftarrow Y_i \oplus E_K(C)$. 6. $X \xleftarrow[\lvert Y\rvert]{} (X_1, \ldots, X_r)$. 7. Encode $\lvert I\rvert$ and $\lvert X\rvert$ by $W \in \{0,1\}^n$. 8. $T' \leftarrow (T' \oplus W) * H$. 9. $T' \leftarrow E_K(T')$. 10. Return X if $T = T'$ and $\perp$ otherwise.

It is assumed that in Step 10 of both algorithms, different pairs $(|I|, |Y|)$ give different words W and nonzero $|I|$ or $|Y|$ gives a nonzero W.

The algorithm Wrap can be explained in the following way.

C. First, the CTR2 encryption is performed: $Y \leftarrow \mathrm{CTR2}[E_K](X, S)$. The encrypted nonce $H = E_K(S)$ is used to build internal counters $\mathtt{next}^i(H)$, $i = 1, 2, \ldots$.

H. Second, a polynomial $f_{(Y,I)}(\lambda) \in F[\lambda]$ is implicitly constructed from the pair (Y, I). This polynomial has a positive degree, its constant term equals 0, different pairs give different polynomials. The polynomial is evaluated at the point H, the result $Z = f_{(Y,I)}(H)$ becomes a hash value of (Y, I).

E. Third, the hash value Z is encrypted and returned as T along with Y.

Suppose that $\deg f_{(Y,I)} \leqslant d$. In other words, at most $d-1$ blocks of I and Y are processed during a single invocation of polynomial hashing. Suppose further that $d < N-1$. The restrictions on structure and degree of the polynomials $f_{(Y,I)}$ and the form of `next` lead to the following estimates (see [1] for details):

$$\left.\begin{array}{l} \mathsf{P}\left\{f_{(Y,I)}(H) = f_{(Y',I')}(H') \mid H \neq H'\right\} \\ \mathsf{P}\left\{f_{(Y,I)}(H) = a\right\} \\ \mathsf{P}\left\{f_{(Y,I)}(H) = \mathtt{next}^i(H)\right\} \\ \mathsf{P}\left\{f_{(Y,I)}(H) = \mathtt{next}^i(H') \mid H \neq H'\right\} \end{array}\right\} \leqslant \frac{d}{N}.$$

Here $(Y,I) \neq (Y',I')$, $1 \leqslant i \leqslant d$, a is a fixed element of F, the probabilities are taken over independent random $H, H' \in F$. These estimates form the basis for justifying the security of CHE.

Dealing with the security, we keep the model introduced in the previous section. An adversary interacts with an oracle $O\colon (X,I,S) \mapsto (Y,T)$ which either implements the Wrap algorithm (the real implementation, CHE$[E_K]$) or generates $Y \in \{0,1\}^{|X|}$ and $T \in \{0,1\}^n$ at random (the ideal implementation, ρ). The adversary again follows one of the two contracts: random nonces or non-repeating nonces. Any of the word X and I can be empty, but not both.

An advantage of the adversary is defined in the standard way. We only change the abbreviation `ind-cpa` to `priv` (privacy). This corresponds to the tradition when moving from basic encryption to AEAD.

We again idealize E and replace its representative E_K with a permutation π chosen uniformly at random from Perm(n).

Theorem 2. *Let an adversary A make at most q queries (X,I,S) with either random or non-repeating S. Let r be the total number of X's and I's blocks in these queries. Then*

$$\mathbf{Adv}^{\mathtt{priv}}_{\mathrm{CHE}[\mathrm{Perm}(n)]}(A) \leqslant \frac{(r+q)(r+q-1)}{2N} + \varepsilon,$$

where

$$\varepsilon = \\ = \frac{(2r^2+9qr+2q^2-3r+2q+2)(r+q)(r+3q-1)}{4N^2} + \\ + \frac{r^2+5qr-q^2-2r+3q+2}{2N}.$$

Proof. We adapt the proof of Theorem 1 preserving notations and following the general line. Additional notations: I_i — associated data in the ith query, T_i — a tag in the ith answer, $H_i = \pi(S_i)$, $Z_i = f_{(Y_i,I_i)}(H_i)$.

Let d be the maximum degree of polynomials $f_{(Y_i,I_i)}$. In other words, $d-1$ is the maximum total amount of blocks in (X_i, I_i). It is clear that $(d-1)q \leqslant r$. Therefore, if $d \geqslant N-1$, then $r \geqslant N-2$ and the bound of the theorem obviously holds. Further we assume that $d < N-1$.

We preserve the probabilistic model of Theorem 1 assuming additionally that T_i are chosen uniformly independently at random. Now the event $\mathcal{B}$ additionally means that T_i are distinct and different from $\Gamma_{j,k}$. It is clear that

$$\mathsf{P}\left\{\bar{\mathcal{B}}\right\} \leqslant \frac{(r+q)(r+q-1)}{2N}.$$

For the probability

$$p = \mathsf{P}\left\{\mathrm{CHE}[\pi](X_i, I_i, S_i) = (Y_i, T_i) \colon i = 1, \ldots, q \mid \mathcal{B}\right\},$$

it is necessary to construct an inequality $p \geqslant (1-\varepsilon)/N^{r+q}$. To do this, we again deal with the events $\mathcal{C}$, $\mathcal{D}$, $\mathcal{E}$.

The semantics of $\mathcal{C}$ is not changed. In $\mathcal{D}$, we allow empty X_i and that the total number of plaintext blocks is less than r (r covers both plaintext and associated data blocks). As we have discussed at the end of Section 2, with these relaxations the bound on the probability p_D becomes even better.

In addition, we block in $\mathcal{CD}$ the following collisions:

collisions	quantity	probability
$Z_i = Z_j$	$q(q-1)/2$	$\leqslant d/N$
$Z_i = S_j$	q^2	$\leqslant d/N$
$Z_i = C_{j,k}$	$\leqslant qr$	$\leqslant d/N$

With this, the bound on p_{CD} becomes weaker:

$$p_{CD} \geqslant 1 - \frac{4qr - q^2 - 2r + 3q + 2}{2N} - \left(\frac{dq(q-1)}{2N} + \frac{dq^2}{N} + \frac{dqr}{N}\right).$$

Using the inequality $(d-1)q \leqslant r$, we get

$$p_{CD} \geqslant 1 - \frac{2r^2 + 9qr + 2q^2 - 3r + 2q + 2}{2N}.$$

In $\mathcal{E}$, we require that π not only maps $C_{i,j}$ to $\Gamma_{i,j}$, but also maps Z_i to T_i. The previous events mean that all preimages here are pairwise distinct, all images are pairwise distinct, and q images of π at points S_i that differ from $C_{j,k}$ and Z_j are already known. The total number of preimages is at most $r + q$. Therefore,

$$p_E \geqslant \frac{1}{(N-q)^{[r+q]}} \geqslant \frac{1}{N^{r+q}}\left(1 + \frac{(r+q)(r+3q-1)}{2N}\right).$$

In result,

$$p \geqslant p_{CD}p_E \geqslant$$
$$\geqslant \frac{1}{N^{r+q}}\left(1 - \frac{2r^2 + 9qr + 2q^2 - 3r + 2q + 2}{2N}\right)\left(1 + \frac{(r+q)(r+3q-1)}{2N}\right),$$

from which the expresion for ε follows. □

As in the previous section, ε increases as a function of q for $q \leqslant r$. Substituting $q = r$ into the expressions of the theorem and simplifying them, we get:

$$\mathbf{Adv}^{\texttt{ind-cpa}}_{\mathrm{CHE}[\mathrm{Perm}(n)]}(A) \leqslant \frac{9r^2 - r + 2}{2N} + \frac{26r^4}{N^2}.$$

References

[1] Agievich S., "EHE: nonce misuse-resistant message authentication", *Prikl. Discr. Mat*, **39** (2018), 33–41, https://eprint.iacr.org/2017/231.

[2] Babash A.V., Shankin G.P., *Cryptography*, Solon-Press, Moscow, 2007, In Russian.

[3] Bellare M., Namprempre C., "469–491", *J. of Cryptology*, **21**:4 (2008).

[4] Bernstein D., "A short proof of the unpredictability of cipher block chaining", 2005, http://cr.yp.to/papers.html#easycbc.

[5] Chen S., Steinberger J., "Tight Security Bounds for Key-Alternating Ciphers", *LNCS*, EUROCRYPT 2014., **8441**, ed. Nguyen P.Q., Oswald E., Springer, Berlin, Heidelberg, 2014, 327–350.

[6] Dworkin M., *NIST SP 800-38A. Recommendation for Block Cipher Modes of Operation: Methods and Techniques.*, National Institute of Standards and Technology (NIST) of the U.S., 2001.

[7] *GOST 28147. Cryptographic Protection for Information Processing Systems. Government Standard of the USSR. Government Committee of the USSR for Standards*, 1989, In Russian.

[8] *GOST R 34.13-2015. Information technology. Cryptographic data security. Block ciphers operation modes. Government Standard of the Russian Federation.*, Standardinform, Moscow, 2015, In Russian.

[9] *ISO/IEC 10116. Information technology — Security techniques — Modes of operation of an n-bit cipher. Third edition*, 2006.

[10] McGrew D.A., Viega J., "The security and performance of the Galois / Counter Mode (GCM) of operation", *LNCS*, Progress in Cryptology – INDOCRYPT 2004, **3348**, ed. Canteaut, A., Viswanathan, K., Springer, Berlin, Heidelberg, 2004.

[11] Nandi M., "Improved security analysis for OMAC as a pseudorandom function", *J. Math. Cryptol*, **3** (2009), 133–148.

[12] Patarin J., "The "Coefficients H" Technique", *LNCS*, SAC 2008, **5381**, ed. Avanzi R.M., Keliher L., Sica F., Springer, Berlin, Heidelberg, 2009, 328–345.

[13] Rogaway P., "Evaluation of Some Blockcipher Modes of Operation", Cryptography Research and Evaluation Committees (CRYPTREC), 2011, http://www.cryptrec.go.jp/estimation/techrep_id2012_2.pdf.

[14] Rogaway P., "Nonce-Based Symmetric Encryption", *LNCS*, FSE 2004, **3017**, ed. Roy B., Meier W., Springer, Berlin, Heidelberg, 2004, 348–358.

[15] *STB 34.101.31-2011. Information Technology and Security. Data Encryption and Integrity Algorithms. Standard of Belarus*, 2011, http://apmi.bsu.by/assets/files/std/belt-spec27.pdf, In Russian.

MGM Beyond the Birthday Bound

Alexey Kurochkin[1] and Denis Fomin[2]

[1] Certification Research Center LLC, Russia
[2] National Research University Higher School of Economics, Russia
playfootball77@mail.ru, dfomin@hse.ru

Abstract

In this work we study the security of the prospective Russian authenticated encryption with associated data mode that is known as MGM. We examine the mode properties under the condition that we have $\mathcal{O}\left(2^{n/2}\right)$ queries, where n is the state size of the used block cipher. Two attacks that are based on birthday paradox are proposed.

Keywords: authentication, birthday paradox, AE, AEAD, MGM.

1 Introduction

The Multilinear Galois Mode (MGM) is an authenticated encryption with associated data (AEAD) block cipher mode. It was originally proposed in [1] and was fully described later in [2]. MGM mode was developed by the Technical Committee for standardization "Cryptography and Security Mechanism" (TC-26) and now is a prospective Russian standard of AEAD mode [3].

In 2019 the MGM mode was analysed in the paradigm of provable security [4]. That work shows that the privacy and authenticity of MGM mode is provably guaranteed (under security of the used block cipher) up to the birthday paradox bound. The modern level of cryptography allows us to build AEAD modes that are secure beyond the birthday bound [5, 6, 7, 8, 9]. Thus, on the one hand, it has been shown that MGM has so-called $n/2$-bit security, but on the other hand, no real attack has been published so far even in the unlimited amount of queries.

This work proposes some attacks on the MGM mode in the case when we can get $\mathcal{O}\left(2^{n/2}\right)$ queries, where n is the state size of the block cipher. The work describes two attacks on the MGM mode:

– Nonce reusing attack. We present a simple theoretical attack that shows a way to proceed an authentication tag in case if nonce can be reused.

– Attack on Authenticity. We will the way to get an authentication tag for a special type of messages.

Both attacks are based on the well-known birthday paradox and these attacks do not threaten the security claims of the MGM mode.

2 Definition and Notations

Let V^n be the boolean (bit) vector space of dimension n. For a vector $x \in V^n$ we call the value $|x| = n$ the length of the vector x. For the brevity we denote the union of all vectors of arbitrary length V^i, $i \geqslant 0$ as V^*.

In this work we assume that any element of the vector space $x \in V_n$ can be represented as an element of the ring $\mathbf{Z}_{2^n}$, where "+" and "−" are respectively the plus and the minus operand in the ring. We also use the representation of x as an element of a finite field $\mathbb{F}_{2^n}$ $(\oplus, \otimes)$.

We denote $msb_l(x)$ and $lsb_l(x)$ respectively the most and the least l significant bits of a vector x.

For the brevity we define the following operators. Let $x, y \in V^{n/2}$ and $t \in \mathbf{Z}_{2^{n/2}}$:

$$
\begin{aligned}
(x\|y) \boxplus^l t &= (x + t\|y)\,; \\
(x\|y) \boxminus^l t &= (x - t\|y)\,; \\
(x\|y) \boxplus^r t &= (x\|y + t)\,; \\
(x\|y) \boxminus^r t &= (x\|y - t)\,.
\end{aligned}
$$

First, we remind the MGM mode description following the description in [3]. Let e be a block cipher with block length n and $K \in V^k$ be a key. Denote by $e_K(x)$ the encryption of a plaintext block x under the key k.

The input of the MGM mode based on a cipher e is (K, N, P, A), where:

– $K \in V^k$ – key ;

– $N \in V^{n-1}$ – nonce ;

– $P \in V^*$, $0 \leqslant |P| \leqslant 2^{n/2}$ – plain text ;

– $A \in V^*$, $0 \leqslant |A| \leqslant 2^{n/2}$ – associated data.

The length of the plain text and associated data must be less than $|P| + |A| \leqslant 2^{n/2}$. The output of the mode is $(N, A, C, , T)$, where:

- $C \in V^*$, $|P| = |C|$ is a cipher text;
- $T \in V^m$ is an authenticating tag.

The cipher text and the authenticating tag are calculated as follows:

1. The plain text and associated data are divided into equal blocks of length n (perhaps except the last ones):

$$A = A_1 \| \ldots \| A_h^*,\ A_j \in V_n, A_h^* \in V_t,$$
$$P = P_1 \| \ldots \| P_q^*,\ P_i \in V_n, P_q^* \in V_u,$$

   where $j = 1, 2, \ldots, h-1$, $i = 1, 2, \ldots, q-1$, $1 \leqslant u \leqslant n$, $1 \leqslant t \leqslant n$ and $h + q > 0$.

2. The cipher text is calculated as follows:

$$\begin{cases} Y_1 \ = \ e_K(0\|N), \\ Y_i = Y_{i-1} \boxplus^r 1, \quad 2 \leqslant i \leqslant q, \\ C_i = P_i \oplus e_K(Y_i), \quad 1 \leqslant i \leqslant q-1, \\ C_q^* = C_q^* \oplus MSB_u(e_K(Y_q)), \end{cases} \tag{1}$$

3. The blocks A_h^* and C_q^* are padded till the full block size if needed:

$$\begin{cases} A_h = A_h^* \| 0^{n-t}, \\ C_q = C_q^* \| 0^{n-u}. \end{cases} \tag{2}$$

4. The authenticating tag is calculated as follows:

$$T = e_K \left(\sum_{i=1}^{h} H_i \otimes A_i \oplus \sum_{j=1}^{q} H_{h+j} \otimes C_j \oplus H_{h+q+1} \otimes \Big(|A| \big\| |C| \Big) \right),$$

   where $H_i = e_K(Z_i)$, and values Z_i, $i = 1, 2, \ldots$, are defined as follows:

$$\begin{cases} Z_1 = e_K(1\|N), \\ Z_i = Z_{i-1} \boxplus^l 1,\ 2 \leqslant i \leqslant h+q+1. \end{cases} \tag{3}$$

Tag verification and decryption occurs in a similar way. For the brevity we denote $\Big(|A|\|C|\Big)$ as L and call it "length tag".

3 Nonce reusing

In this section we will show that if we have two different messages with the same authenticating tags and if we in addition have a possibility to authenticate an arbitrary message, it is possible to calculate the authenticating tag for the special message.

Let we have two messages received using MGM mode under the same key K: (N_1, A_1, C_1, T_1), (N_2, A_2, C_2, T_2). And we also suppose that $T_1 = T_2$:

$$\sum_{i=1}^{h_1} H_{1,i} \otimes A_{1,i} \oplus \sum_{j=1}^{q_1} H_{1,h_1+j} \otimes C_{1,j} \oplus H_{1,h_1+q_1+1} \otimes L_1 = \\ = \sum_{i=1}^{h_2} H_{2,i} \otimes A_{2,i} \oplus \sum_{j=1}^{q_2} H_{2,h_2+j} \otimes C_{2,j} \oplus H_{2,h_2+q_2+1} \otimes L_2, \quad (4)$$

where L_1 and L_2 are the length tags. We also suppose that $L_1 = \big(n \cdot k_1^1 \| n \cdot k_2^1\big)$ and $L_2 = \big(n \cdot k_1^2 \| n \cdot k_2^2\big)$.

If the left and the right sides of the equation (4) are multiplied by the same element α of the finite field $\mathbb{F}_2^n$ then we get the correct equation. Let's make the following message:

$$\begin{cases} A'_i = A_{1,i} \otimes \alpha, \quad 1 \leqslant i \leqslant h_1, \\ A'_{i+h_1} = C_{1,i} \otimes \alpha, \quad 1 \leqslant i \leqslant q_1; \end{cases} \quad (5)$$

where α can be calculated from the equation:

$$L_1 \otimes \alpha = \big(n \cdot k_1^1 \| n \cdot k_2^1\big) \otimes \alpha = \big(0 \| n \cdot (k_1^1 + k_2^1)\big), \ k_i^j \in \mathbb{Z}, i, j \in \{1, 2\}.$$

We suppose that it's possible to request authenticating tag for the associated data $A' = A'_1 \| \ldots \| A'_{q_1+h_1}$:

$$T' = e_K \left(\left(\sum_{i=1}^{h_2} H_{2,i} \otimes A_{2,i} \oplus \sum_{j=1}^{q_2} H_{2,h_2+j} \otimes C_{2,j} \oplus H_{2,h_2+q_2+1} \otimes L_2 \right) \otimes \alpha \right)$$

Let's examine the value $L_2 \otimes \alpha = (s_1 \| s_2)$. To carry out the attack values L_1 and L_2 should meet the following conditions:

1. $(s_1, s_2) = (n \cdot k_1'' \| n \cdot k_2'')$. The propability of meeting this condition is about

$$P\left(n|s_1, n|s_2\right) \approx n^{-2}.$$

2. $k_1'' + k_2'' < 2^{n/2}$.

$$P\left(s_1 + s_2 < 2^{n/2}\right) = \frac{1}{2^{n/2}} \sum_{i=0}^{2^{n/2}-1} P\left(i + s_2 < 2^{n/2}\right) =$$

$$= \frac{1}{2^n} \sum_{i=0}^{2^{n/2}-1} (2^{n/2} - i) = 1/2 + \frac{1}{2^{n/2+1}}.$$

3. $h_2 + q_2 \leqslant k_1'' + k_2''$.

$$P\left(h_2 + q_2 \leqslant k_1'' + k_2''\right) \approx 1/2.$$

So with the probability $P_S = (2n)^{-2}$ the value T' will be a correct authenticating tag for message (N_2, C'', A'', T').

$$\begin{cases} A_i'' = B_i, & 1 \leqslant i \leqslant k_1''; \\ C_i'' = B_{i+k_1''}, & 1 \leqslant i \leqslant h_2 + q_2; \\ C_i'' = 0, & h_2 + q_2 < i \leqslant k_2''. \end{cases}$$

As example: $n = 128$, $k_1^1 = 38$, $k_1^2 = 48$, $k_2^1 = 39$, $k_2^2 = 111$. Then

$$L_1 = 0x13000000000000001800, L_2 = 0x13800000000000003780.$$

If $\mathbb{F}_{2^{128}} = \mathbb{F}_2[x]/(x^{128} + x^7 + x^2 + x + 1)$ then

$$\alpha = L_1^{-1} \otimes 0x2b00 = 0x3c3f14aa0b4941e598bccb28951fe354$$

and $k_1'' = 0xe6b0864cb7a77080$, $k_2'' = 0x8cb53060fc31c100$.

At the same time if $L_1 = L_2$ then $P_S = 1$ and it is possible to implement the following attack.

Nonce reusing attack

Let all the messages have the following structure: (N_i, A_i, C_i, T_i), where $|A_i| = |A_j|$, $|C_i| = |C_j|$, and all messages are calculated under the same key K.

1. Request D messages. With the probability $p \approx 1 - \exp\left\{-\frac{(D-1)^2}{2^{n+1}}\right\}$ two messages with numbers i and j such as $T_i = T_j$ will appear.
2. Make a new message from (N_i, A_i, C_i, T_i) using the equation (5).
3. Ask to authenticate this message.
4. Get the message $(K, N_i, \alpha \otimes (A_i \| C_i), T')$.
5. Make a new message with correct authenticated tag $(K, N_j, \alpha \otimes (A_j \| C_j), T')$.

4 Authenticity of data attack

4.1 How to get H_i?

All values H_i are hidden under encryption algorithm and it's quite difficult to get at least one. In this section we'll propose a way to find such values. As in the past we assume that all message are calculated under the same key K.

First, we show how to find x and y, such as $x, y \in V^n$: $e_K(x) = y$.

Let's consider the following message (N_1, A_1, C_1, T_1), where $|A_1| = 0$, $C_1 = 0$, and $|C_1|$ is equal to 1. Then

$$T_1 = e_K(H_2 \otimes 1) = e_K\left(e_K\left(e_K\left(1||N_1\right) \boxplus^l 1\right)\right).$$

Let (K, N_2, A_2, C_2, T_2) be another message and $P_1 \oplus C_1 = e_K(Y_1) = e_K\left(e_K\left(0||N_2\right)\right)$ is equal to authenticating tag T_1. Then we can argue that:

$$e_K\left(e_K\left(e_K\left(1||N_1\right) \boxplus^l 1\right)\right) = e_K\left(e_K\left(0||N_2\right)\right) \Rightarrow e_K(1||N_1) = 0||N_2 \boxminus^l 1.$$

Then in our notations $x = 1||N_1$ — known value, $y = 0||N_2 \boxminus^l 1$ — also known value. Thus, we know the equality $e_K(x) = y$.

According to the MGM mode description $e_K\left(1\|N\right) = Z_1$. Let $lsb_{\frac{n}{2}}(N_1) = lsb_{\frac{n}{2}}(N_2)$ there is such a value t: $t \in \mathbb{Z}$, $t < 2^{n/2}$:

$$Z_{t-1} = e_K(1||N_1) \boxplus^l t = 0||N_2 \boxplus^l (t-1) = (1||N_1),$$

and it is possible to calculate

$$H_{t-1} = e_K\left(Z_{t-1}\right) = e_K(e_K(1||N_1) \boxplus^l t) = e_K(1||N_1) = 0||N_2 \boxminus^l -1.$$

And in our notations $e_K(x) = y$, where $y = H_{t-1} = 0||N_2 \boxminus^l -1$ and $x = Z_{t-1} = e_K(1||N_1) \boxplus^l t$.

If there is a limit of$|P| + |A| \leqslant 2^{n/2-\Delta}$, then (since the operation $\boxplus^l$ changes only the left side of $0\|N_2$) the equation $e_K(x) = y$ can be obtained with probability

$$P_1 = P\left(t \leqslant 2^{n/2-\Delta}\right) = 2^{-\Delta}.$$

At the same time, if there are no limitations on the amount of processed material, this equation exists with the probability equal to $P_1 = 1$.

4.2 Double H attack

Let's suppose that we have two different values $H_i = e_K(Z_i)$, $H_j = e_K(Z_j)$ and $e_K^{-1}(H_i)$, $e_K^{-1}(H_j)$ for some values $i < j < 2^{n/2}$. We also assume that $lsb_{\frac{n}{2}}(Z_i) = lsb_{\frac{n}{2}}(Z_j)$.

Let $h, q \in \mathbb{N}_0$ and $h + q + 1 = j$ then we can form the following message S (value x will be determined later):

$$S = \left(\underbrace{0, 0, \ldots, 0}_{i-1}, x, \underbrace{0, 0, \ldots, 0}_{j-i-2}\right) = \left(\underbrace{A_1, \ldots, A_h, C_1, \ldots, C_q}_{j-1}\right).$$

The authenticating tag T of the message S is calculated as follows:

$$T = e_K(x \otimes H_i \oplus L \otimes H_j),$$

where $L = (l(A)\|l(C))$ — length tag of message S.

Fixing the values h and q we can calculate the value x using one of the following equations:

$$x \otimes H_i \oplus L \otimes H_j = e_K^{-1}(H_i);$$
$$x \otimes H_i \oplus L \otimes H_j = e_K^{-1}(H_j)$$

and authenticated tag will be equal to H_i and H_j respectively.

A pair of values h and q can be fixed by any of the j possible values and which means that we can calculate authenticating tag for $2 \cdot j$ messages without

knowing the secret key K and moreover, half of these messages will have $T = H_i$ and the other half will have authenticated tag equal to H_j. That also means that in case of $j > 1$ we can also find a collision.

Double H attack

We suppose that all $lsb_{\frac{n}{2}}(N'_i)$ and $lsb_{\frac{n}{2}}(N''_i)$ are equal.

1. Get m_1 messages (N'_i, A'_i, C'_i, T'_i), where $|A'_i| = 0$, $|C'_i| = n$:
$$M_1 = \{Y_1(N_i)\}_{i=0}^{m_1} = \{e_K(e_K(0\|N_i))\}_{i=0}^{m_1}.$$

2. Get $2{\cdot}m_2$ messages $(N''_i, A''_i, C''_i, T''_i)$, where $|A''_i| = 0$, $|C''_i| = 1$. We suppose that about the half of these messages is equal to zero $C''_2 = 0$ (one bit) and we have
$$M_2 = \{T_j\}_{j=0}^{m_2} = \left\{e_K\left(e_K\left(e_K\left(1\|N_i\right)\boxplus^l\right)\right)\right\}_{j=0}^{m_2}.$$

3. With some probability P_2 we find two equalities:
$$e_K(1\|N_1) = 0\|N_2 \boxminus^l 1,$$
$$e_K(1\|N_3) = 0\|N_4 \boxminus^l 1,$$
$$lsb_{\frac{n}{2}}(N_1) = lsb_{\frac{n}{2}}(N_2) = lsb_{\frac{n}{2}}(N_3) = lsb_{\frac{n}{2}}(N_4).$$

4. In accordance with the section 4.1 with the probability P_1^2 we can find two pair of values: H_{t_1}, H_{t_2}, $e_K^{-1}(H_{t_1})$, $e_K^{-1}(H_{t_2})$.

5. Without loss of generality we suppose that $t_2 > t_1$. Fixing $h, q \in \mathbb{N}_0$ by any values such as: $h + q + 1 = t_2$ form the message:
$$S = (\underbrace{0, 0, \ldots, 0}_{i-1}, x, \underbrace{0, 0, \ldots, 0}_{t_2-t_1-2}) = (A_1, \ldots, A_h, C_1, \ldots, C_q),$$
where x is calculated as follows:
$$x \otimes H_{t_1} \oplus L \otimes H_{t_2} = e_K^{-1}(H_{t_1}),\ L = (h\|q).$$

6. The authenticating tag of message S is H_{t_1}.

4.3 Difficulty and Probability of Double H attack

Let's find the probability P_2 from section 4.2.

We have the sets M_1 and M_2 such that $|M_1| = m_1$, $|M_2| = m_2$. The elements of these sets are integers from the set: $\overline{0, 2^n - 1}$. We can assume that every set has no identical elements: $M_i = \left\{M_i^1, M_i^2, \ldots, M_i^{m_i}\right\}$ and $M_i^{j_1} = M_i^{j_2}$ if and only if $j_1 = j_2$.

The probability that we can find at least one identical element in the sets M_1 and M_2 can be calculated as follows:

$$p_1 = 1 - \frac{\binom{2^n - m_1}{m_2}}{\binom{2^n}{m_2}} \approx 1 - \exp\left\{-\frac{m_1 m_2}{2^n}\right\}.$$

At the same time, exactly one identical element will be found with probability:

$$p = 2^n \frac{\binom{2^n - 1}{m_1 - 1} \cdot \binom{2^n - m_1}{m_2 - 1}}{\binom{2^n}{m_1} \cdot \binom{2^n}{m_2}} \approx \frac{m_1 m_2}{2^n} \cdot \exp\left\{\frac{-1 + 2m_1 + 2m_2 - 2m_1 m_2}{2^{n+1}}\right\}.$$

And the required probability that more that one identical element will be found can be calculated using the equation $P_2 = p_1 - p$.

To implement this attack we need:

- $m_1 + 2 \cdot m_2$ queries;
- memory $\mathcal{O}(m_1)$.

The attack success probability is equal to $P_1^2 \cdot P_2$. In the case of the absence of restrictions on the amount of material processed the probability is equal to P_2.

Conclusion

In this paper we examined some aspects of the MGM AEAD mode and proposed two theoretical attacks that describe some properties of the studied mode.

Both attacks require about $\mathcal{O}\left(2^{n/2}\right)$ queries, with n the state size of used block cipher.

The core of the first attack is a possibility of manipulating length tag. If we have two messages with the same authenticating tag and if we can ask to

authenticate some associated data with repeated nonce we can make a message that haven't been ever encrypted and authenticated.

The core of the second attack is a possibility to find H_i which is used to make authenticating tag. If we have two values H_i and H_j, $i < j$, we can make j messages that have authenticating tag equal to H_i and j messages that have authenticating tag equal to H_j.

At the same time a constituent part of both attacks are birthday paradox and these attacks do not threaten the security claims of MGM [3].

Acknowledgement of the Reviewers

Authors would like to thank the CTCrypt 2019 reviewers for their helpful comments and efforts towards improving this article.

References

[1] Nozdrunov V., "Parallel and double block cipher mode of operation (PD-mode) for authenticated encryption", CTCrypt 2017, *Pre-proceedings*, 2017, 36–45.

[2] Nozdrunov V. and Shishkin V., "Multilinear Galois Mode (MGM)", *CFRG Draft*, 2018, https://datatracker.ietf.org/doc/draft-smyshlyaev-mgm.

[3] Federal Agency on Technical Regulating and Metrology, "Recommendations for standardization. Cryptography. Authentication encryption modes of block ciphers", 2018, In Russian.

[4] Akhmetzyanova L., Alekseev E., Karpunin G., and Nozdrunov V., "Security of Multilinear Galois Mode (MGM)", *Cryptology ePrint Archive*, **2019/123** (2019), https://eprint.iacr.org/2019/123.

[5] Datta N., Dutta A., Nandi M., Paul G., and Zhang L., "Single key variant of PMAC_Plus", *IACR Trans. Symm. Cryptol*, 2017, 268–305.

[6] Iwata T. and Minematsu K., "Stronger security variants of GCM-SIV", *IACR Trans. Symm. Cryptol*, 2016, 134–157, http://tosc.iacr.org/index.php/ToSC/article/view/539.

[7] Yasuda K., "The sum of CBC MACs is a secure PRF", *LNCS*, RSA 2010, **5985**, ed. Pieprzyk J., 366–381.

[8] Yasuda K., "A new variant of PMAC: Beyond the birthday bound", *LNCS*, CRYPTO 2011, **6841**, ed. Rogaway P., 596–609.

[9] Zhang L., Wu W., Sui H., Wang P., "Enhancing 3GPP-MAC beyond the birthday bound", *LNCS*, ASIACRYPT 2012, **7658**, ed. Wang X., Sako K., 296–312.

Improving OBDD Attacks Against Stream Ciphers

Matthias Hamann, Matthias Krause, and Alexander Moch

Universität Mannheim, Germany
{hamann, krause, moch}@uni-mannheim.de

Abstract

We present and discuss new algorithmic ideas for improving OBDD-attacks against stream ciphers, which compute the secret initial state by generating a sequence of $\mathcal{O}(n)$ ordered binary decision diagrams (OBDDs) of maximal width $\mathcal{O}(2^{\frac{1-\alpha}{1+\alpha}n})$, where n denotes the inner state length and $\alpha \in (0,1)$ the *compression rate* of the cipher. We propose and experimentally verify the following strategy of avoiding the huge storage demand of $\mathcal{O}(2^{\frac{1-\alpha}{1+\alpha}n})$. (1) Generate in parallel two OBDDs P and Q such that $P \wedge Q$ has only a few satisfying assignments. (2) Compute $(P \wedge Q)^{-1}(1)$, including the secret inner state, by a new breadth-first-search based algorithm. We show that this approach improves standard OBDD-attacks drastically.

Keywords: Symmetric Cryptography, Stream Ciphers, OBDD Attacks.

1 Introduction

Stream ciphers are symmetric encryption algorithms intended for the online encryption of plaintext bitstreams X which have to pass an insecure channel. The encryption is performed by bitwise addition of a keystream S, which is generated in dependence of a secret symmetric session key k and, possibly, a public initial value $\mathtt{IV}$. The legal recipient, who also knows k, decrypts the encrypted bitstream $Y = X \oplus S$ by generating S and computing $X = Y \oplus S$. In this paper, we consider KSG-based stream ciphers, i.e., stream ciphers that generate the keystream using a so-called keystream generator (KSG).

KSGs are stepwise working devices that can be formally specified by finite automata. KSGs are defined by an inner state length n and the corresponding set of inner states $\{0,1\}^n$, a state update function $\pi : \{0,1\}^n \to \{0,1\}^n$ and an output function $\mathtt{out} : \{0,1\}^n \to \{0,1\}$. Starting from an initial state $q_0 \in \{0,1\}^n$, in each clock cycle $i \geqslant 0$, the KSG produces a keystream bit $z_i = \mathtt{out}(q_i)$ and

changes the inner state according to $q_{i+1} = \pi(q_i)$. The output bitstream $S(q_0)$ is defined by concatenating all the outputs $z_1 z_2 z_3 \cdots$.

The main security requirement for stream ciphers is the following: when a secret initial state is chosen randomly, it must be hard to distinguish the generated keystream from a truly random bitstream. This implies the hardness of the problem S^{-1}: given a piece z of keystream of length $\ell \geqslant n$, compute an initial state q which generates z in the sense that z is a prefix of $S(q)$. In this paper, we focus on analyzing the security of KSGs with regard to S^{-1}-attacks, which try to solve the S^{-1}-problem for a given piece z of keystream.

The main building blocks of many KSG-constructions are so-called Feedback Shift Registers (FSRs). FSRs are defined over a number m of register cells and a feedback function $f : \{0,1\}^m \to \{0,1\}$. In each clock cycle, given the inner state $(b_1, \ldots, b_m)$, the bit b_1 is the output bit and the state changes to $(b_2, \ldots, b_{m-1}, f(b_1, \ldots, b_m))$. The FSR is called a linear FSR (LFSR) if the feedback function is $GF(2)$-linear, and it is called a nonlinear FSR (NFSR) otherwise.

The output sequences defined by LFSRs with a primitive connection polynomial have several very useful pseudorandomness properties, e.g., the maximal period of $2^m - 1$. However, an LFSR alone does not represent a secure KSG as the problem S^{-1} can be efficiently solved by inverting an $(n \times n)$-matrix.

Many KSG-constructions use a small number of FSRs (LFSRs or NFSRs) to generate a secret inner bitstream, which has to pass a nonlinear filter function to produce the output keystream. This allows to split the keystream generation process into two parts: (1) the generation of the inner bitstream $\mathtt{IB}(q)$ from an initial state q, consisting of the bits produced by the FSRs, and (2) the generation of the public output keystream $S(q) = C(\mathtt{IB}(q))$ that is generated from the secret inner bitstream $\mathtt{IB}(q)$ using an online compression function C. The compression rate of a keystream generator is defined to be $\alpha = 1/A$, where A denotes the average number of inner keystream bits needed to produce one output bit. Note that the secret initial state q is a part (often the prefix) of the inner bitstream $\mathtt{IB}(q)$.

Concerning S^{-1}-attacks against stream ciphers, one distinguishes short-keystream attacks (e.g., OBDD-attacks) and long-keystream attacks (e.g., time-memory-data tradeoff (TMD-TO) attacks). Short-keystream attacks solve the S^{-1}-problem where only a small sequence of keystream bits was observed. In this case, the sequence is usually not significantly longer than n. Long-keystream

attacks require a lot more keystream bits (e.g., $2^{n/2}$ for the TMD-TO attacks of Babbage [1] and Golić [5]) to solve the S^{-1}-problem.

Ordered binary decision diagrams (OBDDs) are a graph based data structure for representing Boolean functions. Due to their specific algorithmic properties, there is a wide range of practical applications of OBDDs, especially in the field of hardware verification (see, e.g., [9]). OBDD-attacks were introduced by Krause in [6] and further studied in, e.g., [7], [11], [4]. They represent the most efficient kind of short-keystream attack against stream ciphers. The number of necessary keystream bits nearly matches the information-theoretic lower bound, which corresponds to the unicity distance of the cipher.

OBDD-attacks refer to the decision if for a given initial state q, a given piece of inner bitstream y and a given piece of keystream z it holds that y is prefix of $\mathtt{IB}(q)$ and $C(y)$ is prefix of z. For many KSGs, this problem can be formulated as a set $\mathcal{R} = \{R_1, \ldots, R_t\}$ of easy relations between the q-bits and the y-bits, and between the y-bits and the z-bits, which all can be tested by small OBDDs.

OBDD-attacks compute the OBDDs $Q(I)$ for increasing subsets $I \subseteq \{1, \ldots, t\}$, which test if all relations R_i, $i \in I$, are fulfilled. They are based on certain algorithmic properties of OBDDs:

- The conjunction $P \wedge Q$ of two OBDDs can be computed within time and space $\mathcal{O}(|P| \cdot |Q|)$.
- OBDDs P can be efficiently minimized in time $\mathcal{O}(|P|)$.
- The width of a minimized OBDD P is bounded from above by the amount of satisfying assignments of P.

In the standard OBDD-attack, one generates a sequence $Q(I_1) \to Q(I_2) \to \cdots \to Q(I_s)$, where $Q(I_1)$ is small and the number of satisfying assignments of $Q(I_1)$ is bounded by $2^{(1-\alpha)n}$. For $j = 2$ to s, I_j is derived from I_{j-1} by adding one new relation from $\mathcal{R}$. This implies that on average $|Q(I_j)^{-1}(1)| \leqslant 2^{\alpha} \cdot |Q(I_{j-1})^{-1}(1)|$ and $\mathtt{width}\left(Q(I_j)\right) \leqslant \min\{2 \cdot \mathtt{width}\left(Q(I_{j-1})\right), |Q(I_j)^{-1}(1)|\}$. It follows that on average after $r = \frac{1-\alpha}{1+\alpha}n$ iterations the maximal width of $\mathcal{O}(2^{\frac{1-\alpha}{1+\alpha}n})$ is reached and that after $\frac{1}{\alpha}n$ iterations we obtain an OBDD for which the secret initial state is the only satisfying assignment, i.e., which solves the problem.

OBDD-attacks (and their generalization based on free binary decision diagrams (FBDDs)) yield the best known short-keystream attacks against sev-

eral practically used ciphers such as the A5/1-generator included in the GSM-standard [3], the E_0-generator included in the Bluetooth standard [2], or the self-shrinking generator [8].

1.1 Our Results

We propose the following modification of OBDD-attacks for circumventing the bottleneck consisting in the huge amount of storage needed for the intermediate OBDDs in the iterations near the critical round number $r = \frac{1-\alpha}{1+\alpha}n$.

(1) Generate two disjoint subsets I and J of relations from $\mathcal{R}$ in parallel such that the OBDDs $Q(I)$ and $Q(J)$ are of moderate size and the set $Q(I \cup J)^{-1}(1)$ of satisfying assignments of $Q(I) \wedge Q(J) = Q(I \cup J)$ is small. Then compute $Q(I \cup J) = Q(I) \wedge Q(J)$ using the standard OBDD-synthesis algorithm.

(2) Suppose that strategy (1) leads to sets I and J for which the set $Q(I \cup J)^{-1}(1)$ is not just "small" but even comprises of only a *single* element. Then compute this satisfying assignment directly from $Q(I)$ and $Q(J)$ by means of our new breadth-first-search-based algorithm instead of performing the OBDD synthesis $Q(I) \wedge Q(J)$.

The first part of our experimental results, presented in Section 4, shows that strategy (1) is a lot more space efficient than computing $Q(I \cup J)$ using the classical OBDD-attack suggested in [6] (and employed in all follow-up works since). Strategy (2) allows us to investigate the complexity of the following *Bounded Synthesis Problem*: given two OBDDs P_1 and P_2 for which it is known that $P_1 \wedge P_2$ has only one satisfying assignment, compute this satisfying assignment.

The standard solution is to compute $P_1 \wedge P_2$ using the standard OBDD-synthesis algorithm and then to minimize the resulting OBDD. Note that for solving the bounded synthesis problem, it is sufficient to find the only existing directed path connecting the root to the 1-sink of $P_1 \wedge P_2$, as this path corresponds to the only satisfying assignment of $P_1 \wedge P_2$.

In Section 5, we describe a depth-first-search (DFS) approach and a breadth-first-search (BFS) approach for solving the Bounded Synthesis Problem. Our most promising candidate is the BFS-approach as it allows to identify nodes in $P_1 \wedge P_2$ which do not occur in the minimized OBDD.

Structure of the paper: The remaining part of this paper is organized as follows. In **Section 2**, we first provide the basic information about OBDDs. We do this by considering a subfamily of OBDDs, so-called leveled OBDDs (LOBDDs). The reason for considering LOBDDs is that the relevant algorithmic properties of OBDDs can be explained more easily for LOBDDs, and in our cryptographic context we obtain LOBDDs in a natural way. Note that all of our results about LOBDDs hold also for OBDDs. In **Section 3**, we describe the classic BDD attack against stream ciphers on the basis of a toy example. Note that like this toy example, also the generators used in our experiments do not provide sufficient cryptographic hardness for modern practical applications. However, due to their feasible inner state size and simple definition, they are well suited for applying and verifying our algorithmic ideas. In **Section 4**, we describe strategy (1) and the intuition behind it in further detail and provide corresponding experimental results. In **Section 5**, we present our algorithmic approaches for the bounded synthesis problem. Finally, in **Section 6**, we present the experimental results which compare our DFS- and BFS-approach to the standard synthesis algorithm, and conclude the paper.

2 Preliminaries

2.1 Leveled Ordered Binary Decision Diagrams (LOBDDs)

Definition 1. *An LOBDD P over $X_n = \{x_1, \ldots, x_n\}$ is a directed acyclic labeled Graph $G = (V, E)$ with one root and one sink, where V is the set of vertices and E is the set of edges. The following properties apply to P:*

- *The set of vertices V is partitioned into $n + 1$ pairwise disjoint levels, $L_1(P), \ldots, L_{n+1}(P)$. The nodes in $L_i(P)$, $1 \leqslant i \leqslant n$, are labeled x_i.*
- *$L_1(P)$ contains only the root $\texttt{root}(P)$, labeled x_1.*
- *$L_{n+1}(P)$ contains only the sink $\texttt{sink}(P)$, labeled 1.*
- *The edges are labeled with either 0 or 1. The labels correspond to the Boolean value of the respective variable.*
- *For all $v \in V \backslash \{\texttt{sink}(P)\}$, there are either two outgoing edges with different labels or there is only one outgoing edge (labeled 0 or 1).*

– *For each edge* $(u, v) \in E$*, there exists some level* $i \in \{1, \ldots, n\}$ *such that* $u \in L_i(P)$ *and* $v \in L_{i+1}(P)$*.*

– *The size of* P *is denoted by* $|P|$ *and it is defined to be equal to the amount of nodes, i.e.,* $|P| := |V|$*.*

– *The width of* P *is denoted by* $\mathtt{width}(P)$ *and it is defined to be equal to the size of the largest level, i.e.,* $\mathtt{width}(P) := \max_{1 \leqslant i \leqslant n} \{|L_i(P)|\}$*.*

Fix some $i \in \{1, \ldots, n+1\}$ and fix some node $v \in L_i(P)$. A path leading from the root $\mathtt{root}\,(P)$ to v corresponds to a unique $\{0,1\}$-assignment of the variables $\{x_1, \ldots, x_{i-1}\}$. Similarly, a path leading from v to the sink $\mathtt{sink}\,(P)$ corresponds to a unique $\{0,1\}$-assignment of the variables $\{x_i, \ldots, x_n\}$. This motivates the following definitions:

Definition 2. *For all* $i \in \{1, \ldots, n+1\}$ *and all nodes* $v \in L_i(P)$*, we define* $\mathtt{Reach}_P(v)$ *to be the set of all those* $\{0,1\}$*-assignments of* $\{x_1, \ldots, x_{i-1}\}$ *which correspond to paths from* $\mathtt{root}\,(P)$ *to* v*.*

Definition 3. *For all* $i \in \{1, \ldots, n+1\}$ *and all nodes* $v \in L_i(P)$*, we define* $\mathtt{Sat}_P(v)$ *to be the set of all those* $\{0,1\}$*-assignments of* $\{x_i, \ldots, x_n\}$ *which correspond to paths from* v *to* $\mathtt{sink}(P)$*.*

Definition 4. *The Boolean function* $f : \{0,1\}^n \longrightarrow \{0,1\}$ *computed by* P *is defined by*

$$f(x) = 1 \Longleftrightarrow x \in \mathtt{Sat}_P(\mathtt{root}\,(P)) = \mathtt{Reach}_P(\mathtt{sink}(P))\,.$$

It can be straightforwardly shown that for each Boolean function $f : \{0,1\}^n \to \{0,1\}$ there exists an LOBDD P over X_n that computes f. Further, there exists an efficient algorithm running in space and time $\mathcal{O}(|P|)$ that minimizes P, i.e., it computes the minimal LOBDD that represents the same function as P. Note that P is minimal if and only if each node of P is reachable from $\mathtt{root}\,(P)$ and if for all $i \in \{2, \ldots, n\}$ and for all $v \neq v' \in L_i(P)$ we have $\mathtt{Sat}_P(v) \neq \mathtt{Sat}_P(v')$. Otherwise v and v' could be merged into one node. We refer the reader to [12] for further details.

For any two LOBDDs P and Q over X_n, we can define the canonical LOBDD $P \wedge Q$ over X_n computing the logical conjunction of the functions computed by P and Q. For defining $P \wedge Q$, we again consider a graph $G = (V, E)$. The set

of vertices is partitioned into $n+1$ levels $L_1(G), \ldots, L_{n+1}(G)$ and the nodes in $L_i(G)$, $1 \leqslant i \leqslant n$, are labeled by x_i.

Definition 5. *Let P and Q be two arbitrary LOBDDs over X_n. The LOBDD $G := P \wedge Q$ is defined as follows. For all $i \in \{1, \ldots, n+1\}$:*

- *Level i is defined to be $L_i(G) := L_i(P) \times L_i(Q)$.*
- *There exists an edge $((u,u'),(v,v')) \in L_i(G) \times L_{i+1}(G)$ labeled $b \in \{0,1\}$ if and only if (u,v) is an edge labeled b in P and (u',v') is an edge labeled b in Q.*

It follows directly that $L_1(G) = \{(\mathtt{root}\,(P), \mathtt{root}\,(Q))\}$ and $L_{n+1} = \{(\mathtt{sink}\,(P), \mathtt{sink}\,(Q))\}$. Further, the LOBDD $P \wedge Q$ is formed by all nodes of G reachable from the root $(\mathtt{root}\,(P), \mathtt{root}\,(Q))$. It can be easily checked that $P \wedge Q$ is an LOBDD computing the logical conjunction of the functions computed by P and Q. In general, $P \wedge Q$ is not minimal, even if P and Q are minimal. In the worst case, $P \wedge Q$ has the width $\Theta(\mathtt{width}\,(P) \cdot \mathtt{width}\,(Q))$.

The above definition can be straightforwardly generalized to compute the LOBDD $P_1 \wedge \cdots \wedge P_k$ for given LOBDDs $P_1, \ldots, P_k$. The LOBDD $P_1 \wedge \cdots \wedge P_k$ computes the logical conjunction of the functions computed by $P_1, \ldots, P_k$. In the worst case, it has the width $\Theta(\mathtt{width}\,(P_1) \cdot \cdots \cdot \mathtt{width}\,(P_k))$.

2.2 Comparison of LOBDDs and OBDDs

In Subsection 2.1, we defined LOBBDs, which respect the canonical variable ordering $(x_1, x_2, \ldots, x_n)$ on all paths from the root to the sink. Similarly, π-LOBDDs can be defined, which respect the variable ordering $(x_{\pi(1)}, x_{\pi(2)}, \ldots, x_{\pi(n)})$ on all paths from the root to the sink, where π is some permutation on $\{1, \ldots, n\}$. LOBDDs differ from general OBDDs in two ways:

1. OBDDs are usually defined to have a 1-sink and a 0-sink. Moreover, each node has exacly two edges, labeled 0 and 1, respectively. A missing edge in our LOBDD definition is equivalent to an edge pointing to the 0-sink. In fact, this is just a matter of notation.

2. General OBDDs do not have to be leveled, i.e., there may be an edge (u,v) from a node u labeled x_i to a node v labeled x_j, where $1 \leqslant i < j \leqslant n+1$. If $j > i+1$, this implies that the subfunction computed at v does not

depend on the variables $x_{i+1}, \ldots, x_{j-1}$. In LOBDDs, on contrast, we always demand $j = i + 1$.

Each OBDD P can be easily converted to an LOBDD P' with $|P'| \leqslant n \cdot |P|$: Consider the edge (u, v), where u is a node labeled x_i, v is a node labeled x_j, and $j > i + 1$. In the LOBDD, (u, v) is replaced by a sequence of $j - i - 1$ dummy nodes labeled $x_{i+1}, \ldots, x_{j-1}$.

Note that if an OBDD P is satisfying the property that for all $x \neq x' \in \mathtt{Sat}_P(\mathtt{root}(P))$ the Hamming distance between x and x' is larger than one, then P has to be an LOBDD. As all functions considered in our cryptographic context have this property, we decided to work with LOBDDs in most parts of the paper (esp. in Section 4) since minimization and the construction of $P \wedge Q$ can be defined more easily in the model of LOBDDs.

3 Classical BDD Attacks against Stream Ciphers

Let us consider the toy example of a simple KSG with inner state size five, whose secret inner bitstream $x_1, x_2, \ldots$ is defined by the feedback relation

$$x_{t+5} := x_t \oplus x_{t+2} \quad \text{for } t \geqslant 1, \tag{1}$$

where $(x_1, \ldots, x_5)$ denotes the secret initial state. Moreover, let the output function of this KSG be given as

$$z_t := x_{t+2} \cdot x_{t+4} \quad \text{for } t \geqslant 1. \tag{2}$$

It can be easily checked that for this KSG, the initial state $(x_1, \ldots, x_5) = (0, 1, 1, 0, 1)$ leads to the keystream prefix $(z_1, \ldots, z_7) = (1, 0, 1, 0, 1, 0, 1)$.

Classical OBDD-based cryptanalysis now proceeds as follows. An attacker who gets hold of the above keystream prefix starts by turning his information about step $t = 1$ into an OBDD. From Eq. (1), he knows that

$$x_1 \oplus x_3 \oplus x_6 = 0 \tag{3}$$

holds w.r.t. the newly generated inner state stream bit x_6. This knowledge is represented by the OBDD R_1 depicted in Fig. 1, whose satisfying assignments (leading to the OBDD's 1-sink) are those satisfying Eq. (3).

Also at $t = 1$, the attacker learns from the observed first keystream bit

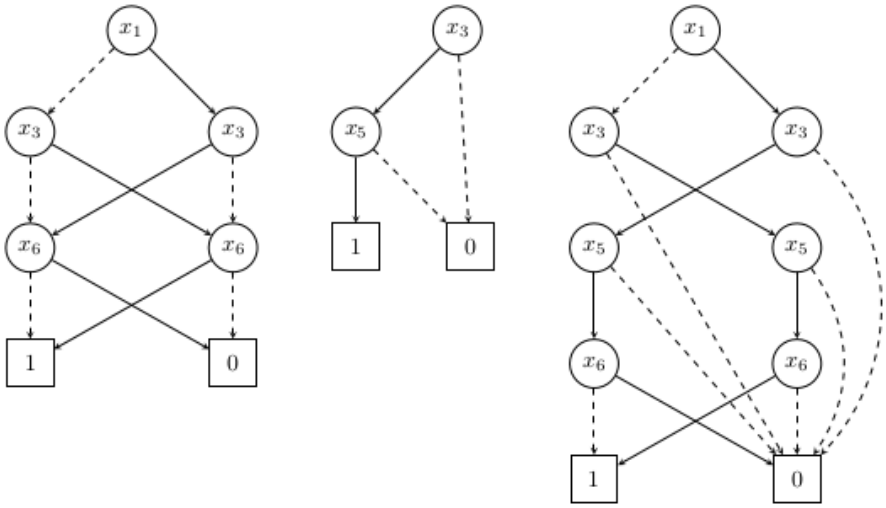


Figure 1: OBDDs R_1 (left), Q_1 (mid.), $P_1 = R_1 \wedge Q_1$ (right). The solid (resp. dashed) edges denote that the variable labeling the edge's source node takes the value 1 (resp. 0).

$z_1 = 1$ together with Eq. (2) that

$$x_3 \cdot x_5 = 1 \tag{4}$$

must hold. This knowledge is represented by the OBDD Q_1 in Fig. 1.

Through AND-synthesis of R_1 and Q_1, the attacker finally obtains the OBDD P_1 depicted in Fig. 1, whose satisfying assignments are exactly those assignments to x_1, x_3, x_5, x_6 which simultaneously fulfill Eqs. (3) and (4).

For the next step, $t = 2$, the attacker proceeds analogously. More precisely, he builds the OBDDs R_2 and Q_2 corresponding to the relations $x_2 \oplus x_4 \oplus x_7 = 0$ and $x_4 \cdot x_6 = 0$ (as $z_2 = 0$), respectively. The new main OBDD P_2 is computed as $P_2 := P_1 \wedge R_2 \wedge Q_2$.

The general attack strategy is now as follows. The attacker will treat the subsequent iterations $t = 3, 4, \ldots$ accordingly, obtaining further growing OBDDs P_3, P_4, and so on. However, as explained in detail in [6], at some point, the size of the OBDDs P_t will eventually reach a maximum and henceforth (usually quickly) decrease. Note that this maximum actually dominates the overall complexity of the attack.

In the case of our toy example, after only seven steps, the main OBDD P_7 has degraded into a list as depicted in Fig. 2. The only satisfying assignment to the first twelve bits of the inner state stream can be derived from P_7 directly as $(x_1, \ldots, x_{12}) = (0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0)$. The first five bits $(x_1, \ldots, x_5)$ are

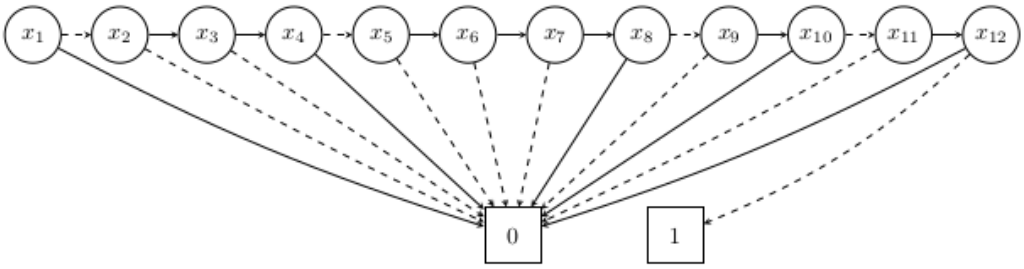


Figure 2: The OBDD P_7, which contains the solution of our cryptanalysis.

the secret initial state that underlies the attacked keystream prefix and, hence, represent the solution of our OBDD-based cryptanalysis. From this initial state, an attacker can now generate the full keystream.

4 Attack Improvement and Experimental Results

Instead of building only one main OBDD P_t (see our example in Section 3) we now suggest to work with 'two main OBDDs' P_t^1 and P_t^2 as follows.

Algorithm 2 A new approach for more efficient, parallelizable OBDD attacks.

$X \leftarrow$ *merging point parameter* (see explanation below)
$(P_0^1, P_0^2) \leftarrow (1\text{-OBDD}, 1\text{-OBDD})$
for $t = 1$ to X **do**
  if t is odd **then**
    $(P_t^1, P_t^2) \leftarrow (P_{t-1}^1 \wedge R_t \wedge Q_t, P_{t-1}^2)$
  else
    $(P_t^1, P_t^2) \leftarrow (P_{t-1}^1, P_{t-1}^2 \wedge R_t \wedge Q_t)$
  end if
end for
$\tilde{P}_X \leftarrow P_X^1 \wedge P_X^2$
$t \leftarrow X$
while $\tilde{P}_t$ has more than one satisfying assignments **do**
  $t \leftarrow t + 1$
  $\tilde{P}_t \leftarrow \tilde{P}_{t-1} \wedge R_t \wedge Q_t$
end while
  return $\tilde{P}_t$

The intuition behind the efficiency of our new approach is the following. Let $t^{\max}$ denote the step in which the single main OBDD P_t would have reached its maximum size $|P_{t^{\max}}|$ and remember that for $t > t^{\max}$, the sizes $|P_t|$ would henceforth constantly decrease until only one satisfying assignment is left. Then,

for a properly chosen parameter X (in particular, $X > t^{\max}$), the result $\tilde{P}_X := P_X^1 \wedge P_X^2$ in Algorithm 2 will already have much less nodes than $P_{t^{\max}}$ with the effect that the respective synthesis operation will have advantageous (as compared to working with $P_{t^{\max}}$) expected time and memory consumption. Note that for the new approach to be actually more efficient than the classical one, X must be still small enough such that the sizes of the OBDDs P_X^1 and P_X^2 still stay significantly below $|P_{t^{\max}}|$.

In order to assess the efficiency gains achievable with our new approach, we performed an experimental evaluation using the well-known BDD package CUDD [10]. As the attack target, we considered a simple KSG of size 39 bits, whose inner state stream is defined by the relation

$$x_{t+39} := x_t \oplus x_{t+13} \oplus x_{t+5} \cdot x_{t+17} \oplus x_{t+24} \cdot x_{t+29} \quad \text{for } t \geqslant 1,$$

and whose output function is $z_t := x_{t+9} \oplus x_{t+19} \oplus x_{t+29}$, $t \geqslant 1$.

The classical approach described in Section 3 leads to a memory consumption of almost 900 MB and takes about 215 seconds. The single main OBDD used there reaches its maximum size at $t^{\max} = 17$. In contrast, the same attack using our new approach in Algorithm 2 (with $X = 22$) requires only about 110 MB of memory and is completed in as few as 8 seconds.

Also note that the OBDDs P_t^1 and P_t^2 are built on the basis of completely separate information. Thus, they could as well be easily computed in parallel on different CPUs in order to further speed up the attack.

5 A Related Special Synthesis Problem for LOBDDs and Algorithmic Approaches

We have seen in the previous sections that given a piece of keystream z, the problem of computing the secret inner state x which generates z can be reduced to computing a sequence $P_1, \ldots, P_k$ of (L)OBDDs of a moderate size such that x is a satisfying assignment of $P_1 \wedge \cdots \wedge P_k$ and the number of satisfying assignments of $P_1 \wedge \cdots \wedge P_k$ is moderately bounded. In Section 4, we described a corresponding approach with $k = 2$ and verified through experiments that this approach was indeed significantly more efficient than the classical BDD-attack described in Section 3.

The standard way of computing the set of satisfying assignments of $P_1 \wedge \cdots \wedge$

P_k from $P_1, \ldots, P_k$ is to construct $P_1 \wedge \cdots \wedge P_k$ as described in Subsection 2.1 and to minimize it. This algorithm needs space and time $\Theta(|P_1 \wedge \cdots \wedge P_k| + \sum_{i=1}^{k} |P_i|)$, where in the worst case we have $|P_1 \wedge \cdots \wedge P_k| = \Theta(\Pi_{i=1}^{k} |P_i|)$. The resulting question is the following:

> *Can we find more efficient algorithms to compute the set S of satisfying assignments of $P_1 \wedge \cdots \wedge P_k$ if $|S|$ is small?*

We are convinced that this question represents a fundamental open problem in complexity theory. So far, we cannot yet present an algorithm for this problem which has an asymptotic worst case running time essentially better than $\Theta(\Pi_{i=1}^{k} |P_i|)$. Nonetheless, we are able to come up with new algorithmic approaches which are much simpler than the standard synthesis algorithm and which explicitly use the fact that the set of satisfying assignments of $P_1 \wedge \cdots \wedge P_k$ is small. Our experimental results in Section 6 show that our algorithms are more space efficient than the standard synthesis.

In the following, we restrict ourselves to the case that $k = 2$ and $|S| = 1$; i.e., we are given two LOBDDs P_1 and P_2 over the set of Boolean variables $X_n = \{x_1, \ldots, x_n\}$ for which we know $|(P_1 \wedge P_2)^{-1}(1)| = 1$. The problem is to compute the only satisfying assignment x of $P_1 \wedge P_2$ in a more efficient way than by the standard synthesis algorithm described above. One obvious consequence of the assumption $|(P_1 \wedge P_2)^{-1}(1)| = 1$ is that we do not have to compute the complete LOBDD $P_1 \wedge P_2$. It is sufficient to compute the only path in $P_1 \wedge P_2$ leading from the root (v_1^0, v_2^0) to the sink (s_1, s_2).

There are two elementary graph algorithms that compute all nodes of a given directed graph which are reachable from a given starting point: depth-first search (DFS) and breadth-first search (BFS). Hence, we present two algorithmic approaches for computing $(P_1 \wedge P_2)^{-1}(1)$: a *DFS approach* and a *BFS approach*.

5.1 The DFS Approach

The underlying data structure is a stack S used to store the nodes of $P_1 \wedge P_2$. The stack maintains a pointer $\texttt{head}(S)$ to the node on top of the stack. During the execution of the algorithm, some nodes of $P_1 \wedge P_2$ will be labeled *gray* (discovered) and some will be labeled *black* (discovered but useless). The algorithm starts by labeling the root (v_1^0, v_2^0) gray and pushing it on top of the stack. In each iteration, the node $(v, v') = \texttt{head}(S)$ on top of the stack is considered. If there is a successor (w, w') of (v, v') which is not black, then

(w, w') will be labeled gray and pushed onto S, i.e., $\texttt{head}\,(S) = (w, w')$. If not, (v, v') will be labeled black and removed from the stack. The algorithm stops if it discovers the sink (s_1, s_2). In this case, the nodes in the stack identify the only assignment in $(P_1 \wedge P_2)^{-1}(1)$.

It is a well known fact that this algorithm has time and space costs of $\mathcal{O}(|P_1 \wedge P_2|)$. While the stack always contains at most n nodes, the space consuming part of this approach results from the necessity to store all nodes labeled black. In the worst case, a large part of the nodes of unminimized $P_1 \wedge P_2$ have to be discovered before finding the path to (s_1, s_2).

5.2 The BFS Approach

This approach uses the following easy fact resulting from $|(P_1 \wedge P_2)^{-1}(1)| = 1$.

Lemma 1. *For all nodes (v, v') in $P_1 \wedge P_2$, the property $|\mathit{Sat}_{P_1 \wedge P_2}(v, v')| \leqslant 1$ is fulfilled, and if $|\mathit{Reach}_{P_1 \wedge P_2}(v, v')| > 1$, then $\mathit{Sat}_{P_1 \wedge P_2}(v, v') = \varnothing$.*

The underlying data structure for BFS is a queue Q used to store the nodes of $P_1 \wedge P_2$. Q is equipped with the pointers $\texttt{tail}\,(Q)$, pointing to the back of Q, and the pointer $\texttt{head}\,(Q)$, pointing to the front of Q. The nodes (v, v') of the LOBDD $P_1 \wedge P_2$ have an additional property $(v, v').\texttt{path}$. $(v, v').\texttt{path}$ contains the assignment of the Boolean variables corresponding to the only path in $P_1 \wedge P_2$ from the root to (v, v').

During the execution of the algorithm, some nodes of $P_1 \wedge P_2$ will be labeled *gray* (discovered) and some nodes will be labeled *black* (discovered but useless). The algorithm starts by labeling the root (v_1^0, v_2^0) gray and putting it into the queue Q, i.e., $\texttt{tail}\,(Q) = \texttt{head}\,(Q) = (v_1^0, v_2^0)$.

In each iteration, the node $(v, v') = \texttt{head}\,(Q)$ is removed from Q. If (v, v') is gray, each successor (w, w') of Q is processed in the following way: If (w, w') was not discovered before, then $(w, w').\texttt{path}$ is computed by adding the label of the edge $((v, v'), (w, w'))$ to $(v, v').\texttt{path}$. (w, w') will be labeled gray and put into the queue Q, i.e., $\texttt{tail}\,(Q) = (w, w')$. If (w, w') is gray (already discovered and contained in Q), it is labeled black. The algorithm stops when (s_1, s_2) is labeled gray. In this case, $(s_1, s_2).\texttt{path}$ yields the only satisfying assignment of $P_1 \wedge P_2$.

Note that if a node (v, v') is black, then $|\texttt{Reach}_{P_1 \wedge P_2}(v, v')| > 1$. This implies $\texttt{Sat}_{P_1 \wedge P_2}(u, u') = \varnothing$ for all nodes (u, u') reachable from (v, v') in $P_1 \wedge P_2$.

Consequently, if a black node is dequeued from Q, its successors will not be enqueued into Q. Compared to the DFS approach, it is not necessary to store nodes removed from Q. Hence, the space consumption equals the maximum size of Q. It can be upper bounded by $\texttt{width}\,(P_1) \cdot \texttt{width}\,(P_2)$.

6 Further Experimental Results and Conclusion

In order to assess the space reduction achievable with our new algorithmic approaches presented in Section 5, we performed corresponding experiments based on size-reduced KSG prototypes. But unlike in Section 4, we could not use an existing BDD package such as CUDD, which is highly optimized for real-world usage and would not have easily allowed for the kind of algorithmic analysis (w.r.t. implementation-independent metrics) required in our context. Consequently, we had to write our own LOBDD package from scratch in order to derive experimental results for

- a 28-bit Geffe-generator with three maximum-period LFSRs defined by $l_{t+7} := l_t \oplus l_{t+1}$, $m_{t+10} := m_t \oplus m_{t+3}$, and $n_{t+11} := n_t \oplus n_{t+2}$, and the output function $z_t := l_t \cdot m_t \oplus m_t \cdot n_t \oplus n_t$;
- a 26-bit NFSR with feedback function $x_{t+26} := x_t \oplus x_{t+13} \oplus x_{t+5} \cdot x_{t+17}$ and output function $z_t := x_{t+9} \oplus x_{t+19}$.

As described in Section 5, the relevant memory consumption metrics for our new algorithms are the maximum size (i.e., number of elements) of the set of black nodes for the DFS approach and of the queue for the BFS approach. We compare this to the approach that the LOBDD $P_1 \wedge P_2$ is actually computed in order to find the only assignment in $(P_1 \wedge P_2)^{-1}(1)$. The relevant metric there is the size (i.e., number of nodes) of the LOBDD *before minimization*. The results in the following table are an average based on 100 randomly sampled (*Initial State*, *Keystream Prefix*)-pairs per KSG:

	DFS-based (black nodes)	BFS-based (queue nodes)	Synthesis-based (LOBDD nodes)
Geffe-KSG	4003455	667694	6530494
NFSR-KSG	7631335	557169	10977359

Clearly, both of our new algorithmic approaches are more space efficient than using standard synthesis to compute $(P_1 \wedge P_2)^{-1}(1)$. Especially the BFS-based

variant seems extremely promising and could initiate a new phase of BDD-based cryptanalysis. As future work, we suggest to integrate our new approaches into standard BDD packages like CUDD.

References

[1] Babbage S.H., “Improved "exhaustive search" attacks on stream ciphers”, European Convention on Security and Detection, 1995, 161–166.

[2] *Bluetooth SIG*, Bluetooth Core Specification 4.2, 2014, https://www.bluetooth.org/DocMan/handlers/Downloa

[3] Briceno M., Goldberg I., and Wagner D., “A pedagogical implementation of A5/1”, 1999, http://www.scard.org/gsm/a51.html.

[4] Eibach T., Pilz E., and Völkel G., “Attacking Bivium Using SAT Solvers”, SAT 2008, 2008, 63–76.

[5] Golić J. Dj., “On the security of nonlinear filter generators”, FSE 1996, *Proceedings*, 1996, 173–188.

[6] Krause M., “BDD-Based Cryptanalysis of Keystream Generators”, EUROCRYPT 2002, *Proceedings*, 2002, 222–237.

[7] Krause M. and Stegemann D., “Reducing the Space Complexity of BDD-Based Attacks on Keystream Generators”, FSE 2006, 2006, 163–178.

[8] Meier W. and Staffelbach O., “The Self-Shrinking Generator”, *LNCS*, EUROCRYPT ’94, **950**, Springer, 1994, 205–214.

[9] Meinel C. and Theobald T., Algorithms and Data Structures in VLSI Design, 1998.

[10] Somenzi F., “CUDD 3.0.0”, 2015., http://vlsi.colorado.edu/ fabio/, Website (accessed on February 05, 2018).

[11] Stegemann D., “Extended BDD-Based Cryptanalysis of Keystream Generators”, SAC 2007, *Proceedings*, 2007, 17–35.

[12] Wegener I., *Branching Programs and Binary Decision Diagrams: Theory and Applications*, SIAM e-books. Society for Industrial and Applied Mathematics, 2000.

Applications

On Security of TLS 1.2 Record Layer with Russian Ciphersuites

Liliya Akhmetzyanova, Evgeny Alekseev, Grigory Sedov, and Stanislav Smyshlyaev

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

Abstract

The TLS protocol is the most widely used cryptographic protocol providing secure communications over the Internet. In 2019 Rosstandart has approved recommendations for standardization those define the use of the TLS 1.2 ciphersuites based on the actual Russian cryptographic algorithms. This paper presents security bounds for the Record protocol defined by the Russian ciphersuites that provides authenticity and confidentiality of transmitted data. The bounds were obtained in the IND-sfCCSA model, which relevance we pay special attention to, and are presented as a function of the used cryptographic primitives security bounds. When using these bounds to obtain the certain security parameters of the Record protocol, one must take into account the current state of research concerning security of the used primitives.

Keywords: TLS protocol, information security, Russian cryptographic algorithms.

1 Introduction

One of the main applications of cryptography is the establishment of a secure connection, namely provision of an authenticated channel between a client and a server, ensuring integrity and confidentiality of transmitted data. The most widely used protocol solving this task is the Transport Layer Security (TLS) protocol.

The TLS protocol consists of two layers. The Record protocol represents the low layer and works over some transport protocol (for example TCP) providing reliable connection with guaranteed delivery of data packets. The Record protocol provides confidentiality and integrity of transmitted data and uses keys and cryptographic parameters those are negotiated during the Handshake protocol running. The specific modes of operation of the TLS protocol, which determine

all its sub-protocols, in particular, the Record and Handshake protocols, are specified within the framework of the so-called «ciphersuites».

Due to its use in the vast majority of modern web applications, TLS is the most researched for the presence of both theoretical [13, 37, 38] and practical [24, 23, 35, 31] vulnerabilities. Therefore the protocol has changed dramatically from the first versions in purposes of security. In 2018 the TLS protocol version 1.3 was adopted as the current standard [34], which was designed with all modern cryptographic principles taken into account. However, the introduction and distribution of a new solution take a lot of time, so the protocol version 1.2 still remains the current standard. In particular, now it is supported by 95% of sites in the Internet [1]. In addition, TLS 1.3 is supported lower than 15% of sites. So we can claim that more than 80% sites in the Internet supports only TLS 1.2.

In 2019 Rosstandart has approved recommendations for standardization [8] which define the use of TLS 1.2 ciphersuites based on the actual Russian cryptographic algortihms [3, 2, 7, 5, 6]. In this document the following ciphersuites are defined:

- TLS_GOSTR341112_256_WITH_MAGMA_CTR_OMAC (0xC1,0x01);
- TLS_GOSTR341112_256_WITH_KUZNYECHIK_CTR_OMAC (0xC1,0x00).

In 2019 IANA has added these ciphersuites to the "TLS Cipher Suite" registry with the numbers listed above in the brackets.

The main principles of the Handshake protocol almost was not changed compared to the previous version of the Russian ciphersuites [4]. At the same time, the Record protocol is not similar to its analogue from the previous version of the Russian ciphersuites, nor to the foreign versions. It uses all the advanced developments related to the tasks of providing a secure channel and increasing a key lifetime (that is, the amount of data processed under a single key). In this paper, we focus on the security analysis of the Record protocol only, assuming that the key material produced during the Handshake protocol running and used by the Record protocol, is chosen at random according to the uniform distribution on the set of fixed length binary strings.

The Record protocol is based on an authenticated encryption scheme with associated data that ensures confidentiality and integrity of transmitted data.

The analyzed ciphersuites use the Mac-then-Encrypt [15] scheme that consists of the CTR-ACPKM [7] encryption mode and the OMAC message authentication code [2]. The MAC-then-Encrypt scheme without the possibility of associated data processing was analyzed in [28] where its security was proven when using the stream cipher mode or the CBC mode (only for the data lengths that are multiple of the blockcipher block length). The security analysis of the Mac-then-Encrypt scheme with associated data using the CBC mode with a random initialization vector was carried out in [32], which also took into account the features of using the message padding procedure. The analysis was made in the model, that ensures integrity only within the separate messages. To achieve integrity at the message flow level, it is necessary to consider so-called *stateful schemes*. In [22] the sfAE model was introduced for such schemes analysis. Also the method for constructing secure stateful schemes based on secure stateless schemes was proposed.

Unfortunately, the results mentioned above cannot be directly applied to obtain the security bounds for the Record protocol specified by the Russian ciphersuites. Indeed, the work of [32] does not provide the security analysis of the stateful Mac-then-Encrypt scheme with associated data that uses arbitrary basic encryption and MAC modes in the way defined by the TLS version 1.2 protocol standard. The direct application of the results of [22] for the analysis of the Russian ciphersuites is also impracticable, since this paper consider the scheme, where the encryption mode uses random and independent of the internal state values of IV. At the same time, for the Record protocol defined by the Russian ciphersuites, the nonce-based CTR-ACPKM encryption mode is used where IVs depend on the current state.

In this paper we analyze a general stateful MAC-then-Encrypt AEAD scheme that uses nonce-based encryption mode and MAC as it is specified in TLS 1.2 RFC (see [25], the case of GenericStreamCipher). The analysis was carried out in IND-sfCCSA model that extends the sfAE model. We focus on the relevance of the proposed model for analyzing the Record protocol. The lower bound of the proposed scheme security was obtained as a function of the used encryption mode and MAC security level in the standard ROR-CPNA [36] and PRF [16] models, respectively. As far as the authors know, such a bound has not been previously presented explicitly in the literature. Also this work presents bounds for a general stateful AEAD scheme with a pseudorandom generator used for re-keying purpose. The obtained bound are applied to the

Record protocol defined by the Russian ciphersuites. The resulting bounds depend on the block cipher (Magma or Kuznyechik) security in the PRP-CPA model and the HMAC function (based on the Stribog hash function) security in the PRF model.

The paper is organized as follows. In Section 2 and Section 3 basic definitions and notation are introduced, basic schemes and security models are defined. Section 4 introduces the IND-sfCCSA security model, explains its design features. Section 5 is devoted to the security analysis of the general stateful Mac-then-Encrypt scheme with associated data and of the general AEAD scheme with pseudorandom generator. Section 6 describes the main object of the research — the Record protocol defined by the Russian ciphersuites. In Section 7, the bound obtained in the previous sections is applied to the analyzed protocol.

2 Basic notations and definitions

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. For $a \in \{0,1\}$ let a^r be the string, consisting of r symbols a. For bit strings a and b we denote by $a\|b$ their concatenation. Let $|a|$ be the bit length of the string a.

For a bit string u and a positive integer $l \leqslant |u|$ let $\mathsf{msb}_l(u)$ ($\mathsf{lsb}_l(u)$) be the string, consisting of the leftmost (rightmost) l bits of u. For integers $l > 0$ and $i \geqslant 0$ let $\mathsf{str}_l(i)$ be l-bit representation of i with the least significant bit on the right. For an integer $l > 0$ and a bit string $u \in \{0,1\}^l$ let $\mathsf{int}(u)$ be the integer i such that $\mathsf{str}_l(i) = u$.

For any set S, define $Perm(S)$ as the set of all bijective mappings on S (permutations on S). A *block cipher* E (or just a *cipher*) with block size n and key size k is a permutation family $\left(E_K \in Perm(\{0,1\}^n) \mid K \in \{0,1\}^k\right)$, where K is a key. If the value s is chosen from a set S uniformly at random, then we denote $s \xleftarrow{U} S$.

If the variable x gets the value val then we denote $x \leftarrow val$. Similarly, if the variable x gets the value of the variable y then we denote $x \leftarrow y$. If the variable x gets the result of a probabilistic algorithm A we denote $A \xrightarrow{\$} x$ ($x \xleftarrow{\$} A$). If we need to emphasize that A is deterministic than we denote it by $A \rightarrow x$ ($x \leftarrow A$). The event when A returned value val as a result is denoted by $A \rightarrow val$

We model an adversary using an interactive probabilistic algorithm that has access to one or more oracles. The resources of an adversary A are measured

in terms of time and query complexities. For a fixed model of computation and a method of encoding the time complexity includes the description size of A. The query complexity usually includes the number of queries and the maximal length of queries or the total length of queries. Denote by $\mathsf{Adv}^{\mathrm{M}}_{\mathrm{S}}(A)$ the measure of the success of the adversary A in realizing a certain threat, defined by the model M, for the cryptographic scheme S. The formal definition of this measure will be given in each specific case.

3 Basic algorithms and security models

Standard security model for block ciphers is PRP-CPA («Pseudo Random Permutation under Chosen Plaintext Attack») (see, e.g. [17]). The formal description is presented in Appendix A.1.

Introduce the definition of a symmetric encryption scheme SE. In the current paper we consider encryption schemes those use an additional initialization vector. Values of the initialization vector may be restricted by some conditions.

Definition 1. *Let $\mathcal{K} \subseteq \{0,1\}^*$ be a set of keys, $\mathcal{M} \subseteq \{0,1\}^*$ be a set of plaintexts, $\mathcal{C} \subseteq \{0,1\}^*$ be a set of ciphertexts, and $\mathtt{IV} \subseteq \{0,1\}^*$ be a set of initialization vectors. An* IV-based symmetric encryption scheme *is a set of algorithms $\mathsf{SE} = \{\mathsf{SE.K},\ \mathsf{SE.E},\ \mathsf{SE.D}\}$, where*

- $\mathsf{SE.K} \xrightarrow{\$} K$*: A probabilistic algorithm outputting a key $K \in \mathcal{K}$.*
- $\mathsf{SE.E}(K, IV, m) \to c$*: A deterministic encryption algorithm taking an initialization vector $IV \in \mathtt{IV}$, a key $K \in \mathcal{K}$, and a plaintext $m \in \mathcal{M}$ as its inputs. An output of the algorithm is a ciphertext $c \in \mathcal{C}$.*
- $\mathsf{SE.D}(K, IV, c) \to m$*: A deterministic decryption algorithm taking an initialization vector $IV \in \mathtt{IV}$, a key $K \in \mathcal{K}$, and a ciphertext $c \in \mathcal{C}$ as its inputs. An output of the algorithm is a plaintext $m \in \mathcal{M}$.*

The standard notion for encryption modes analysis is the ROR-CPNA («Real or Random under Chosen Plaintext and Nonce Attack») model. The formal description is presented in Appendix A.1. This model is similar to the standard ROR-CPA security model [18] but considers nonce-respecting adversaries [36]. Informally, in this model the adversary has to distinguish the obtained ciphertexts from the ciphertext of «garbage», having the capability to adaptively choose plaintexts and nonces (in a unique manner).

Introduce the definitions of a message authentication scheme MA and an security model PRF, which is used in analysis of message authentication schemes.

Definition 2. *Let* $\mathcal{K} \subseteq \{0,1\}^*$ *be a set of keys,* $\mathcal{M} \subseteq \{0,1\}^*$ *be a set of messages,* $\mathcal{T} \subseteq \{0,1\}^*$ *be a set of tags. A* deterministic message authentication scheme *ia a set of algorithms* $\mathsf{MA} = \{\mathsf{MA.K},\ \mathsf{MA.TAG},\ \mathsf{MA.VF}\}$*, where*

- $\mathsf{MA.K} \xrightarrow{\$} K$*: A probabilistic algorithm outputting a key* $K \in \mathcal{K}$*.*
- $\mathsf{MA.TAG}(K, m) \to t$*: A deterministic message authentication algorithm taking a key* $K \in \mathcal{K}$ *and a message* $m \in \mathcal{M}$ *as its input. An output of the algorithm ia s tag* $t \in \mathcal{T}$ *(message authentication code).*
- $\mathsf{MA.VF}(K, m, t) \to r$*: A deterministic algorithm verifying a message tag. An input of the algorithm is a key* $K \in \mathcal{K}$*, a message* $m \in \mathcal{M}$*, and a message tag* $t \in \mathcal{T}$*. An output of the algorithm is a result of tag verifying to be equal to* true *in the case of success, and* false*, otherwise.*

The standard notion for message authentication modes analysis is the PRF («Pseudorandom Function») model [19]. The formal description is presented in Appendix A.1. Informally, in this model the adversary has to distinguish the target mode under a random unknown key from a «truly» random function, having the capability to adaptively choose messages and obtain their tags. The distinguishability threat, considered in the model, is «easier» to implement than the other more intuitively understandable threats, such as key recovery or typical forgeries (universal, selective, existential).

Introduce the notion of a pseudorandom generator G.

Definition 3. *Let* $\mathcal{K} \subseteq \{0,1\}^*$ *be a set of states and* $\mathcal{B} \subseteq \{0,1\}^*$ *be a set of blocks. A* generator *is a pair of algorithms* $\mathsf{G} = \{\mathsf{G.K}, \mathsf{G.N}\}$*, where the deterministic algorithm* G.K *(key generation algorithm) sets an initial state of the generator* $St \in \mathcal{K}$*, the deterministic algorithm* G.N *(algorithm for calculating the next state) takes the current state* $St \in \mathcal{K}$ *as input and returns a block* $Out \in \mathcal{B}$*, viewed as the output of this stage, and an updated state, to be stored and used in the next invocation.*

The standard notion for generators analysis is the PRG («Pseudorandom Generator») model [16]. The formal description is presented in Appendix A.1. Informally, in this model the adversary has to distinguish the generator output

string from the string of the same length chosen at random according to the uniform distribution.

For brevity sake, hereinafter by $\mathsf{G.N}(St, i)$ denote the i-th block of the output sequence of the generator with the initial state St.

4 IND-sfCCSA **security model**

In order to estimate the security of the protocol from both the confidentiality and the integrity point of view we introduce the notion of an AEAD-scheme with internal state.

In the paper we consider only schemes the update function of which can depend on a previous state only and doesn't depend on a key, associated data, plaintext or ciphertext. Also an encryption state and a decryption state are supposed to be chosen from the same set and to be equal at the beginning of work.

Definition 4. *Let $\mathcal{K} \subseteq \{0,1\}^*$ be a set of keys, $\mathcal{M} \subseteq \{0,1\}^*$ be a set of messages, $\mathcal{AD} \subseteq \{0,1\}^*$ be a set of associated data, $\mathcal{C} \subseteq \{0,1\}^*$ be a set of ciphertexts, and $\mathcal{S}$ be a set of states. An* AEAD-scheme with internal state (stateful AEAD-scheme) *is a set of algorithms* $\mathsf{sfAEAD} = \{\mathsf{sfAEAD.K}, \mathsf{sfAEAD.Init}, \mathsf{sfAEAD.Upd}, \mathsf{sfAEAD.E}, \mathsf{sfAEAD.D}\}$, *where*

- $\mathsf{sfAEAD.K}() \xrightarrow{\$} K$*: A probabilistic key generation algorithm outputting a key $K \in \mathcal{K}$.*
- $\mathsf{sfAEAD.Init}(st) \to (st_E, st_D)$*: A deterministic algorithm for scheme initialization. An input of the algorithm is an initial state of the scheme $st \in \mathcal{S}$. An output of the algorithm is a pair of initial encryption $st_E = st \in \mathcal{S}$ and decryption $st_D = st \in \mathcal{S}$ states.*
- $\mathsf{sfAEAD.Upd}(st) \to st'$*: A deterministic algorithm taking a state $st \in \mathcal{S}$ (encryption or decryption state). An output of the algorithm is an updated state $st' \in \mathcal{S}$.*
- $\mathsf{sfAEAD.E}(K, ad, m, st_E) \xrightarrow{\$} c$*: A probabilistic algorithm of authenticated encryption taking a key $K \in \mathcal{K}$, associated data $ad \in \mathcal{AD}$, a plaintext $m \in \mathcal{M}$, and an encryption state$st_E \in \mathcal{S}$ as an input. An output of the algorithm is a ciphertext $c \in \mathcal{C}$.*

– $\mathsf{sfAEAD.D}(K, ad, c, st_D) \to m$: *A deterministic algorithm of authenticated decryption taking a key* $K \in \mathcal{K}$*, associated data* $ad \in \mathcal{AD}$*, a ciphertext* $c \in \mathcal{C}$*, and a decryption state* $st_D \in \mathcal{S}$*. An output of the algorithm is a plaintext* $m \in \mathcal{M}$ *or errpr symbol* $\perp$*.*

The stateful *AEAD*-scheme is *correct* if for all $K \in \mathcal{K}$, $m \in \mathcal{M}$, $ad \in \mathcal{AD}$ and $st \in \mathcal{S}$ such that $c \leftarrow \mathsf{sfAEAD.E}(K, ad, m, st)$, it is true that $\mathsf{sfAEAD.D}(K, ad, c, st) = m$. We consider only correct schemes.

Introduce an IND-sfCCSA («Indistinguishability under stateful Chosen Ciphertext and State Attack») security model to analyze stateful AEAD schemes. This model takes into account threats related to a message flow such as replay, dropping and shuffling messages. This model differs from stateful Authenticated Encryption (sfAE) model defined in [22] and [32]. In the sfAE model, only protocols that allow no more than one incorrect query to the decryption oracle are secure. However, there can be protocols which allow the adversary to forge a message with particular number more than one time. The IND-sfCCSA model allows adversaries to forge one message multiple times that extends the class of protocols we can analyze.

Definition 5. *The advantage of an adversary A in the model* IND-sfCCSA *for the stateful AEAD scheme* sfAEAD *is defined as:*

$$\mathsf{Adv}^{\text{IND-sfCCSA}}_{\mathsf{sfAEAD}}(A) = \\ = \Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA}-1}_{\mathsf{sfAEAD}}(A) \to 1\right] - \Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA}-0}_{\mathsf{sfAEAD}}(A) \to 1\right],$$

where experiments $\mathbf{Exp}^{\text{IND-sfCCSA}-b}_{\mathsf{sfAEAD}}(A)$*,* $b \in \{0, 1\}$*, are defined in the following way:*

$\underline{\mathbf{Exp}^{\text{IND-sfCCSA}-b}_{\text{sfAEAD}}(A)}$

$K \xleftarrow{\$} \mathsf{sfAEAD.K}()$
$u \leftarrow 0,\ v \leftarrow 0$
$sent \leftarrow \varnothing$
$st \leftarrow A$
$(st_E,\ st_D) \leftarrow \mathsf{sfAEAD.Init}(st)$
$b' \leftarrow A^{\mathsf{Encrypt}^b, \mathsf{Decrypt}^b}$
return b'

$\underline{\mathsf{Oracle\ Encrypt}^b(ad, m)}$

if $b = 0$ ***then***
$m \xleftarrow{\mathcal{U}} \{0,1\}^{|m|}$
end if
$c \leftarrow \mathsf{sfAEAD.E}(K, ad, m, st_E)$
$sent \leftarrow sent \cup (ad, c, u)$
$st_E \leftarrow \mathsf{sfAEAD.Upd}(st_E)$
$u \leftarrow u + 1$
return c

$\underline{\mathsf{Oracle\ Decrypt}^1(ad, c)}$

$m \leftarrow \mathsf{sfAEAD.D}(K, ad, c, st_D)$
if $(m \neq \perp)$ ***then***
if $((ad, c, v) \in sent)$ ***then***
$m \leftarrow \perp$
end if
$st_D \leftarrow \mathsf{sfAEAD.Upd}(st_D)$
$v \leftarrow v + 1$
end if
return m

$\underline{\mathsf{Oracle\ Decrypt}^0(ad, c)}$

return $\perp$

Note that the basic principle of defining experiments in the IND-sfCCSA model (the contents of the encrypted string and the procedure for the formation of decryption result) is similar to the principle of defining experiments in the basic models used for analyzing the schemes that aims to provide confidentiality and integrity only at the level of a single message. These basic models are constructed in such a way that epy decryption queries repeated the responses of the encryption oracle do not give the adversary any new information (such queries we will call *trivial*). This makes the model meaningful. Indeed, otherwise for any scheme there would be an adversary which could realize the threat in this model, i.e. the model would not allow anything to be said about the security properties of the scheme. The distinctive feature of the IND-sfCCSA model is that the decryption queries are trivial only when the message received in response to the encryption query with the number u is transmitted to the decryption oracle as the query with the same number (this is implemented using counters u and v). Thus, the responsibility for detecting replay, dropping or changing the order of messages now lies on the scheme, and the analysis in the IND-sfCCSA model reflects this security property of the scheme. Due to the fact that the counter v of decrypted messages increases only in case of

successful query processing, the adversary can try to forge the v-th message as many times as possible, making *testing* queries. In this case, the adversary can always change the attacked value of the counter by making the corresponding trivial query.

The IND-sfCCSA model is relevant only for analyzing unidirectional secure channels. However, most of bidirectional channels are symmetric and are obtained via establishing two unidirectional channels by the usage of one protocol on two independent keys. The paper [30] shows that a sufficient condition for the security of protocols providing a symmetric bidirectional channel running on independent keys is the security of the basic protocol providing unidirectional channel in the model similar to IND-sfCCSA. Therefore, models extended in the case of a bidirectional channel will not be considered in this paper.

5 Security bounds

5.1 Security bound of MtE-AD scheme in IND-sfCCSA model

Introduce a stateful AEAD-scheme of type MtE-AD («MAC-then-Encrypt-with-Associated-Data»).

Definition 6. *Let for sets $\mathcal{AD}$, $\mathcal{M}$, $\mathcal{M}_{\mathsf{MA}}$, $\mathcal{M}_{\mathsf{SE}}$, $\mathtt{IV}$, $\mathcal{T}$ and finite set $\mathcal{S}$ the following deterministic functions are defined:*

- $\mathsf{encode}_{\mathsf{MA}} \colon \mathcal{AD} \times \mathcal{M} \times \mathcal{S} \to \mathcal{M}_{\mathsf{MA}}$;
- $\mathsf{encode}_{\mathsf{SE}} \colon \mathcal{M} \times \mathcal{T} \to \mathcal{M}_{\mathsf{SE}}$, $\mathsf{decode}_{\mathsf{SE}} \colon \mathcal{M}_{\mathsf{SE}} \to \mathcal{M} \times \mathcal{T}$, *such that* $\mathsf{decode}_{\mathsf{SE}}(\mathsf{encode}_{\mathsf{SE}}(m,t)) = (m,t)$ *for all* $m \in M$, $t \in \mathcal{T}$;
- $\mathsf{StateToIV} \colon \mathcal{S} \to \mathtt{IV}$;
- $\mathsf{Next} \colon \mathcal{S} \to \mathcal{S}$.

Let MA *be a deterministic message authentication scheme for the sets $\mathcal{K}_{\mathsf{MA}}$, $\mathcal{M}_{\mathsf{MA}}$, $\mathcal{T}$. Let* SE *be an IV-based encryption scheme for the sets $\mathcal{K}_{\mathsf{SE}}$, $\mathcal{M}_{\mathsf{SE}}$, $\mathcal{C}_{\mathsf{SE}}$, $\mathtt{IV}$. Let $\mathcal{K} = (\mathcal{K}_{\mathsf{SE}} \times \mathcal{K}_{\mathsf{MA}})$ be a set of keys, $\mathcal{M} \subseteq \{0,1\}^*$ be a set of plaintexts, $\mathcal{AD}$ be a set of associated data, $\mathcal{S}$ be a set of states and $\mathcal{C}_{\mathsf{SE}} \subseteq \{0,1\}^*$ be a set of ciphertexts, and $\mathtt{IV} \subseteq \{0,1\}^*$ be a set of initialization vectors. A* stateful AEAD-scheme of type MtE-AD *is a set of algorithms* $\mathsf{sfAEAD} = (\mathsf{sfAEAD.K}, \mathsf{sfAEAD.Init}, \mathsf{sfAEAD.Upd}, \mathsf{sfAEAD.E}, \mathsf{sfAEAD.D})$ *where:*

$\underline{\mathsf{sfAEAD.K}:}$
$K_{\mathsf{SE}} \xleftarrow{\$} \mathsf{SE.K}()$
$K_{\mathsf{MA}} \xleftarrow{\$} \mathsf{MA.K}()$
$\boldsymbol{return} K$

$\underline{\mathsf{sfAEAD.Init}(st):}$
$st_E \leftarrow st$
$st_D \leftarrow st$
$\boldsymbol{return}(st_E, st_D)$

$\underline{\mathsf{sfAEAD.Upd}(st):}$
$st' \leftarrow \mathsf{Next}(st)$
$\boldsymbol{return} st'$

$\underline{\mathsf{sfAEAD.E}(K, ad, m, st_E)}$
$\widehat{m} \leftarrow \mathsf{encode}_{\mathsf{MA}}(ad, m, st_E)$
$t \leftarrow \mathsf{MA.TAG}(K_{\mathsf{MA}}, \widehat{m})$
$IV_E \leftarrow \mathsf{StateToIV}(st_E)$
$\widetilde{m} \leftarrow \mathsf{encode}_{\mathsf{SE}}(m, t)$
$c \xleftarrow{\$} \mathsf{SE.E}(K_{\mathsf{SE}}, IV_E, \widetilde{m})$
$st_E \leftarrow \mathsf{sfAEAD.Upd}(st_E)$
$\boldsymbol{return} c$

$\underline{\mathsf{sfAEAD.D}(K, ad, c, st_D)}$
$IV_D \leftarrow \mathsf{StateToIV}(st_D)$
$\widetilde{m} \leftarrow \mathsf{SE.D}(K_{\mathsf{SE}}, IV_D, c)$
$(m, t) \leftarrow \mathsf{decode}_{\mathsf{SE}}(\widetilde{m})$
$\widehat{m} \leftarrow \mathsf{encode}_{\mathsf{MA}}(ad, m, st_D)$
$\boldsymbol{if}\ \mathsf{MA.VF}(K_{\mathsf{MA}}, \widehat{m}, t) \neq \mathsf{true}\ \boldsymbol{then}$
  $\boldsymbol{return} \perp$
$\boldsymbol{end\ if}$
$st_D \leftarrow \mathsf{sfAEAD.Upd}(st_D)$
$\boldsymbol{return} m$

Definition 7. *Let* $\mathtt{IV}$ *be a set,* $\mathcal{S}$ *be a finite set. A function* $\mathsf{StateToIV}\colon \mathcal{S} \to \mathtt{IV}$ *is* an injective with according to a bijective function $\mathsf{Next}\colon \mathcal{S} \to \mathcal{S}$, *if* $\mathsf{StateToIV}(st) \neq \mathsf{StateToIV}(st')$ *for all* $st \neq st'$, $st, st' \in \mathcal{S}$, *such that* $\exists\ \alpha \in \mathbb{N}\colon \mathsf{Next}^{\alpha}(st) = st'$.

Definition 8. *Let* $\mathcal{AD}$, $\mathcal{M}$, $\mathcal{M}_{\mathsf{MA}}$ *be sets,* $\mathcal{S}$ *be a finite set. A function* $\mathsf{encode}_{\mathsf{MA}}\colon \mathcal{AD} \times \mathcal{M} \times \mathcal{S} \to \mathcal{M}_{\mathsf{MA}}$ *is* a collision free function with according to a bijective function $\mathsf{Next}\colon \mathcal{S} \to \mathcal{S}$, *if* $\mathsf{encode}_{\mathsf{MA}}(ad, m, st) \neq \mathsf{encode}_{\mathsf{MA}}(ad', m', st')$ *for all* $(ad, m, st) \neq (ad', m', st')$, $ad, ad' \in \mathcal{AD}$, $m, m' \in \mathcal{M}$, $st, st' \in \mathcal{S}$, *such that* $\exists\ \alpha \in \mathbb{N}\colon \mathsf{Next}^{\alpha}(st) = st'$.

Definition 9. *Let* $\mathcal{AD}$, $\mathcal{M}$, $\mathcal{M}_{\mathsf{MA}} \subseteq \{0,1\}^*$ *be sets,* $\mathcal{S}$ *be a finite set. A function* $\mathsf{encode}_{\mathsf{MA}}\colon \mathcal{AD} \times \mathcal{M} \times \mathcal{S} \to \mathcal{M}_{\mathsf{MA}}$ *is* r-adding, $r \in \mathbb{N}$ *if* $|\mathsf{encode}_{\mathsf{MA}}(ad, m, st)| \leqslant |ad| + |m| + r$ *for all* $ad \in \mathcal{AD}$, $m \in \mathcal{M}$.

Definition 10. *An* SE *encryption scheme is a* CRD-scheme *(Collision Resistant Decryption) if* $\mathsf{SE.D}(K, IV, c) \neq \mathsf{SE.D}(K, IV, c')$ *for all* $K \in \mathcal{K}_{\mathsf{SE}}$, $IV \in \mathtt{IV}$ *and* $c \neq c'$, $c, c' \in \mathcal{C}_{\mathsf{SE}}$.

Definition 11. *Let* $\mathsf{Next}\colon \mathcal{S} \to \mathcal{S}$ *be a bijective function. Then define the function* $\alpha\colon \mathcal{S} \to \mathbb{N}$ *as follows:* $\alpha(st) = \min\limits_{\alpha\colon \mathsf{Next}^{\alpha}(st)=st} \alpha$.

Theorem 1. *Let* sfAEAD *be a stateful AEAD-scheme of type MtE-AD and the following conditions hold:*

1. *the IV-based encryption scheme* SE *is a* CRD*-scheme;*
2. *the Message authentication scheme* MA *is such that the set* $\mathcal{T}$ *is* $\{0,1\}^{\tau}$*;*
3. Next *is a bijective function such that* $\alpha_{\min} = \min\limits_{st\in\mathcal{S}} \alpha(st)$*;*
4. StateToIV *is an injective function with according to* Next*;*
5. $\mathsf{encode}_{\mathsf{MA}}$ *is an* r*-adding collision free function with according to* Next*;*
6. $\mathsf{decode}_{\mathsf{SE}}$ *is injective.*

Let A *be an adversary for the* sfAEAD *scheme in the* IND-sfCCSA *model with time complexity at most* t*, making at most* $q_E \leqslant \alpha_{\min}$ *queries to the* Encrypt *oracle, at most* $\alpha_{\min}-1$ *trivial queries and at most* q_D *test queries to the* Decrypt *oracle with length at most* l*. Then there exists an adversary* B *for the* SE *scheme in the* ROR-CPNA *model, making at most* q_E *queries to the* Encrypt *oracle with length at most* $l+\tau$*, and exists an adversary* C *for the* MA *scheme in the* PRF *model, making at most* $q_E + q_D$ *queries to the* TAG *oracle with length at most* $l + r$*, such that*

$$\mathsf{Adv}^{\text{IND-sfCCSA}}_{\mathsf{sfAEAD}}(A) \leqslant 2 \cdot \mathsf{Adv}^{\text{PRF}}_{\mathsf{MA}}(C) + \mathsf{Adv}^{\text{ROR-CPNA}}_{\mathsf{SE}}(B) + \frac{q_D}{2^{\tau}}.$$

Furthermore, the time complexities of B *and* C *are at most* $t + c(q_E+q_D)(l+r+\tau)(T_{\mathsf{MA}}+T_{\mathsf{SE}})$*, where* T_{MA} *and* T_{SE} *is computational resources needed to process data with length at most* $l + r + \tau$ *by algorithms of the* MA *and* SE *schemes respectively,* c *is a constant that depends only on a model of computation and a method of encoding.*

The proof can be found in Appendix B.1

5.2 Security bound of MtE-AD scheme with key diversification

Consider a stateful AEAD-scheme of type MtE-AD with key diversification. We build such scheme $(\mathsf{sfAEAD}, \mathsf{G})$ from a scheme sfAEAD and a generator G.

Definition 12. *Let* $\mathcal{K} = \mathcal{B}$, $\mathcal{AD}$, $\mathcal{M}$, $\mathcal{C}$, $\mathcal{S}$ *be sets. Let* sfAEAD *be a stateful AEAD-scheme,* G *be a generator. A* stateful AEAD-scheme with key diversification *is a set of algorithms*

$$(\mathsf{sfAEAD}, \mathsf{G})_h = \\ = \{(\mathsf{sfAEAD}, \mathsf{G})_h.\mathsf{K},\ (\mathsf{sfAEAD}, \mathsf{G})_h.\mathsf{Init},\ (\mathsf{sfAEAD}, \mathsf{G})_h.\mathsf{Upd},\\ (\mathsf{sfAEAD}, \mathsf{G})_h.\mathsf{E},\ (\mathsf{sfAEAD}, \mathsf{G})_h.\mathsf{D}\}$$

where:

$\underline{(\mathsf{sfAEAD}, \mathsf{G})_h.\mathsf{K}:}$
$K \xleftarrow{\$} \mathsf{G.K}()$
return K

$\underline{(\mathsf{sfAEAD}, \mathsf{G})_h.\mathsf{Init}(st):}$
$(st_E, st_D) \leftarrow \mathsf{sfAEAD.Init}(st)$
return $(st_E, 0), (st_D, 0)$

$\underline{(\mathsf{sfAEAD}, \mathsf{G})_h.\mathsf{Upd}(st):}$
$st'.st \leftarrow \mathsf{sfAEAD.Upd}(st.st)$
$st'.u \leftarrow st'.u + 1$
return st'

$\underline{(\mathsf{sfAEAD}, \mathsf{G})_h.\mathsf{E}(K, ad, m, st_E)}$
$i \leftarrow \lfloor st_E.u/h \rfloor$
$K_i \leftarrow \mathsf{G.N}(K, i)$
$c \leftarrow \mathsf{sfAEAD.E}(K_i, ad, m, st_E.st)$
return c

$\underline{(\mathsf{sfAEAD}, \mathsf{G})_h.\mathsf{D}(K, ad, c, st_D)}$
$i \leftarrow \lfloor st_E.u/h \rfloor$
$K_i \leftarrow \mathsf{G.N}(K, i)$
$m \leftarrow \mathsf{sfAEAD.D}(K_i, ad, m, st_D.st)$
return m

The following theorem shows how key diversification affects the security of the stateful AEAD-scheme sfAEAD.

Theorem 2. *Let* A *be an adversary with time complexity at most* t *in the* IND-sfCCSA *model for the* $(\mathsf{sfAEAD}, \mathsf{G})_h$ *scheme with fixed* h*, making at most* q_E *queries to the* Encrypt *oracle and at most* q_D *test queries to the* Decrypt *oracle with length at most* l*. Then there exists an adversary* B *in the* IND-sfCCSA *model for the* sfAEAD *scheme making at most* $\min(q_E, h)$ *queries to the* Encrypt *oracle and at most* q_D *test queries to the* Decrypt *oracle with length at most* l *bits, and exists an adversary* D *in the* PRG *model for the* G *generator, making the query with value at most* $N = \lceil q_E/h \rceil$*, such that:*

$$\mathsf{Adv}^{\text{IND-sfCCSA}}_{(\mathsf{sfAEAD},\mathsf{G})_h}(A) \quad \leqslant \quad N \cdot \mathsf{Adv}^{\text{IND-sfCCSA}}_{\mathsf{sfAEAD}}(B) + 2 \cdot \mathsf{Adv}^{\text{PRG}}_{\mathsf{G}}(D).$$

Furthermore, the time complexities of B *and* D *are at most* $t + clN(q_E + q_D)T_{\mathsf{sfAEAD}}$*, where* T_{sfAEAD} *is computational resources needed to process data*

with length at most l by algorithm of the sfAEAD *scheme, c is a constant that depends only on a model of computation and a method of encoding.*

The proof can be found in Appendix B.2.

5.3 Final bounds

Summarizing the results of Theorem 1 and Theorem 2 we claim the following theorem:

Theorem 3. *Let* sfAEAD *be a stateful AEAD-scheme of type MtE-AD fulfilling the Theorem 1 conditions. Let A be an adversary with time complexity at most t in the* IND-sfCCSA *model for the $(\mathsf{sfAEAD}, \mathsf{G})_h$ scheme with fixed h making at most q_E queries to the* Encrypt *oracle and at most q_D test queries to the* Decrypt *oracle with length at most l bits. Then there exists an adversary B for the* SE *scheme in the* ROR-CPNA *model making at most $\min(q_E, h)$ queries to the* Encrypt *oracle with length at most $l + \tau$ bits, exists an adversary C for* MA *the scheme in the* PRF *model making at most $\min(q_E, h) + q_D$ queries to the* TAG *oracle with length at most $l + r$, and exists an adversary D in the* PRG *model for the* G *generator, making the query with value at most $N = \lceil q_E/h \rceil$, such that:*

$$\mathsf{Adv}^{\text{IND-sfCCSA}}_{(\mathsf{sfAEAD},\mathsf{G})_h}(A) \leqslant$$
$$\leqslant 2 \cdot \mathsf{Adv}^{\text{PRG}}_{\mathsf{G}}(D) + N \cdot \mathsf{Adv}^{\text{ROR-CPNA}}_{\mathsf{SE}}(B) + 2N \cdot \mathsf{Adv}^{\text{PRF}}_{\mathsf{MA}}(C) + \frac{Nq_D}{2^n}.$$

Furthermore, the time complexities of B and C are at most $t + c(q_E + q_D)(l + r + \tau)(T_{\mathsf{MA}} + T_{\mathsf{SE}}) + clN(q_E + q_D)T_{\mathsf{sfAEAD}}$ and the time complexity of D is at most $t + clN(q_E + q_D)T_{\mathsf{sfAEAD}}$, where T_{MA} and T_{SE} is computational resources needed to process data with length at most $l + r + \tau$ by algorithms of the MA *and* SE *schemes respectively, T_{sfAEAD} is computational resources needed to process data with length at most l by algorithm of the* sfAEAD *scheme, c is a constant that depends only on a model of computation and a method of encoding.*

6 Russian ciphersuites for TLS 1.2

The Record protocol provides bidirectional secure channel therefore during a Handshake protocol running, every side generate key material for sending and receiving messages separately.

We describe the Record protocol only in the case when the data is transmitted unidirectionally, because the bidirectional channel is symmetrical and the key material is chosen independent at random according to uniform distribution (this follows from the Handshake security assumption).

During a correct running of the Record protocol the sender must count sent messages and the receiver must count received messages. The Record Protocol takes messages to be transmitted, fragments the data into blocks, forms from every block a record with the latest number, and transmits the result. Received data is interpreted as a record with certain number and then decrypted, verified, and delivered to higher-level protocols according to the header of this record.

The procedure of formation of a protected record in the Record protocol, defined by the Russian ciphersuites, corresponds to the use of an AEAD-scheme with internal state and re-keying. Here the MAC-then-Encrypt scheme is used as a AEAD-scheme with internal state. This MAC-then-Encrypt scheme is based on the OMAC and CTR-ACPKM block cipher modes of the Magma or Kuznyechik cipher (further this scheme is called a TLS-REC scheme). The key tree construction algorithm, defined by the algorithm TLSTREE [8], is used as a generator for re-keying (further this algorithm is called a TREE generator).

Now assume that a key $K = (K_{\mathsf{MA}}, K_{\mathsf{SE}})$ and IV were produced by the Handshake protocol, where $K_{\mathsf{MA}} \in \{0,1\}^k$ is a key for the OMAC mode, $K_{\mathsf{SE}} \in \{0,1\}^k$ and $IV \in \{0,1\}^{n/2}$ are a key and an initialization vector for the CTR-ACPKM mode, k and n are a key length and block length of the cipher to be used. For processing of the record with sequential number sn, new keys are generated:

$$K^{sn} = (K^{sn}_{\mathsf{MA}}, K^{sn}_{\mathsf{SE}}) = (\mathrm{TLSTREE}(K_{\mathsf{MA}}, sn), \mathrm{TLSTREE}(K_{\mathsf{SE}}, sn)) .$$

Let m be a data block produced during the fragmentation of the original message which should be protected. Three stages can be distinguished in the formation process of a protected record $PRec$, with a sequential number sn, from m.

1. Formation of the unprotected record header: $header = t\|v\|l$, where $t \in \{0,1\}^8$ is the record type (1 byte), $v \in \{0,1\}^{16}$ is the protocol version (2 bytes), $l \in \{0,1\}^{16}$ is the byte representation of the byte length of m (2 bytes).

2. Formation of the protected record header: $header' = t\|v\|\mathsf{str}_{16}(\mathsf{int}(l) +$

$|mac|$).

3. Formation of the protected record: $PRec = header' \| c$, where c is calculated by the following way: $c = \text{TLS-REC.}\mathsf{E}(K^{sn}, header, m, (IV, sn))$.
   (a) Formation of the unprotected record $Rec = header \| m$.
   (b) Calculation of the message authentication code: $mac = \text{OMAC.}\mathsf{TAG}(K^{sn}_{\mathsf{MA}}, \mathsf{str}_{64}(sn) \| Rec)$.
   (c) Calculation of the current initialization vector: $IV^{sn} = \mathsf{str}_{n/2}((\mathsf{int}(IV) + sn) \mod 2^{n/2})$.
   (d) Encryption of the unprotected record payload and the message authentication code from the previous step: $c = \text{CTR-ACPKM.}\mathsf{E}(K^{sn}_{\mathsf{SE}}, IV^{sn}, m \| mac)$.

The receiver decrypts the protected record $PRec$ with the sequential number sn by the following way:

1. Formation of the header $header' = t \| v \| l$ of the protected record $PRec$ and verification of correctness of its format. If the result of verification is error, then the error code *unexpected_message* or *decode_error* is sent, and the connection is terminated by the receiver.

2. Verification of the following condition on the value l from the header: the length $\mathsf{int}(l)$ doesn't exceed $2^{14} + 16$ bytes. If the verification result is error, then the error code *record_overflow* is sent, and the connection is terminated by the receiver.

3. Accumulation of the data of length $\mathsf{int}(l)$ by the receiver to form a ciphertext c.

4. Formation of the unprotected record header: $header = t \| v \| \mathsf{str}_{16}(\mathsf{int}(l) - |mac|)$.

5. Formation of the record $Rec = header \| m$, where m is calculated by the following way: $m = \text{TLS-REC.}\mathsf{D}(K^{sn}, header, c, (IV, sn))$:
   (a) Calculation of the current initialization vector: $IV^{sn} = \mathsf{str}_{n/2}((\mathsf{int}(IV) + sn) \mod 2^{n/2})$.
   (b) Decryption of the record payload and the message authentication code: $m \| mac = \text{CTR-ACPKM.}\mathsf{D}(K^{sn}_{\mathsf{SE}}, IV^{sn}, c)$.

(c) Formation of the unprotected record $Rec = header \| m$.

(d) Verification of the message authentication code: $\mathsf{true} \stackrel{?}{=} \mathrm{OMAC}.\mathsf{VF}(K^{sn}_{\mathsf{MA}}, \mathsf{str}_{64}(sn) \| Rec, mac)$. If the verification result is error, then the error code bad_record_mac is sent, and the connection is terminated by the receiver.

6.1 Relevance of the security model

Before analyzing the target protocol in the IND-sfCCSA model, it is necessary to make sure that the model really covers all the capabilities of an adversary, which it has in practice and which largely depend on a specific implementation of the protocol.

Below we present the justifications for accordance the capabilities, provided by the experiment in the IND-sfCCSA model, to the real capabilities of the active adversary.

— The IND-sfCCSA model allows the adversary to make the encryption and decryption queries with arbitrary associated data, while the Record protocol allows sending and receiving only those records which header satisfies a strictly defined format. Note that the fact of success or failure of verifying the header format does not give the adversary any additional secret information about the internal state of the protocol, since the header is transmitted in the channel in the open form and is verified immediately before the start of the record decryption. Therefore, the theoretical capabilities of the adversary in this case are even wider than practical ones.

— The decryption oracle in the IND-sfCCSA model can return either plaintext or the error symbol $\perp$, which in the Record protocol corresponds to the bad_record_mac error code. Other error codes can occur only when verifying the header format, and in accordance with the preceding rationale item, the absence of analysis of these errors in the model cannot lead to cryptographically dangerous consequences, therefore, consideration of the type of attack with processing various errors (IND$-sfCCA3 [20], SAE [14]) is redundant.

— The IND-sfCCSA model allows the adversary to encrypt and decrypt data of any bit length, unlike the Record byte-oriented protocol, which controls the record length by verifying the header ($record\_overflow$ error). In the

model, the query length is the parameter of the adversary, the restriction on which is taken into account when obtaining specific estimates of the insecurity value.

— Despite data to be protected is received from the transport layer protocol as a stream, the Record protocol starts processing of the record and the subsequent transfer of service information (for example, an upper level protocol error code) to the channel only after accumulating enough data necessary to form the entire record. Therefore, the queries processing perceived by the oracles corresponds to the message processing by protocol on practice, and, thus, consideration of the type of attack with the possibility of ciphertext fragmentation (CFA [33]), as well as attacks with adaptive selection of text blocks (BCPA [27] and IND-BLK-CCA [21]), is redundant.

— The IND-sfCCSA model does not take into account the time of query processing by the oracles, which obviously does not correspond to the practice, where the time depends primarily on the record length. However, in the case of the Record protocol construction, the record length is not confidential information, since it is written to the header. Length hiding property is useful and was introduced in TLS 1.3, but when using TLS 1.2 this property should be achieved by some higher-level solutions, so we do not take this property into account when analyzing TLS 1.2.

— The Record protocol does not transfer an incoming message to the application level until it is verified for integrity, which means that consideration of the INT-RUP and AE-RUP [12] models is redundant.

7 Applying results to TLS-REC

7.1 TLS-REC scheme

Consider the stateful MtE-AD scheme TLS-REC. In TLS-REC the OMAC function with $\mathcal{T} = \{0,1\}^n$ is used as an MA scheme, and the CTR-ACPKM encryption mode with $\mathtt{IV} = \{0,1\}^{n/2}$ is used as a CRD-scheme SE. For TLS-REC the set $\mathcal{AD}$ is the set $\{0,1\}^{40}$ (5 bytes) and the set $\mathcal{S}$ is the set $\mathtt{IV} \times \mathbb{Z}_{2^{n/2}}$. Then we define functions from Definition 6 as follows:

– $\mathsf{encode}_{\mathsf{MA}}(ad, m, st) \to \widehat{m} = \mathsf{str}_{64}(st.sn) \| ad \| m$;

– $\mathsf{encode}_{\mathsf{SE}}(m,t) \to \widetilde{m} = m\|t$, $\mathsf{decode}_{\mathsf{SE}}(\widetilde{m}) \to (m,t) = (\mathsf{msb}_{|\widetilde{m}|-\tau}(\widetilde{m}), \mathsf{lsb}_{\tau}(\widetilde{m}))$;

– $\mathsf{StateToIV}(st) \to IV = \mathsf{str}_{n/2}((st.sn + \mathsf{int}(st.IV)) \mod 2^{n/2})$;

– $\mathsf{Next}(st) \to st' = (st.IV, (st.sn + 1) \mod 2^{n/2})$.

It is easy to see that Next is bijective. Note that if for $st, st' \in \mathcal{S}$ exists α: $\mathsf{Next}^{\alpha}(st) = st'$ then $st.IV = st'.IV$ (denote this property by *). Therefore, $\alpha(st) = 2^{n/2}$ for all $st = (IV, sn)$. Obviously, decode is injective. The function $\mathsf{encode}_{\mathsf{MA}}$ is defined as concatenation of fixed length bit strings $\mathsf{str}_{64}(st.sn) \in \{0,1\}^{64}$, $ad \in \{0,1\}^{40}$ and variable length string $m \in \{0,1\}^*$. Due to this and the property *, $\mathsf{encode}_{\mathsf{MA}}$ is a 64-adding collision free function with according to Next. If for st, $st' \in \mathcal{S}$ the property * holds and $st'.sn \neq st.sn$, then $(st.sn + \mathsf{int}(st.IV)) \not\equiv (st.sn' + \mathsf{int}(st.IV)) \mod 2^{n/2}$. Therefore, the StateToIV function is injective with according to Next.

Then we can apply the results of Theorem 1 to the TLS-REC scheme.

Now consider the TREE generator. Note that the C_3 constant in the TLSTREE definition specifies the parameter h — the amount of messages processed on one «leaf» key. Therefore we can define the TREE algorithms as follows:

1. $\text{TREE.}\mathsf{K}() \to K = (K_{\mathsf{MA}}, K_{\mathsf{SE}})$
2. $\text{TREE.}\mathsf{N}(K, i) \to K_i = (\text{TLSTREE}(K_{\mathsf{MA}}, i \cdot h), \text{TLSTREE}(K_{\mathsf{SE}}, i \cdot h))$

Consider the $(\text{TLS-REC}, \text{TREE})_h$ scheme with fixed h. Note that an adversary can not choose initial state (IV, sn) when attacking Record protocol because it is equal to $(IV', 0)$ where IV' is a part of key material produced by the Handshake protocol. So the capabilities of adversaries in the IND-sfCCSA model are wider then in practice. Note that for initial state $(IV', 0)$ the scheme $(\text{TLS-REC}, \text{TREE})_h$ is equal to the one described in Section 6.

Thus, we can apply the results of Theorem 3 to the $(\text{TLS-REC}, \text{TREE})_h$ scheme with fixed h.

7.2 Applying obtained bounds on practice

Let M be some security model and $\mathsf{Adv}^M(\mathcal{A})$ is some characteristic of possibilities of $\mathcal{A}$ when implementing the threat in the model M. Let T be a tuple

of limitations on computational resources of an adversary and any other parameters characterizing its work in the model M. T may additionally contain the values limiting the total number of queries to the oracles considered by the model M. For example, in the ROR-CPNA model this tuple contains time complexity of the adversary, number of queries to encryption oracle and the maximal length of one query. The set $\mathcal{A}(T)$ of the adversaries, satisfying the limitations that are defined by T, is finite. By $\mathsf{InSec}^M(T)$ we'll denote the maximal value $\mathsf{Adv}^M(\mathcal{A})$ for adversaries $\mathcal{A}$ in the set $\mathcal{A}(T)$.

Consider particular numbers of messages processed on one key and queries, those are defined in [8] and [25]. The TLS 1.2 specification define that if the MAC value is incorrect then the connection must be aborted with *bad_record_mac* fatal error. So number of test queries q_D is at most 1. Furthermore, the length of one record is no more then $l = 2^{17} + n$ bits, where n is a bit length of the MAC tag.

Considering HMAC as a base primitive, we bound $\mathsf{InSec}^{\text{PRF}}_{\text{HMAC}}(t, q)$ as $\frac{q^2}{2^{256}}$ using estimates from [29]. Note that to process q_E messages we need $\left\lceil \frac{q_E}{h} \right\rceil$ keys, where h is a number of messages processed on one key. Let the section length in the CTR-ACPKM mode be s bits. Then the result of Theorem 3 applied to $(\text{TLS-REC}, \text{TREE})_h$ scheme can be rewritten using InSec notion.

$$\begin{aligned}
&\mathsf{InSec}^{\text{IND-sfCCSA}}_{(\text{TLS-REC},\text{TREE})_h}(t, q_E, 1, l) \leqslant \\
&\leqslant 2 \cdot \mathsf{InSec}^{\text{PRG}}_{\text{TLSTREE}}(t_1, N) + N \cdot \mathsf{InSec}^{\text{ROR-CPNA}}_{\text{CTR-ACPKM}}(t_2, h, \lceil l/n \rceil + 1) + \\
&+ 2N \cdot \mathsf{InSec}^{\text{PRF}}_{\text{OMAC}}(t_3, h+1, \lceil l/n \rceil + 1) + \left\lceil \frac{q_E}{h} \right\rceil \cdot \frac{1}{2^n}.
\end{aligned} \tag{1}$$

There values t_1, t_2, t_3 are obtained as computational complexity of adversaries from Theorem 3. Apply bounds of TLSTREE, CTR-ACPKM and OMAC security from Appendix C.

$$\mathsf{InSec}^{\text{IND-sfCCSA}}_{(\text{TLS-REC,TREE})_h}(t, q_E, 1, l) \leqslant$$
$$2 \cdot 2 \cdot \left(1 + \left\lceil \frac{N}{(2^{13})^2} \right\rceil + \left\lceil \frac{N}{2^{13}} \right\rceil\right) \cdot \mathsf{InSec}^{\text{PRF}}_{\text{HMAC}}(t_4, 2^{13}, 2) +$$
$$+ \left\lceil \frac{q_E}{h} \right\rceil \cdot \frac{2^{17}}{s} \cdot \mathsf{InSec}^{\text{PRP-CPA}}_{E}(t_5, \frac{s}{n} + 2) +$$
$$+ \left\lceil \frac{q_E}{h} \right\rceil \cdot \frac{\frac{2^{17}}{s} \cdot (\frac{s}{n} + 2)^2}{2^{n+1}} + 2 \left\lceil \frac{q_E}{h} \right\rceil \cdot \mathsf{InSec}^{\text{PRP-CPA}}_{E}(t_6, (h+1) \cdot (\lceil l/n \rceil + 1) + 1) +$$
$$+ \left\lceil \frac{q_E}{h} \right\rceil \cdot \frac{4 \cdot (h+1)^2 \cdot (\lceil l/n \rceil + 1)^2}{2^n} + \left\lceil \frac{q_E}{h} \right\rceil \cdot \frac{1}{2^n}.$$

Consider used block cipher E as ideal (then $\mathsf{InSec}^{\text{PRP-CPA}}_{E}(t, q) = 0$). Applying HMAC and block cipher bounds we achieve the following estimate:

$$\mathsf{InSec}^{\text{IND-sfCCSA}}_{(\text{TLS-REC,TREE})_h}(t, q_E, 1, l) \leqslant \left\lceil \frac{q_E}{h} \right\rceil \cdot \frac{4}{2^{11}} \cdot \frac{2^{27}}{2^{256}} + \left\lceil \frac{q_E}{h} \right\rceil \cdot \frac{\frac{2^{17}}{s} \cdot (\frac{s}{n} + 2)^2}{2^{n+1.}} +$$
$$+ \left\lceil \frac{q_E}{h} \right\rceil \cdot \frac{4 \cdot (h+1)^2 \cdot (\lceil l/n \rceil + 1)^2}{2^n} + \left\lceil \frac{q_E}{h} \right\rceil \cdot \frac{1}{2^n}$$

Note that if q_E is less then h then we can rewrite bound 1 as:

$$\mathsf{InSec}^{\text{IND-sfCCSA}}_{(\text{TLS-REC,TREE})_h}(t, q_E, 1, l) \leqslant$$
$$\leqslant 2 \cdot \mathsf{InSec}^{\text{PRG}}_{\text{TLSTREE}}(t_1, 1) + \mathsf{InSec}^{\text{ROR-CPNA}}_{\text{CTR-ACPKM}}(t_2, q_E, \lceil l/n \rceil + 1) +$$
$$+ 2 \cdot \mathsf{InSec}^{\text{PRF}}_{\text{OMAC}}(t_3, q_E + 1, \lceil l/n \rceil + 1) + \frac{1}{2^n}. \quad (2)$$

Analogously applying abovementioned bounds we receive:

$$\mathsf{InSec}^{\text{IND-sfCCSA}}_{(\text{TLS-REC,TREE})_h}(t, q_E, 1, l) \leqslant$$
$$\leqslant \frac{12}{2^{256}} + \frac{\frac{2^{17}}{s} \cdot (\frac{s}{n} + 2)^2}{2^{n+1}} + \frac{4 \cdot (q_E + 1)^2 \cdot (\lceil l/n \rceil + 1)^2}{2^n} + \frac{1}{2^n}$$

With particular values of s, n and h we get the bounds presented in Table 1.

Ciphersuite	s	n	h	Security bound	
				$q_E < h$	$q_E \geqslant h$
..._KUZNYECHIK_CTR_OMAC	2^{15}	2^7	64	$q_E^2 \cdot (l + 2^7)^2 \cdot 2^{-140}$	$q_E \cdot (l + 2^7)^2 \cdot 2^{-126}$
..._MAGMA_CTR_OMAC	2^{13}	2^6	4096	$q_E^2 \cdot (l + 2^6)^2 \cdot 2^{-74}$	$q_E \cdot (l + 2^6)^2 \cdot 2^{-62}$

Table 1: Security bounds for TLS-REC ciphersuites.

8 Conclusion

This paper introduces the IND-sfCCSA model that allows to analyze a wide class of protocols those provide secure channels. We obtain security bounds for the general stateful MAC-then-Encrypt with associated data scheme and for the general stateful AEAD scheme with pseudorandom generator. The presented theorems allows to estimate the security of the above-mentioned scheme by the security of the used cryptographic schemes such as encryption mode, MAC scheme and pseudorandom generator. This paper shows the relevance of the IND-sfCCSA model for the Record protocol and presents the security bounds for the new specification of this protocol defined by the Russian ciphersuites (bounds 1, 2; Table 1).

The open problems are to prove the tightness of the bounds or to improve them. The second objective can be achieved by improving the proof in the IND-sfCCSA model or by improving the model. One of the way to improve the model is to make the capabilities of adversary more real, for example do not let him set or know the IV value, since in TLS 1.2 with the Russian ciphersuites IV is a part of secret key material.

Acknowledgments

We thank Grigory A. Karpunin for useful discussions and comments during this work.

References

[1] SSL pulse. `https://www.ssllabs.com/ssl-pulse/`. Accessed: 04-03-2019.

[2] «Information technology. Cryptographic protection of information. Block cipher modes of operation» (in Russian). National standard of the Russian Federation GOST R 34.13-2015, STANDARTINFORM, 01.01.2016.

[3] «Information technology. Cryptographic protection of information. Block ciphers» (in Russian). National standard of the Russian Federation GOST R 34.12-2015, STANDARTINFORM, 19.06.2015.

[4] «Information technology. Cryptographic data security. Using ciphersuites based on GOST 28147-89 for Transport Layer Security (TLS)» (in Russian). Guidelines, Technical Committee For Standardization «Cryptography and Security Mechanisms», 2013.

[5] «Information technology. Cryptographic data security. Cryptographic algorithms to accompany the usage of digital signature and hash function algorithms» (in Russian). Recommendations for standardization, Federal Agency for Technical Regulation and Metrology (ROSSTANDART), 2016.

[6] «Information technology. Cryptographic data security. Parameters of elliptic curves for cryptographic algorithms and protocols» (in Russian). Recommendations for standardization, Federal Agency for Technical Regulation and Metrology (ROSSTANDART), 2016.

[7] «Information technology. Cryptographic data security. Cryptographic algorithms accompanying the use of block ciphers» (in Russian). Recommendations for standardization, Federal Agency for Technical Regulation and Metrology (ROSSTANDART), 2017.

[8] «Information technology. Cryptographic data security. The use of the Russian cryptographic algorithms in the Transport Layer Security protocol (TLS 1.2)» (in Russian). Recommendations on standardization, Federal Agency for Technical Regulation and Metrology (ROSSTANDART), 2017.

[9] Abdalla, Michel and Bellare, Mihir. Increasing the Lifetime of a Key: A Comparative Analysis of the Security of Re-keying Techniques. In Okamoto, Tatsuaki, editor, *Advances in Cryptology — ASIACRYPT 2000*, pages 546–559, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

[10] L.R. Ahmetzyanova, E.K. Alekseev, G.K. Sedov, E.S. Smyshlyaeva, and S.V. Smyshlyaev. Security bounds for standardized internally re-keyed block cipher modes and their practical significance. CTCrypt'18, 2018.

[11] E.K. Alekseev, I.B. Oshkin, V.O. Popov, and S.V. Smyshlyaev. On the cryptographic properties of algorithms accompanying the applications of standards gost r 34.11-2012 and gost r 34.10-2012 (in russian). *Mathematical Aspects of Cryptography*, 7(1):5–38, 2016.

[12] Elena Andreeva, Andrey Bogdanov, Atul Luykx, Bart Mennink, Nicky Mouha, and Kan Yasuda. How to securely release unverified plaintext in authenticated encryption. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 105–125. Springer, 2014.

[13] Gregory V Bard. The vulnerability of ssl to chosen plaintext attack. *IACR Cryptology ePrint Archive*, 2004(111), 2004.

[14] Guy Barwell, Dan Page, and Martijn Stam. Rogue Decryption Failures: Reconciling AE Robustness Notions. Cryptology ePrint Archive, Report 2015/895, 2015. `https://eprint.iacr.org/2015/895`.

[15] Mihir Bellare and Chanathip Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 531–545. Springer, 2000.

[16] Mihir Bellare and Phillip Rogaway. Introduction to modern cryptography., 2004. URL: `http://www-cse.ucsd.edu/users/mihir/cse207`.

[17] Mihir Bellare and Phillip Rogaway. Introduction to modern cryptography, chapter 2: Block Ciphers, 2004. URL: `http://www-cse.ucsd.edu/users/mihir/cse207`.

[18] Mihir Bellare and Phillip Rogaway. Introduction to modern cryptography, chapter 4: Symmetric Encryption, 2004. URL: `http://www-cse.ucsd.edu/users/mihir/cse207`.

[19] Mihir Bellare and Phillip Rogaway. Introduction to modern cryptography, chapter 7: Message Authentication, 2004. URL: `http://www-cse.ucsd.edu/users/mihir/cse207`.

[20] Alexandra Boldyreva, Jean Paul Degabriele, Kenneth G Paterson, and Martijn Stam. On symmetric encryption with distinguishable decryption failures. In *International Workshop on Fast Software Encryption*, pages 367–390. Springer, 2013.

[21] Alexandra Boldyreva and Nut Taesombut. Online encryption schemes: New security notions and constructions. In *Cryptographers' Track at the RSA Conference*, pages 1–14. Springer, 2004.

[22] Colin Boyd, Britta Hale, Stig Frode Mjølsnes, and Douglas Stebila. From stateless to stateful: Generic authentication and authenticated encryption constructions with application to tls. In *Cryptographers' Track at the RSA Conference*, pages 55–71. Springer, 2016.

[23] David Brumley and Dan Boneh. Remote timing attacks are practical. *Computer Networks*, 48(5):701–716, 2005.

[24] Brice Canvel, Alain Hiltgen, Serge Vaudenay, and Martin Vuagnoux. Password interception in a ssl/tls channel. In *Annual International Cryptology Conference*, pages 583–599. Springer, 2003.

[25] Tim Dierks and Eric Rescorla. The transport layer security (tls) protocol version 1.2. Technical report, 2008.

[26] Iwata, Tetsu and Kurosawa, Kaoru. Stronger Security Bounds for OMAC, TMAC, and XCBC. In Johansson, Thomas and Maitra, Subhamoy, editor, *Progress in Cryptology - INDOCRYPT 2003*, pages 402–415, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[27] Antoine Joux, Gwenaëlle Martinet, and Frédéric Valette. Blockwise-adaptive attackers revisiting the (in) security of some provably secure encryption modes: Cbc, gem, iacbc. In *Annual International Cryptology Conference*, pages 17–30. Springer, 2002.

[28] Hugo Krawczyk. The order of encryption and authentication for protecting communications (or: How secure is ssl?). In *Annual International Cryptology Conference*, pages 310–331. Springer, 2001.

[29] Gaëtan Leurent, Thomas Peyrin, and Lei Wang. New generic attacks against hash-based macs. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology - ASIACRYPT 2013*, pages 1–20, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[30] Giorgia Azzurra Marson and Bertram Poettering. Security notions for bidirectional channels. *IACR Transactions on Symmetric Cryptology*, pages 405–426, 2017.

[31] Bodo Möller, Thai Duong, and Krzysztof Kotowicz. This poodle bites: exploiting the ssl 3.0 fallback. *Security Advisory*, 2014.

[32] Kenneth G Paterson, Thomas Ristenpart, and Thomas Shrimpton. Tag size does matter: Attacks and proofs for the tls record protocol. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 372–389. Springer, 2011.

[33] Kenneth G Paterson and Gaven J Watson. Plaintext-dependent decryption: A formal security treatment of ssh-ctr. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 345–361. Springer, 2010.

[34] Eric Rescorla. The Transport Layer Security (TLS) Protocol Version 1.3. PROPOSED STANDARD RFC 8446, Internet Engineering Task Force, August 2018.

[35] J Rizzo and T Duong. Browser exploit against ssl/tls. 2011. In *Ekoparty security conference–2011 (Buenos Aires*, 2011.

[36] Rogaway, Phillip. Nonce-Based Symmetric Encryption. In Roy, Bimal and Meier, Willi, editor, *Fast Software Encryption*, pages 348–358, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[37] Serge Vaudenay. Security flaws induced by cbc padding—applications to ssl, ipsec, wtls... In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 534–545. Springer, 2002.

[38] David Wagner, Bruce Schneier, et al. Analysis of the ssl 3.0 protocol. In *The Second USENIX Workshop on Electronic Commerce Proceedings*, volume 1, pages 29–40, 1996.

A Security models

A.1 Basic security models

In this appendix we formally define basic security models.

Definition 13. *For a cipher E with parameters n and k define*

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

where the experiments $\mathbf{Exp}_E^{\text{PRP-CPA}-1}(A)$ and $\mathbf{Exp}_E^{\text{PRP-CPA}-0}(A)$ are defined in the following way:

$\underline{\mathbf{Exp}_E^{\text{PRP-CPA}-b}(A)}$
if $b = 0$ ***then***
  $P \xleftarrow{\mathcal{U}} Perm(\{0,1\}^n)$
else
  $K \xleftarrow{\mathcal{U}} \{0,1\}^k$
end if
$b' \xleftarrow{\$} A^{\mathsf{P}^b}$
return b'

$\underline{\mathsf{OracleP}^b(M),\ M \in \{0,1\}^n}$
if $b = 0$ ***then***
  return $P(M)$
else
  return $E_K(M)$
end if

Definition 14. *For an IV-based encryption scheme* SE *define advantage of adversary A in* ROR-CPNA *model as*

$$\mathsf{Adv}_{\mathsf{SE}}^{\text{ROR-CPNA}}(A) = \\ = \Pr\left[\mathbf{Exp}_{\mathsf{SE}}^{\text{ROR-CPNA}-1}(A) \to 1\right] - \Pr\left[\mathbf{Exp}_{\mathsf{SE}}^{\text{ROR-CPNA}-0}(A) \to 1\right],$$

where the experiments $\mathbf{Exp}_{\mathsf{SE}}^{\text{ROR-CPNA}-b}(A),\ b \in \{0,1\}$ are defined in the following way:

$\underline{\mathbf{Exp}_{\mathsf{SE}}^{\text{ROR-CPNA}-b}(A)}$

$K \xleftarrow{\$} \mathcal{K}$
$used \leftarrow \varnothing$
$b' \leftarrow A^{\mathsf{Encrypt}^b(\cdot)}$
return b'

$\underline{\mathsf{Oracle\ Encrypt}^b(IV, m)}$

if $IV \in used$ ***then***
  return $\perp$
end if
$used \leftarrow used \cup \{IV\}$
if $b = 0$ ***then***
  $m \xleftarrow{\$} \{0,1\}^{|m|}$
end if
$c \leftarrow \mathsf{SE.E}(IV, K, m)$
return c

ROR-CPNA model differs from ROR-CPA in definition of Encrypt oracle. First in ROR-CPNA model Encrypt oracle takes as input one additional value $IV \in \mathtt{IV}$. It states as initialization vector for the correct work of encryption scheme. The ROR-CPNA experiment also uses the set *used* to check if this IV value was queried yet to avoid the trivial attack. If IV from new query was queried early then oracle Encrypt returns error($\perp$).

Often the analysis results are received in stronger model IND-CPNA. In this model the adversary has to distinguish the obtained ciphertexts from the «garbage», having the capability to adaptively choose plaintexts and nonces (in a unique manner). This model can't be applied to Record protocol analysis because header is simply distinguishable from random value. But this model is applyable to the cipher modes such as CTR-ACPKM

Definition 15. *Let* SE = {SE.K, SE.E, SE.D} *be a symmetric encryption scheme and let A be an adversary. The advantage of A for the scheme* SE *in the* IND-CPNA *model (*IND-CPNA-advantage*) is defined as*

$$\mathsf{Adv}_{\mathsf{SE}}^{\text{IND-CPNA}}(A) = \\ = \Pr\left[\mathbf{Exp}_{\mathsf{SE}}^{\text{IND-CPNA}-1}(A) \to 1\right] - \Pr\left[\mathbf{Exp}_{\mathsf{SE}}^{\text{IND-CPNA}-0}(A) \to 1\right],$$

where the experiments $\mathbf{Exp}_{\mathsf{SE}}^{\text{IND-CPNA}-b}(A),\ b \in \{0,1\}$ *is defined as follows*

$\underline{\mathbf{Exp}_{\mathsf{SE}}^{\text{IND-CPNA}-b}(A)}$

$K \xleftarrow{\$} \mathcal{K}$
$used \leftarrow \varnothing$
$b' \leftarrow A^{\mathsf{Encrypt}^b(\cdot)}$
return b'

$\underline{\mathsf{Oracle\ Encrypt}^b(IV, m)}$

if $IV \in used$ ***then***
  return $\perp$
end if
$used \leftarrow used \cup \{IV\}$
$c \leftarrow \mathsf{SE.E}(IV, K, m)$
if $b = 0$ ***then***
  $c' \xleftarrow{\$} \{0,1\}^{|c|}$
  return c'
end if
return c

The following inequality holds:

Statement 1.

$$\mathsf{InSec}_{\mathsf{SE}}^{\text{ROR-CPNA}}(t, q, l) \leqslant \mathsf{InSec}_{\mathsf{SE}}^{\text{IND-CPNA}}(t, q, l).$$

Definition 16. *Let* MA *be a message authentication scheme and let* A *be an adversary. Then the advantage of* A *in* PRF *model is defined as:*

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

where experiments $\mathbf{Exp}_{\mathsf{MA}}^{\text{PRF}-b}(A),\ b \in \{0,1\}$*, is defined as follows:*

$\underline{\mathbf{Exp}_{\mathsf{MA}}^{\text{PRF}-1}(A)}$

$K \xleftarrow{\$} \mathsf{MA.K}()$
$b' \xleftarrow{\$} A^{\mathsf{Tag}^1}$
return b'

$\underline{\mathsf{Oracle\ Tag}^1(m)}$

return $\mathsf{MA.TAG}(K, m)$

$\underline{\mathbf{Exp}_{\mathsf{MA}}^{\text{PRF}-0}(A)}$

$Rnd \leftarrow \varnothing$
$b' \xleftarrow{\$} A^{\mathsf{Tag}^0}$
return b'

$\underline{\mathsf{Oracle\ Tag}^0(m)}$

if $\nexists\ \tau' \in \mathcal{T} : (m, \tau') \in Rnd$
then
  $\tau \xleftarrow{\mathcal{U}} \mathcal{T}$
  $Rnd \leftarrow Rnd \cup \{(m, \tau)\}$
else
  $\tau \leftarrow \tau'$
end if
return τ

Definition 17. *Let* G *be a pseudorandom generator and let* A *be an adversary. Then the advantage of* A *in* PRG *model is defined as:*

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

where the experiments $\mathbf{Exp}_{\mathsf{G}}^{\mathrm{PRG}-1}(A)$ *and* $\mathbf{Exp}_{\mathsf{G}}^{\mathrm{PRG}-0}(A)$ *are defined as follows:*

$\underline{\mathbf{Exp}_{\mathsf{G}}^{\mathrm{PRG}-1}(A)}$
$St_0 \xleftarrow{\$} \mathsf{G.K}()$
$Out \leftarrow \varepsilon$
$N \xleftarrow{\$} A$
for i ***do***0N-1
  $(Out_i, St_{i+1}) \leftarrow \mathsf{G.N}(St_i)$
end for
$Out \leftarrow (Out_0, \ldots, Out_{N-1})$
$b \xleftarrow{\$} A(Out)$
return b

$\underline{\mathbf{Exp}_{\mathsf{G}}^{\mathrm{PRG}-0}(A)}$
$N \xleftarrow{\$} A$
$Out \xleftarrow{\mathcal{U}} \mathcal{B}^N$
$b \xleftarrow{\$} A(Out)$
return b

B Security analysis of AEAD schemes

B.1 Security analysis of MtE-AD scheme

Consider the advantage of the adversary A in IND-sfCCSA model for sfAEAD scheme.

$$\mathsf{Adv}_{\mathsf{sfAEAD}}^{\text{IND-sfCCSA}}(A) = \\ = \Pr\left[\mathbf{Exp}_{\mathsf{sfAEAD}}^{\text{IND-sfCCSA}-1}(A) \to 1\right] - \Pr\left[\mathbf{Exp}_{\mathsf{sfAEAD}}^{\text{IND-sfCCSA}-0}(A) \to 1\right]$$

Construction of adversary C**.** Construct adversary C in PRF model for MA scheme that uses A as a black box next way

$\underline{C^{\mathsf{Tag}^b}}$

$\hat{b} \xleftarrow{\mathcal{U}} \{0,1\}$
$K_{\mathsf{SE}} \xleftarrow{\$} \mathsf{SE.K}()$
$u \leftarrow 0;\ v \leftarrow 0$
$sent \leftarrow \varnothing$
$st \leftarrow A$
$(st_E, st_D) \leftarrow \mathsf{sfAEAD.Init}(st)$
$b' \leftarrow A^{SimEnc,\ SimDec}$
return $b' = \hat{b}$

$\underline{SimEnc(ad, m)}$

$\widehat{m} \leftarrow \mathsf{encode}_{\mathsf{MA}}(ad, m, st_E)$
$t \leftarrow \mathsf{Tag}^b(\widehat{m})$
$IV_E \leftarrow \mathsf{StateToIV}(st_E)$
$\widetilde{m} \leftarrow \mathsf{encode}_{\mathsf{SE}}(m, t)$
$c \leftarrow \mathsf{SE.E}(K_{\mathsf{SE}}, IV_E, \widetilde{m})$
$sent \leftarrow sent \cup (ad, c, u)$
$st_E \leftarrow \mathsf{sfAEAD.Upd}(st_E)$
$u \leftarrow u + 1$
return c

$\underline{SimDec(ad, c)}$

$IV_D \leftarrow \mathsf{StateToIV}(st_D)$
$\widetilde{m} \leftarrow \mathsf{SE.D}(K_{\mathsf{SE}}, c, IV_D)$
$(m, t) \leftarrow \mathsf{decode}_{\mathsf{SE}}(\widetilde{m})$
$\widehat{m} \leftarrow \mathsf{encode}_{\mathsf{MA}}(ad, m, st_D)$
if $\mathsf{Tag}^b(\widehat{m}) = t$ **then**
  if $((ad, c, v) \in sent)$ **or** $(\hat{b} = 0)$ **then**
    $m \leftarrow \perp$
  end if
  $st_D \leftarrow \mathsf{sfAEAD.Upd}(st_D)$
  $v \leftarrow v + 1$
else
  $m \leftarrow \perp$
end if
return m

Adversary C first generate bit $\hat{b}$ and key K_{SE} at random and then simulates oracle $\mathsf{Encrypt}$ and $\mathsf{Decrypt}$ using $SimEnc$ and $SimDec$ to answer the A queries.

Denote $\mathbf{Exp}^{\text{IND-sfCCSA}-b_1,b_2}_{\mathsf{sfAEAD}}(A)$ the experiment where the adversary interacts with encryption oracle $\mathsf{Encrypt}^{b_1}$ and decryption oracle $\mathsf{Decrypt}^{b_2}$ ($\mathbf{Exp}^{\text{IND-sfCCSA}-1,1}_{\mathsf{sfAEAD}}(A) = \mathbf{Exp}^{\text{IND-sfCCSA}-1}_{\mathsf{sfAEAD}}(A)$, $\mathbf{Exp}^{\text{IND-sfCCSA}-0,0}_{\mathsf{sfAEAD}}(A) = \mathbf{Exp}^{\text{IND-sfCCSA}-0}_{\mathsf{sfAEAD}}(A)$).

Note that if adversary C interacts with $\mathbf{Exp}^{\text{PRF}-1}_{\mathsf{MA}}$ then C simulates to A experiment $\mathbf{Exp}^{\text{IND-sfCCSA}-1,1}_{\mathsf{sfAEAD}}$ if $\hat{b} = 1$ and experiment $\mathbf{Exp}^{\text{IND-sfCCSA}-1,0}_{\mathsf{sfAEAD}}$ otherwise. If C interacts with $\mathbf{Exp}^{\text{PRF}-0}_{\mathsf{MA}}$ then he simulates to A experiment $\mathbf{Exp}^{\text{IND-sfCCSA}-1,1}_{\mathsf{sfAEAD}'}$ if $\hat{b} = 1$ and experiment $\mathbf{Exp}^{\text{IND-sfCCSA}-1,0}_{\mathsf{sfAEAD}'}$ otherwise. There sfAEAD' is sfAEAD scheme with MA changed to random oracle. Random oracle does not use the key $\mathcal{K}_{\mathsf{MA}}$, but for every new value $\widehat{m} \in \mathcal{M}_{\mathsf{MA}}$ returns value $t \xleftarrow{\mathcal{U}} \mathcal{T}$ at random.

The advantage of the adversary C in PRF model is

$$\mathsf{Adv}^{\mathrm{PRF}}_{\mathsf{MA}}(C) = \Pr\left[\mathbf{Exp}^{\mathrm{PRF}-1}_{\mathsf{MA}}(C) \to 1\right] - \Pr\left[\mathbf{Exp}^{\mathrm{PRF}-0}_{\mathsf{MA}}(C) \to 1\right] =$$
$$= \frac{1}{2} \cdot \left(\Pr\left[\mathbf{Exp}^{\mathrm{PRF}-1}_{\mathsf{MA}}(C) \to 1 \mid \hat{b} = 1\right] + \Pr\left[\mathbf{Exp}^{\mathrm{PRF}-1}_{\mathsf{MA}}(C) \to 1 \mid \hat{b} = 0\right]\right) -$$
$$- \frac{1}{2} \cdot \left(\Pr\left[\mathbf{Exp}^{\mathrm{PRF}-0}_{\mathsf{MA}}(C) \to 1 \mid \hat{b} = 1\right] - \Pr\left[\mathbf{Exp}^{\mathrm{PRF}-0}_{\mathsf{MA}}(C) \to 1 \mid \hat{b} = 0\right]\right) =$$
$$= \frac{1}{2} \cdot \left(\Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA}-1,1}_{\mathsf{sfAEAD}}(A) \to 1\right] + \Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA}-1,0}_{\mathsf{sfAEAD}}(A) = 0\right]\right) -$$
$$- \frac{1}{2} \cdot \left(\Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA}-1,1}_{\mathsf{sfAEAD}'}(A) \to 1\right] + \Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA}-1,0}_{\mathsf{sfAEAD}'}(A) = 0\right]\right) =$$
$$= \frac{1}{2} \cdot \underbrace{\left(\Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA}-1,1}_{\mathsf{sfAEAD}}(A) \to 1\right] - \Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA}-1,0}_{\mathsf{sfAEAD}}(A) \to 1\right]\right)}_{\alpha} -$$
$$- \frac{1}{2} \cdot \underbrace{\left(\Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA}-1,1}_{\mathsf{sfAEAD}'}(A) \to 1\right] - \Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA}-1,0}_{\mathsf{sfAEAD}'}(A) \to 1\right]\right)}_{\beta}.$$

Then $\alpha = 2 \cdot \mathsf{Adv}^{\mathrm{PRF}}_{\mathsf{MA}}(C) + \beta$. Estimate the α value. Note that

$$\mathsf{Adv}^{\text{IND-sfCCSA}}_{\mathsf{sfAEAD}}(A) =$$
$$= \Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA}-1,1}_{\mathsf{sfAEAD}}(A) \to 1\right] - \Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA}-0,0}_{\mathsf{sfAEAD}}(A) \to 1\right] =$$
$$= \Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA}-1,1}_{\mathsf{sfAEAD}}(A) \to 1\right] - \Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA}-1,0}_{\mathsf{sfAEAD}}(A) \to 1\right] +$$
$$+ \Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA}-1,0}_{\mathsf{sfAEAD}}(A) \to 1\right] - \Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA}-0,0}_{\mathsf{sfAEAD}}(A) \to 1\right] =$$
$$= \alpha + \underbrace{\Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA}-1,0}_{\mathsf{sfAEAD}}(A) \to 1\right] - \Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA}-0,0}_{\mathsf{sfAEAD}}(A) \to 1\right]}_{\gamma}.$$

Then, $\alpha = \mathsf{Adv}^{\text{IND-sfCCSA}}_{\mathsf{sfAEAD}}(A) - \gamma$, and estimate can be rewritten as

$$\mathsf{Adv}^{\text{IND-sfCCSA}}_{\mathsf{sfAEAD}}(A) = 2 \cdot \mathsf{Adv}^{\mathrm{PRF}}_{\mathsf{MA}}(C) + \beta + \gamma.$$

Estimate the value $\beta = \Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA}-1,1}_{\mathsf{sfAEAD}'}(A) \to 1\right] - \Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA}-1,0}_{\mathsf{sfAEAD}'}(A) \to 1\right]$.

Introduce the following modifications of the experiment: $\mathbf{Exp}^{\text{IND-sfCCSA}_j-1,1}_{\mathsf{sfAEAD}'}$,

$j = 0, 1, \ldots, q_D$. The encryption oracle $\mathsf{Encrypt}^1$ does not change and the decryption oracle $\mathsf{Decrypt}^b,\ b \in \{0,1\}$ is defined as follows:

$\underline{\mathbf{Exp}^{\text{IND-sfCCSA}_j-1,1}_{\mathsf{sfAEAD}'}(A)}$

$K \xleftarrow{\$} \mathsf{sfAEAD.K}()$
$u \leftarrow 0,\ v \leftarrow 0,\ r \leftarrow 0$
$sent \leftarrow \varnothing$
$st \leftarrow A$
$(st_E,\ st_D) \leftarrow \mathsf{sfAEAD.Init}(st)$
$b' \leftarrow A^{\mathsf{Encrypt}^1, \mathsf{Decrypt}^b}$
return b'

$\underline{\text{Oracle } \mathsf{Decrypt}^1(ad, c)}$

$m \leftarrow \mathsf{sfAEAD.D}(K, ad, c, st_D)$
if $((ad, c, v) \notin sent)$ **then**
  if $(r < j)$ **then**
    $m \leftarrow \perp$
  end if
  $r \leftarrow r + 1$
end if
if $(m \neq \perp)$ **then**
  if $((ad, c, v) \in sent)$ **then**
    $m \leftarrow \perp$
  end if
  $st_D \leftarrow \mathsf{sfAEAD.Upd}(st_D)$
  $v \leftarrow v + 1$
end if
return m

In this modification the decryption oracle $\mathsf{Decrypt}^1$ returns error while answering to first j testing queries regardless of their correctness.

Note that $\mathbf{Exp}^{\text{IND-sfCCSA}_0-1,1}_{\mathsf{sfAEAD}'}$ is equal to $\mathbf{Exp}^{\text{IND-sfCCSA}-1,1}_{\mathsf{sfAEAD}'}$ and $\mathbf{Exp}^{\text{IND-sfCCSA}_{q_D}-1,1}_{\mathsf{sfAEAD}'}$ is equal to $\mathbf{Exp}^{\text{IND-sfCCSA}-1,0}_{\mathsf{sfAEAD}'}$.

So the following equation holds:

$$\begin{aligned}
\beta &= \Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA}-1,1}_{\mathsf{sfAEAD}'}(A) \to 1\right] - \Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA}-1,0}_{\mathsf{sfAEAD}'}(A) \to 1\right] = \\
&= \Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA}-1}_{\mathsf{sfAEAD}'}(A) \to 1\right] - \Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA}-1,0}_{\mathsf{sfAEAD}'}(A) \to 1\right] = \\
&= \Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA}_0-1,1}_{\mathsf{sfAEAD}'}(A) \to 1\right] - \Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA}_{q_D}-1,1}_{\mathsf{sfAEAD}'}(A) \to 1\right] = \\
&= \sum_{j=0}^{q_D-1} \underbrace{\left(\Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA}_j-1,1}_{\mathsf{sfAEAD}'}(A) \to 1\right] - \Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA}_{j+1}-1,1}_{\mathsf{sfAEAD}'}(A) \to 1\right]\right)}_{\mu_j}.
\end{aligned}$$

Estimate μ_j for some $j \in \{0, \ldots, q_D - 1\}$.

Let ω_j be an event such that $j+1$-th adversary query is a correct testing query (adversary forged). Note that $\mathbf{Exp}_{\mathsf{sfAEAD}'}^{\text{IND-sfCCSA}_{j+1}-1,1}$ returns $\perp$ answering to this query and $\mathbf{Exp}_{\mathsf{sfAEAD}'}^{\text{IND-sfCCSA}_{j}-1,1}$ returns some value $m \neq \perp$. Note that ω_j probability is determined by randomness source of the adversary A, random key K_{SE} and the choosing mac values at random. Then the value of $j+1$-th query depends on randomness source of the adversary A anf the previous answers from oracles.

But when the $j+1$-th testing query is formed both $\mathbf{Exp}_{\mathsf{sfAEAD}'}^{\text{IND-sfCCSA}_{j}-1,1}$ and $\mathbf{Exp}_{\mathsf{sfAEAD}'}^{\text{IND-sfCCSA}_{j+1}-1,0}$ have returned identical answers so the distributions are identical too. It is true because the encryption oracle $\mathsf{Encrypt}^1$ is equal and $\mathsf{Decrypt}^1$ decryption oracle returned only error value $\perp$. Then the distributions those determine the probability of ω_j event in experiments $\mathbf{Exp}_{\mathsf{sfAEAD}'}^{\text{IND-sfCCSA}_{j}-1,1}(A)$ and $\mathbf{Exp}_{\mathsf{sfAEAD}'}^{\text{IND-sfCCSA}_{j}-1,1}(A)$ are indistinguishable for every adversary A. Moreover if the event ω_j does not occured for any j then the distributions on results of experiments $\mathbf{Exp}_{\mathsf{sfAEAD}'}^{\text{IND-sfCCSA}-1}(A)$ and $\mathbf{Exp}_{\mathsf{sfAEAD}'}^{\text{IND-sfCCSA}-1,0}(A)$ are indistinguishable too.

So for any $j \in \{0, \ldots, q_D - 1\}$:

$$\mu_j = \Pr\left[\mathbf{Exp}_{\mathsf{sfAEAD}'}^{\text{IND-sfCCSA}_{j}-1,1}(A) \to 1\right] - \Pr\left[\mathbf{Exp}_{\mathsf{sfAEAD}'}^{\text{IND-sfCCSA}_{j+1}-1,0}(A) \to 1\right] =$$
$$= \Pr[\omega_j] \cdot \underbrace{\left(\Pr\left[\mathbf{Exp}_{\mathsf{sfAEAD}'}^{\text{IND-sfCCSA}_{j}-1,1}(A) \to 1 \mid \omega_j\right] - \Pr\left[\mathbf{Exp}_{\mathsf{sfAEAD}'}^{\text{IND-sfCCSA}_{j+1}-1,1}(A) \to 1 \mid \omega_j\right]\right)}_{\leqslant 1} +$$
$$+ \Pr[\overline{\omega_j}] \cdot \underbrace{\Pr\left[\mathbf{Exp}_{\mathsf{sfAEAD}'}^{\text{IND-sfCCSA}_{j}-1,1}(A) \to 1 \mid \overline{\omega_j}\right] - \Pr\left[\mathbf{Exp}_{\mathsf{sfAEAD}'}^{\text{IND-sfCCSA}_{j+1}-1,0}(A) \to 1 \mid \overline{\omega_j}\right]}_{=0} \leqslant$$
$$\leqslant \Pr[\omega].$$

Estimate the probability $\Pr[\omega_j]$.

Denote by (ad', c') the $j+1$-th testing request to the decryption oracle for some counter value v' and the internal state st'_D. Let $\widehat{m}'$ be such that $\widehat{m}' = \mathsf{encode}_{\mathsf{MA}}(ad', m', st'_D)$, $(m', t') = \mathsf{decode}_{\mathsf{SE}}(\mathsf{SE.D}(K_{\mathsf{SE}}, IV_D, c'))$.

Since the request (ad', c') is testing, i.e. $(ad', c', v') \notin sent$, then one of the following conditions is met:

1. at the time of processing the request (ad', c'), the counter value u is not greater than the value of v' (the record with the number v' has not yet been sent to the channel by an honest sender).

2. $ad \neq ad' \ \vee \ c \neq c'$, where (ad, c) is a query to the oracle $\mathsf{Encrypt}^1$ with the value of the counter $u = v'$ and the internal state $st_E = st'_D$ (trivial query).

Let us show that the $j+1$-th testing request to the decryption oracle causes a new request to the random oracle.

The first case. By the condition of the theorem, the number of queries to the oracle $\mathsf{Encrypt}^1$ is not greater than $\alpha_{\min}$, and the trivial queries to the oracle $\mathsf{Decrypt}^1$ are not greater than $\alpha_{\min} - 1$. Also, due to the definition of the oracle $\mathsf{Decrypt}^1$, the preceding testing queries do not increase the value of the v counter, therefore $v' < \ alpha_{\min}$ (counter from zero). Therefore, due to the properties of the Next function, when processing all queries to the $\mathsf{Encrypt}^1$ oracle with $u \neq v'$, the states $st^* \ neqst_D$ were used. Thus, the inequality $\mathsf{encode}_{\mathsf{MA}}(ad^*, m^*, st^*) \neq \widehat{m}'$ holds for any ad^*, m^* due to the $\mathsf{encode}_{\mathsf{MA}}$ is collision free function with according with the function Next.

Recall that the oracle $\mathsf{Decrypt}^1$ by definition does not handle all previous requests and returns $\perp$ in response.

Thus, the value of $\widehat{m}'$ has not previously arrived at the input of a random oracle.

The second case. Since the reasoning above for the first case is true for the second case, it suffices to show that $\widehat{m}' \neq \widehat{m}$, where $\widehat{m}$ is value corresponding to the trivial query (ad, c) such that $\widehat{m} = \mathsf{encode}_{\mathsf{MA}}(ad, m, st'_D)$, $(m, t) = \mathsf{decode}_{\mathsf{SE}}(\mathsf{SE.D}(K_{\mathsf{SE}}, IV_D, c))$.

Note that

- if $ad \neq ad'$, then $\widehat{m}' \neq \widehat{m}$, because $\mathsf{encode}_{\mathsf{MA}}$ is collision free function with according with the function Next.
- if $c \neq c'$ and $ad = ad'$, then at least one of the conditions $m \neq m'$ or $t \neq t'$ is met because SE is CRD-scheme and $\mathsf{decode}_{\mathsf{SE}}$ is injective. Then
  - if $m \neq m'$, then $\widehat{m} \neq \widehat{m}'$, because $\mathsf{encode}_{\mathsf{MA}}$ is collision free function with according with the function Next.
  - if $(m = m')$ and $(t \neq t')$, then $\widehat{m} = \widehat{m}'$ and $t = t'$, because random oracle returns equal values for equal inputs.

Thus, in the second case, the value of $\widehat{m}'$ has not previously arrived at the input of a random oracle.

Therefore, the required probability is estimated from above by the probability of guessing the value $t \xleftarrow{\mathcal{U}} \mathcal{T}$, i.e.

$$\Pr[\omega_j] \leqslant \frac{1}{2^\tau},\ \forall j \in \{0, \ldots, q_D - 1\}$$

Thus, the bound takes the form

$$\mathsf{Adv}^{\text{IND-sfCCSA}}_{\mathsf{sfAEAD}}(A) = 2 \cdot \mathsf{Adv}^{\text{PRF}}_{\mathsf{MA}}(C) + \frac{q_D}{2^\tau} + \gamma.$$

Construction of adversary B. We now estimate the value of γ by constructing an adversary B for the SE scheme in the ROR-CPNA model, which uses A as the black box.

$\underline{B^{\mathsf{Encrypt}^b}}$
$K_{\mathsf{MA}} \xleftarrow{\$} \mathsf{MA.K}()$
$st \leftarrow A$
$(st_E, st_D) \leftarrow \mathsf{sfAEAD.Init}(st)$
$b' \leftarrow A^{SimEnc,\ SimDec}$
return b'

$\underline{SimEnc(ad, m)}$
$\widehat{m} \leftarrow \mathsf{encode}_{\mathsf{MA}}(ad, m, st_E)$
$t \leftarrow \mathsf{MA.TAG}(K_{\mathsf{MA}}, \widehat{m})$
$\widetilde{m} \leftarrow \mathsf{encode}_{\mathsf{SE}}(m, t)$
$IV_E \leftarrow \mathsf{StateToIV}(st_E)$
$c \leftarrow \mathsf{Encrypt}^b(IV_E, \widetilde{m})$
$st_E \leftarrow \mathsf{sfAEAD.Upd}(st_E)$
return c

$\underline{SimDec(ad, c)}$
return $\perp$

By definition $q_E \leqslant \alpha_{\min}$. Therefore, by virtue of the injectivity of the function $\mathsf{StateToIV}$ with according to Next condition holds $IV_E \notin used$.

Thus, with $b = 0$ the adversary B simulates the conditions of the experiment $\mathbf{Exp}^{\text{IND-sfCCSA}-0,0}_{\mathsf{sfAEAD}}$, and with $b = 1$ simulates the conditions of the experiment $\mathbf{Exp}^{\text{IND-sfCCSA}-1,0}_{\mathsf{sfAEAD}}$. Therefore,

$$\gamma = \Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA}-1,0}_{\mathsf{sfAEAD}}(A) \to 1\right] - \Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA}-0,0}_{\mathsf{sfAEAD}}(A) \to 1\right] =$$
$$= \Pr\left[\mathbf{Exp}^{\text{ROR-CPNA}-1}_{\mathsf{SE}}(B) \to 1\right] - \Pr\left[\mathbf{Exp}^{\text{ROR-CPNA}-0}_{\mathsf{SE}}(B) \to 1\right] =$$
$$= \mathsf{Adv}^{\text{ROR-CPNA}}_{\mathsf{SE}}(B)$$

Substituting the resulting ratio in the estimate, we get

$$\mathsf{Adv}^{\text{IND-sfCCSA}}_{\mathsf{sfAEAD}}(A) \quad = \quad 2\mathsf{Adv}^{\text{PRF}}_{\mathsf{MA}}(C) \quad + \quad \mathsf{Adv}^{\text{ROR-CPNA}}_{\mathsf{SE}}(B) \quad + \quad \frac{q_D}{2^n}.$$

B.2 Security analysis of MtE-AD scheme with generator

We construct the adversary D in the PRG model for the G generator, which uses A as the black box. The adversary D acts as follows. Receiving as input some sequence from $N = \lceil q_E/h \rceil$ of blocks $K_0, K_1, \ldots, K_{N-1}$, he chooses the random bit $b \xleftarrow{\$} \{0,1\}$ and models for A experimental conditions $\mathbf{Exp}^{\text{IND-sfCCSA}-b}_{(\mathsf{sfAEAD},\hat{\mathsf{G}})_h}(A)$, using these blocks as appropriate keys. Note that in the case of the adversary D in the model PRG $-$ 1, $\hat{\mathsf{G}}$ is the generator G. In the case of the adversary D in the model PRG $-$ 0 , $\hat{\mathsf{G}} = \mathsf{G}'$ is the generator that produces the keys, choosing them equiprobably from the set $\mathcal{K}$ independently of friend After the completion of the experiment, the adversary D outputs 1, if the withdrawal of the adversary A coincided with the bit b, and 0, in the opposite case.

Consider the advantage of the adversary D in the PRG model:

$$\begin{aligned}
&\mathsf{Adv}^{\text{PRG}}_{\mathsf{G}}(D) = \Pr\left[\mathbf{Exp}^{\text{PRG}-1}_{\mathsf{G}}(D) \to 1\right] - \Pr\left[\mathbf{Exp}^{\text{PRG}-0}_{\mathsf{G}}(D) \to 1\right] = \\
&= \Big(\Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA}-1}_{(\mathsf{sfAEAD},\mathsf{G})_h}(A) \to 1 \ \cap \ b = 1\right] + \Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA}-0}_{(\mathsf{sfAEAD},\mathsf{G})_h}(A) \to 0 \ \cap \ b = 0\right]\Big) - \\
&\quad - \Big(\Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA}-1}_{(\mathsf{sfAEAD},\mathsf{G}')_h}(A) \to 1 \ \cap \ b = 1\right] + \Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA}-0}_{(\mathsf{sfAEAD},\mathsf{G}')_h}(A) \to 0 \ \cap \ b = 0\right]\Big)
\end{aligned}$$

The bit b is chosen randomly, then the previous expression can be written as follows:

$$\begin{aligned}
&\mathsf{Adv}^{\text{PRG}}_{\mathsf{G}}(D) = \Pr\left[\mathbf{Exp}^{\text{PRG}-1}_{\mathsf{G}}(D) \to 1\right] - \Pr\left[\mathbf{Exp}^{\text{PRG}-0}_{\mathsf{G}}(D) \to 1\right] = \\
&\quad = \Big(\frac{1}{2} \cdot \Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA}-1}_{(\mathsf{sfAEAD},\mathsf{G})_h}(A) \to 1\right] + \frac{1}{2} \cdot \Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA}-0}_{(\mathsf{sfAEAD},\mathsf{G})_h}(A) \to 0\right]\Big) - \\
&\quad - \Big(\frac{1}{2} \cdot \Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA}-1}_{(\mathsf{sfAEAD},\mathsf{G}')_h}(A) \to 1\right] + \frac{1}{2} \cdot \Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA_}0}_{(\mathsf{sfAEAD},\mathsf{G}')_h}(A) \to 0\right]\Big) = \\
&\quad = \frac{1}{2}\Big(\Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA}-1}_{(\mathsf{sfAEAD},\mathsf{G})_h}(A) \to 1\right] + \Big(1 - \Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA}-0}_{(\mathsf{sfAEAD},\mathsf{G})_h}(A) \to 1\right]\Big)\Big) - \\
&\quad - \frac{1}{2}\Big(\Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA}_{h,N}-1}_{(\mathsf{sfAEAD},\mathsf{G}')_h}(A) \to 1\right] + \Big(1 - \Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA}-0}_{(\mathsf{sfAEAD},\mathsf{G}')_h}(A) \to 1\right]\Big)\Big) = \\
&\quad = \frac{1}{2}\Big(\mathsf{Adv}^{\text{IND-sfCCSA}}_{(\mathsf{sfAEAD},\mathsf{G})_h}(A) - \mathsf{Adv}^{\text{IND-sfCCSA}}_{(\mathsf{sfAEAD},\mathsf{G}')_h}(A)\Big)
\end{aligned}$$

Thus,

$$\mathsf{Adv}^{\text{IND-sfCCSA}}_{(\mathsf{sfAEAD},\mathsf{G})}(A) \leqslant 2 \cdot \mathsf{Adv}^{\text{PRG}_N}_{\mathsf{G}}(D) + \mathsf{Adv}^{\text{IND-sfCCSA}}_{(\mathsf{sfAEAD},\mathsf{G}')}(A).$$

Estimate the advantage $\mathsf{Adv}^{\text{IND-sfCCSA}}_{(\mathsf{sfAEAD},\mathsf{G}')_h}(A)$. To do this, we describe the following series of experiments $Hybrid_j(A)$, where $j \in \{0, 1 \ldots N\}$.

$\underline{Hybrid_j(A)}$

$K_0, K_1, \ldots, K_{N-1} \xleftarrow{\mathcal{U}} \{0,1\}^k$
$u \leftarrow 0,\ v \leftarrow 0$
$sent \leftarrow \varnothing$
$st \leftarrow A$
$(st_E,\ st_D) \leftarrow \mathsf{sfAEAD.Init}(st)$
$b' \leftarrow A^{\mathsf{HybridEnc},\ \mathsf{HybridDec}}$
return b'

$\underline{\mathsf{Oracle\ HybridEnc}(ad, m)}$

$i \leftarrow \lfloor u/h \rfloor$
if $u < j \cdot h$ **then**
  $m \xleftarrow{\$} \{0,1\}^{|m|}$
end if
$c \leftarrow \mathsf{sfAEAD.E}(K_i, ad, m, st_E)$
$sent \leftarrow sent \cup (ad, c, u)$
$st_E \leftarrow \mathsf{sfAEAD.Upd}(st_E)$
$u \leftarrow u + 1$
return c

$\underline{\mathsf{Oracle\ HybridDec}(ad, c)}$

$i \leftarrow \lfloor v/h \rfloor$
$m \leftarrow \mathsf{sfAEAD.D}(K_i, ad, c, st_D)$
if $(m \neq \perp)$ **then**
  if $(ad, c, v) \in sent$ **or** $v < j \cdot h$ **then**
    $m \leftarrow \perp$
  end if
  $st_D \leftarrow \mathsf{sfAEAD.Upd}(st_D)$
  $v \leftarrow v + 1$
end if
return m

In these experiments, oracles $\mathsf{Encrypt}^0$ and $\mathsf{Decrypt}^0$ are modeled on the first $j \cdot h$ requests for the adversary A, and on the rest oracles $\mathsf{Encrypt}^1$ and $\mathsf{Decrypt}^1$. We introduce the following notation: $P_j = \Pr[Hybrid_j(A) = 1]$. Note that in this case

$$\gamma = \Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA}_{h,N-1}}_{(\mathsf{sfAEAD},\mathsf{G}')}(A) \to 1\right] - \Pr\left[\mathbf{Exp}^{\text{IND-sfCCSA}_{l,N-0}}_{(\mathsf{sfAEAD},\mathsf{G}')}(A) \to 1\right] =$$
$$= \Pr[Hybrid_N(A) \to 1] - \Pr[Hybrid_0(A) \to 1] = P_N - P_0.$$

To estimate the value $P_N - P_0$, let construct an adversary B in the IND-sfCCSA model for the sfAEAD scheme.

$\underline{B^{\mathsf{Encrypt}^{\mathsf{b}},\mathsf{Decrypt}^{\mathsf{b}}}}$
$j \xleftarrow{\mathcal{U}} \{1, \ldots N\}$
$K_0, \ldots, K_{j-1}, K_{j+1}, \ldots \xleftarrow{\$} \{0,1\}^k$
$u \leftarrow 0,\ v \leftarrow 0$
$sent \leftarrow \varnothing$
$st \leftarrow A$
$(st_E,\ st_D) \leftarrow \mathsf{sfAEAD.Init}(st)$
$b' \leftarrow A^{SimEnc,\ SimDec}$
return b'

$\underline{SimEnc(ad, m)}$
$i \leftarrow \lfloor u/h \rfloor$
if $u < (j-1) \cdot h$ **then**
  $m \xleftarrow{\$} \{0,1\}^{|m|}$
  $c \leftarrow \mathsf{sfAEAD.E}(K_i, ad, m, st_E)$
end if

if $(j-1) \cdot h < u < j \cdot h$ **then**
  if $v < u$ **and** $u = (j-1) \cdot h$ **then**
    send st_E
  end if
  $c \leftarrow \mathsf{Encrypt}^{\mathsf{b}}(ad, m)$
end if

if $u \geqslant j \cdot h$ **then**
  $c \leftarrow \mathsf{sfAEAD.E}(K_i, ad, m, st_E)$
end if

$sent \leftarrow sent \cup (ad, c, u)$
$st_E \leftarrow \mathsf{sfAEAD.Upd}(st_E)$
$u \leftarrow u + 1$
return c

$\underline{SimDec(ad, c)}$
$i \leftarrow \lfloor v/h \rfloor$
if $v < (j-1) \cdot h$ **then**
  $m \leftarrow \mathsf{sfAEAD.D}(K_i, ad, c, st_D)$
end if

if $(j-1) \cdot h < v < j \cdot h$ **then**
  if $u < v$ **and** $v = (j-1) \cdot h$ **then**
    send st_D
  end if
  $m \leftarrow \mathsf{Decrypt}^{\mathsf{b}}(ad, c)$
end if

if $v \geqslant j \cdot h$ **then**
  $m \leftarrow \mathsf{sfAEAD.}Dec(K_i, ad, c, st_D)$
end if

if $(m \neq \perp)$ **then**
  if $(ad, c, v) \in sent$ **or** $v < (j-1) \cdot h$ **then**
    $m \leftarrow \perp$
  end if
  $st_D \leftarrow \mathsf{sfAEAD.Upd}(st_D)$
  $v \leftarrow v + 1$
end if
return m

The adversary B chooses $j \xleftarrow{\mathcal{U}} \{1, \ldots N\}$, queries the A initialization data, and then models for the adversary A the work of its oracles using the $SimEnc$ and $SimDec$ functions. The functions $SimEnc$ and $SimDec$ are designed so that the first $(j-1) \cdot h$ queries they simulate oracles $\mathsf{Encrypt}^0$ and $\mathsf{Decrypt}^0$ of the adversary A respectively. On the queries from $(j-1) \cdot h + 1$ to $j \cdot h$ the

functions use the own oracles of the adversary B to respond to the requests of the adversary A. On the remaining requests, the functions $SimEnc$ and $SimDec$ simulate oracles $\mathsf{Encrypt}^1$ and $\mathsf{Decrypt}^1$ of the adversary A. After completing the experiment, the adversary B outputs the same bit as A.

Note that if the adversary B has access to the oracles $\mathsf{Encrypt}^0$ and $\mathsf{Decrypt}^0$, then the following equality holds:

$$\Pr\left[\mathbf{Exp}_{\mathsf{sfAEAD}}^{\text{IND-sfCCSA}-0}(B) \to 1\right] = \sum_{i=1}^{N} \frac{1}{N} \cdot P_i.$$

And if the adversary B has access to the oracles $\mathsf{Encrypt}^1$ and $\mathsf{Decrypt}^1$,

$$\Pr\left[\mathbf{Exp}_{\mathsf{sfAEAD}}^{\text{IND-sfCCSA}-1}(B) \to 1\right] = \sum_{i=1}^{N} \frac{1}{N} \cdot P_{i-1}.$$

Consider the advantage of the adversary B in the IND-sfCCSA model

$$\begin{aligned}\mathsf{Adv}_{\mathsf{sfAEAD}}^{\text{IND-sfCCSA}}(B) = \\ = \Pr\left[\mathbf{Exp}_{\mathsf{sfAEAD}}^{\text{IND-sfCCSA}-1}(B) \to 1\right] - \Pr\left[\mathbf{Exp}_{\mathsf{sfAEAD}}^{\text{IND-sfCCSA}-0}(B) \to 1\right] = \\ = \sum_{i=1}^{N} \frac{1}{N} \cdot P_i - \sum_{i=1}^{N} \frac{1}{N} \cdot P_{i-1} = \frac{1}{N}(P_N - P_0).\end{aligned}$$

Therefore, $\mathsf{Adv}_{(\mathsf{sfAEAD},\mathsf{G}')_h}^{\text{IND-sfCCSA}}(A) = N \cdot \mathsf{Adv}_{\mathsf{sfAEAD}}^{\text{IND-sfCCSA}}(B)$, which completes the proof of the theorem.

C Basic security estimates

To obtain specific estimates for the TLS-REC protocol, you will need to recall some estimates for the cryptographic schemes used in the protocol. The CTR-ACPKM [7] mode is used as the encryption scheme in TLS-REC, the OMAC [2] mode is used as the MAC scheme, and the TLSTREE 6 function based on `KDF` [5] function is used for key derivation.

C.1 Known estimates for CTR-ACPKM, OMAC, `KDF`

The OMAC mode is analyzed in [26]. This paper formulates the following statement about OMAC security.

Theorem 4 (OMAC security)**.**

$$\mathsf{InSec}^{\mathrm{PRF}}_{OMAC}(t,q,l) \leqslant \mathsf{InSec}^{\mathrm{PRP\text{-}CPA}}_{E}(t_1, ql+1) + \frac{4q^2l^2}{2^n n^2}.$$

The CTR-ACPKM mode is analyzed in [10] This paper formulates the following statement about CTR-ACPKM security.

Theorem 5 (CTR-ACPKM security)**.** *Let N be the parameter of CTR-ACPKM mode. Then for any adversary A with time complexity at most t that makes queries, where the maximal message length is at most m ($m \leqslant 2^{n/2-1}$) blocks and the total message length is at most σ blocks, there exists an adversary B such that*

$$\mathsf{Adv}^{\mathrm{IND\text{-}CPNA}}_{CTR\text{-}ACPKM_N}(A) \leqslant$$
$$\leqslant l \cdot \mathsf{Adv}^{\mathrm{PRP\text{-}CPA}}_{E}(B) + \frac{(\sigma_1+s)^2 + \ldots + (\sigma_{l-1}+s)^2 + (\sigma_l)^2}{2^{n+1}}$$

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 $\sigma_j \leqslant 2^{n-1}$, $\sigma_1 + \ldots + \sigma_l = \sigma$. 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.

The `KDF` function is analyzed in [11] This paper formulates the following statement on the relationship between the security of the `KDF` function in the PRF model and the security of the HMAC function in the PRF model.

Theorem 6 (KDF security)**.**

$$\mathsf{InSec}^{\mathrm{PRF}}_{\mathtt{KDF}}(T,q,u) \leqslant \mathsf{InSec}^{\mathrm{PRF}}_{HMAC}(T+q, q, u+1).$$

C.2 TLSTREE security in the PRG model

Recall the construction of a pseudo-random generator with an internal state from [9], which is a generalization of the TREE algorithm. This construction is a balanced tree, in which each vertex that is not a leaf corresponds to a separate pseudo-random generator with an internal state. Each sheet corresponds to the output unit of the whole structure. The output blocks of all generators, except those on the lower level, fill the random tape of the generators' key generation

algorithms with a lower level. We assume that all levels of this tree are L. More formally, this definition can be written as follows.

Definition 18. *Let $\mathsf{G}_l = (\mathsf{K}_l, \mathsf{N}_l)$ be pseudorandom generators with the set of states $\mathcal{K}$ and the set of blocks $\mathcal{B}$, those are used on level l. Let a_l be a number of blocks, which gives each generator on level l. Define a pseudorandom generator $\mathsf{G} = (\mathsf{K}, \mathsf{N})$ with the set of states $\mathcal{K}$ and the set of blocks $\mathcal{B}$ as follows*

$\underline{\mathsf{K}()}$
$St_1 \leftarrow \mathsf{K}_1()$
for l ***do***1L-1
  $(Out_l, St_l) \leftarrow \mathsf{N}_l(St_l)$
  $St_{l+1} \leftarrow \mathsf{K}_{l+1}(Out_l)$
end for
$St \leftarrow \langle St_1, \ldots, St_{L-1}, 0\rangle$
return St

$\underline{\mathsf{N}(St)}$
$\langle St_1, \ldots, St_{L-1}, i\rangle \leftarrow St$
$l \leftarrow L-1;\ d \leftarrow i+1$
while $d \mod a_l = 0$ ***do***
  $d \leftarrow \lfloor d/a_l \rfloor;\ l \leftarrow l-1$
end while
$(Out_l, St_l) \leftarrow \mathsf{N}_l(St_l)$
while $l < L-1$ ***do***
  $l \leftarrow l+1;\ St_l \leftarrow \mathsf{K}_l(Out_{l-1})$
  $(Out_l, St_l) \leftarrow \mathsf{N}_l(St_l)$
end while
$St \leftarrow \langle St_1, \ldots, St_{L-1}, i+1\rangle$
return (Out_{L-1}, St)

For this construction, it was proved in [9] that the pseudo-random generator G will be secure in the PRG model if all the generators G_l used in the construction are secure in the same model. More strictly, the following theorem is true.

Theorem 7 (Abdalla, Bellare [9])**.** *Let each $\mathsf{G}_l = (\mathsf{K}_l, \mathsf{N}_l)$ be a secure stateful generator for all $l = 1, \ldots, L-1$. Let $a_0 = 1$ Let $n_l = \prod_{j=0}^{l-1} a_l$ be the total number of nodes at level l. Let G be the overall stateful generator formed out of the basic stateful generators as described in Definition 18. Let A be an adversary for G in the* PRG *model with time complexity at most t. Then adversaries $B_1, \ldots, B_{L-1}$ exist with time complexity $t_1 \approx t$ such that*

$$\mathsf{Adv}^{\mathrm{PRG}}_{\mathsf{G},n_L}(A) \leqslant \sum_{l=1}^{L-1} n_l \cdot \mathsf{Adv}^{\mathrm{PRG}}_{\mathsf{G}_l,a_l}(B_l)$$

Apply this theorem to the algorithm TREE, defined in Section 6. If we represent this algorithm as a tree, as suggested in [9], then it is obvious that

the number of levels is $L = 4$, and $a_l = 2^{13},\ l = 1, 2, 3$. Thus we receive the following security estimate for TREE.

$$\mathsf{Adv}^{\mathrm{PRG}}_{\mathrm{TREE}}(A) \leqslant (1 + 2^{13} + (2^{13})^2) \cdot 2 \cdot \mathsf{Adv}^{\mathrm{PRG}}_{\mathtt{KDF}}(B)$$

This unequality can be rewritten as.

$$\mathsf{InSec}^{\mathrm{PRG}}_{\mathrm{TREE}}(t, (2^{13})^3) \leqslant$$
$$\leqslant (1 + 2^{13} + (2^{13})^2) \cdot 2 \cdot \mathsf{InSec}^{\mathrm{PRG}}_{\mathtt{KDF}}(t_1, 2^{13}) \leqslant 2^{28} \cdot \mathsf{InSec}^{\mathrm{PRG}}_{\mathtt{KDF}}(t_1, 2^{13}).$$

Note that the additional factor of 2 with terms $\mathsf{Adv}^{\mathrm{PRG}}_{\mathtt{KDF}}(B)$ and $\mathsf{InSec}^{\mathrm{PRG}}_{\mathtt{KDF}}(t_1, 2^{13})$ arises from the fact that each generator at the top of the tree generates keys from the set $K_{\mathsf{SE}} \times K_{\mathsf{MA}}$, using the KDF function twice.

If the algorithm TREE generated less than $(2^{13})^3$ keys, then the estimate will look like this:

$$\mathsf{InSec}^{\mathrm{PRG}}_{\mathrm{TREE}}(t, N) \leqslant \left(1 + \left\lceil \frac{N}{(2^{13})^2} \right\rceil + \left\lceil \frac{N}{2^{13}} \right\rceil\right) \cdot 2 \cdot \mathsf{InSec}^{\mathrm{PRG}}_{\mathtt{KDF}}(t_1, 2^{13})$$

Applying the estimate for KDF we obtain:

$$\mathsf{InSec}^{\mathrm{PRG}}_{\mathrm{TREE}}(t, N) \leqslant \left(1 + \left\lceil \frac{N}{(2^{13})^2} \right\rceil + \left\lceil \frac{N}{2^{13}} \right\rceil\right) \cdot 2 \cdot \mathsf{InSec}^{\mathrm{PRF}}_{\mathrm{HMAC}}(t_1, 2^{13}, 2)$$

Fuzzy Extractors Security under Several Models of Biometric Data

Grigory Marshalko[1] and Yulia Trufanova[2]

[1] Technical Committee for Standardization
«Cryptography and security mechanisms» (TC 026), Russia
[2] Lomonosov Moscow State University, Russia
marshalko_gb@tc26.ru, kiten72@yandex.ru

Abstract

We study security of a fuzzy extractor under two probablistic models of a biometric vector: a sequence of independent non-equiprobable random variables and a Markov chain. We propose approaches for estimation of the average amount of work needed for searching for the correct secret vector of a fuzzy extractor depending on the parameters of the biometric vector model.

Keywords: biometric identification, fuzzy extractor, Markov chain, sequence of independent non-equiprobable random variables.

1 Introduction

Fuzzy extractors are cryptographic mechanisms that allow linking the vector of biometric data with a string of characters, which can be a cryptographic key or an identification vector, protecting the latter from compromise. Such schemes are convenient for use, for example, in mobile devices, because they actually implement a secure storage of user biometric data.

The milestone study [1] proposes formal proofs for the fuzzy extractor security in the form of evaluaition of the statistical distance between the actual distribution of vector bits of the output sequence (so-called helper string or simply helper) of the extractor and the equiprobable distribution. The model used in this study corresponds to the situation of a single user enrollment. At the same time, it was shown in [2] that if a user enrolls on different resources using the same biometric parameters and the attacker has the ability to analyze the corresponding helpers, then classical constructions of fuzzy extractors

become insecure. In [3] a set of attacks exploiting helper manipulation for fuzzy extractors used in physically unclonable functions are presented.

In this paper we consider a situation when the attacker may have information about the parameters of the biometric vector bits probability distribution. Indeed, depending on the biometric technology used, the method of digitizing the biometric image, technical features of the biometric information registration device, the specified distribution may differ from the equiprobable one and under certain conditions it could be known to the attacker. If the deviation is not large, it can be shown that this will not have a significant impact on the security of fuzzy extractors that uses XOR for linking a secret string and a biometric vector (this follows from the study [6] of Vigenere cipher security), otherwise, the attacker can reduce the number of probable variants of a secret string and mount the search algorithm which complexity is less than the complexity of brute force algorithm.

We consider two biometric vector models:

- a model in which bits of a biometric vector form a sequence of independent non-equiprobable random variables;
- a model in which bits of a biometric vector form a Markov chain.

Based on the general methodology described in [4] for the simplest fuzzy extractor with Hamming correcting code, the paper describes approaches for estimating the average amount of work of the secret string search algorithm.

The work is organized as follows: the necessary definitions are given in Section 2, Section 3 gives the results for the independent non-equiprobable model, Section 4 gives the results for the Markov chain, Section 5 is devoted to the comparison of the estimates for the two models considered. Some proofs and tables are provided in the appendix.

2 Notations and definitions

We will use the following notations throughout the work:

- w – biometric vector,
- n – length of a biometric vector w,
- s – secret string,
- q – helper string (helper),
- k – secret string s length,

– $N = 2^n$ – total number of binary vectors of length n,
– $m = n - k$ – number of parity bits of Hamming code,
– $d = 3$ – code distance,
– dist – Hamming distance,
– wt – Hamming weight.

We consider the following fuzzy extractor scheme.

Let C: $\{0,1\}^k \to \{0,1\}^n$ – the encoding function of the correction code, D: $\{0,1\}^n \to \{0,1\}^k$ – the corresponding decoding function. The fuzzy extractor based on the bitwise vectorial XOR has the following form:

– $\mathsf{Gen}[w] = \langle s, q \rangle$, where $s \in_R \{0,1\}^k$, $q = w \oplus C[s]$;
– $\mathsf{Rep}[w', q] = D[w' \oplus q] = D[w' \oplus w \oplus C[s]] = s$, if $\mathsf{dist}[w, w'] \leqslant d$.

If, as already noted, the attacker knows some a priori information about the distribution of bits of the biometric vector, he can, for example, in accordance with the approach described in [4], rearrange vectors w in descending order according to their probabilities, and subsequently perform search and evaluation of the required vector by the following formula: $C[s] = q \oplus w$.

Next we will be interested in the average number of steps of the search algorithm to determine the correct $C[s]$. For simplicity we will consider the case when Hamming code is used to correct errors.

3 Independent non-equiprobable bits

Consider th case when bits of the biometric vector w are independent and distributed under Bernoulli law with the known parameter $p \neq \frac{1}{2}$. For the sake of certainty $p < \frac{1}{2}, q = 1 - p$.

Arrange all possible vectors of length n in descending order of their probability:

$$w_1 = (0, 0, \ldots, 0, 0),$$

$$\ldots$$

$$w_N = (1, 1, \ldots, 1, 1).$$

The whole set of binary vectors of the length n is divided into $n + 1$ classes $W^0, \ldots, W^n$ whose elements have the same probability. W^i represents a set of

vectors of length n and weight i:

$$|W^i| = \binom{n}{i}, \forall w_j \in W^i : \mathsf{wt}(w_j) = i, \mathsf{Pr}(w_j) = p^i \cdot q^{(n-i)}.$$

We can XOR w_j with $q = (q_1, \ldots, q_n)$. As a result we obtain the following set:

$$s_1 = (q_1, \ldots, q_n),$$

$$\ldots$$

$$s_j = ((w_j)_1 \oplus q_1, \ldots, (w_j)_n \oplus q_n),$$

$$\ldots$$

$$s_N = (q_1 \oplus 1, \ldots, q_n \oplus 1).$$

Note that $s_k \oplus s_j = w_k \oplus w_j$.

The partition $W^0, \ldots, W^n$ generates the corresponding partition of the set $\{s_j, j = \overline{1, N}\}$ into subsets $S^0, \ldots, S^n$:

$$|S^i| = \binom{n}{i}, \mathsf{Pr}(s_j) = p^i \cdot q^{(n-i)}, \forall s_j \in S^i.$$

Let's evaluate the number of codewords in each of $S^0, \ldots, S^n$ classes. Denote this numbers correspondingly as $K^0, \ldots, K^n$.

Lemma 1. *The number K^i of codewords in class $S^i, i = \overline{2, n}$ is equal to*

$$K^i = \frac{\binom{n}{i-1} - K^{i-1} - K^{i-2} \cdot (n - (i-2))}{i}.$$

Thus, we got a partition of the set $K^i = K^{n-i}, i = 0, \ldots, (n-1)/2$.

We have split the set of vectors $\{s_j, j = \overline{1, N}\}$ into non-crossing spheres with centers in code words. The number of such spheres is equal to the number of code words: $\sum\limits_{i=0}^{n} K^i = 2^k$.

Next, we use Arbekov's approach (see [4]) to estimate the average amount of work before determining the true secret string of the fuzzy extractor. The essence of the algorithm is a partial search of the set $\{s_i\}$. In this set, there exists l such that $s_l = C[s]$, where s is the secret string. Instead of searching through the whole ordered set, we will only search through the vectors from its

initial (highly probable) part $\widetilde{S} \subset \{s_i\}$, including all the spheres with centers in the classes $S^0, \ldots, S^r$: $|\widetilde{S}| = M < N$.

We have:

$$R(M) = \mathbf{E}\xi\mathbf{E}(R|s_l \notin \widetilde{S}) + \mathbf{E}(R|s_l \in \widetilde{S}) = \\ = \frac{1}{\pi(M)} \left((1 - \pi(M)) \cdot M + \sum_{j=0}^{V} jp_j \right).$$

We denote here ξ - as the number of steps of guessing algorithm, $V = \sum_{i=0}^{r} K^i$ - the number of spheres in $\widetilde{S}$, r - the maximum number of the set s^j from $\widetilde{S}$, M - the number of vectors in highly probable set $\widetilde{S} \subset \{s_i\}$, which includes all spheres with centers in classes $S^0, \ldots, S^r$: $|\widetilde{S}| = M < N$.

With Lemma 1 according to [4] we have, that the average amount of work is equal to:

$$R(M) = (\pi(M))^{-1} \left((1 - \pi(M)) \cdot M + \sum_{j=0}^{V} jp_j \right) = \\ = \left(K^0 \cdot \left(q^n + \binom{n}{1} p^1 q^{n-1} \right) + \sum_{j=1}^{r} K^j \cdot \left(p^j q^{n-j} + \binom{j}{1} p^{j-1} q^{n-(j-1)} + \right. \right. \\ \left. \left. + \binom{n-j}{1} p^{j+1} q^{n-(j+1)} \right) \right)^{-1} \cdot \left[\left(1 - \left(K^0 \left(q^n + \binom{n}{1} p^1 q^{n-1} \right) + \right. \right. \right. \\ \left. \left. \left. + \sum_{j=1}^{r} K^j \left(p^j q^{n-j} + \binom{j}{1} p^{j-1} q^{n-(j-1)} + \binom{n-j}{1} p^{j+1} q^{n-(j+1)} \right) \right) \right) \cdot \\ \cdot \left(\sum_{i=0}^{r} K^i (1 + n) \right) + \left(K^0 \left(q^n + \binom{n}{1} p^1 q^{n-1} \right) + \sum_{j=1}^{r} \frac{\left(\sum_{i=1}^{j-1} K^i + 1 + \sum_{i=1}^{j} K^i \right)}{2} \cdot \right. \\ \left. \cdot K^j \cdot \left(p^j q^{n-j} + \binom{j}{1} p^{j-1} q^{n-(j-1)} + \binom{n-j}{1} p^{j+1} q^{n-(j+1)} \right) \right) \Bigg].$$

4 Markov chain

Now we consider the case when biometric vector bits form a homogeneous Markov chain with a discrete time and two states $\chi = \{0, 1\}$. Let the initial states of Markov's chain are equiprobable, i.e. $p_0^{(0)} = p_1^{(0)} = 1/2$.

In this case, it is difficult to estimate the average amount of work in general. However, we can propose an algorithm for evaluating the characteristics we are interested in for the number of specific values of fuzzy extractor parameters.

Let the Markov chain transition matrix is known – $P = \begin{pmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{pmatrix}$, where $p_{ij} = \mathsf{Pr}(x_r = i | x_{r-1} = j), i, j \in \{0, 1\}, \forall r \in \{1, 2, \ldots\}$.

Any sample w of this Markov chain could be described by the transition matrix $F = \begin{pmatrix} f_{00} & f_{01} \\ f_{10} & f_{11} \end{pmatrix}$, where f_{ij} – the number of transitions from i to j, $i, j \in \{0, 1\}$.

We will use the following notations:

- $f_{i+} = \sum_{j \in \{0,1\}} f_{ij}$;
- $f_{+j} = \sum_{i \in \{0,1\}} f_{ij}$.

Note that f_{i+} is the sum of elements of the transition matrix F, that is the number of transitions from the state of i from 1 to $(n-1)$ step. Similarly, f_{+j} is the sum of elements of column i of the transition matrix F that is, the number of transitions to the state j from 2 to n step.

Let $N_{xy}^{(n)}(F)$ be the number of vectors obtained by the Markov chain for n steps with the transition matrix F, starting from the state x and finishing at the state y.

4.1 Evaluation of the initial and the final state according to the matrix F

Let us know the transition matrix F for a sample of the Markov chain for n steps.

Proposition 1. $|f_{01} - f_{10}| \leqslant 1$.
Moreover, if $f_{01} < f_{10}$, the vector w starts at 1 and ends at 0.
If $f_{01} > f_{10}$, w starts at 0 and ends at 1.
If $f_{01} = f_{10} \neq 0$, w starts and ends at the same state 0 or 1.

Corollary 1. $f_{01}, f_{10} \leqslant (n-1)/2$.

Using the statement 1, we can easily restore the initial and final states of $w = \{w_1, \ldots, w_n\}$ according to the following rule:

$f_{01} = f_{10} = 0$:

$f_{00} > 0 \Rightarrow w_0 = w_n = 0;$
$f_{11} > 0 \Rightarrow w_0 = w_n = 1;$

$f_{01} = f_{10} \neq 0 \Rightarrow w_0 = w_n = 0$ or $w_0 = w_n = 1;$
$f_{01} > f_{10} \Rightarrow w_0 = 0, w_n = 1;$
$f_{01} < f_{10} \Rightarrow w_0 = 1, w_n = 0.$

4.2 Recovery of all possible transition matrices F

Note, that the number of transitions in a vector of length n is equal to $\sum_{i,j \in \{0,1\}} f_{ij} = n - 1$.

Proposition 2. *Let f_{01}, f_{10} are fixed.*
Then $f_{00} \in \{0, 1, \ldots, (n-1) - f_{01} - f_{10}\}$.

Corollary 2. $f_{11} = (n-1) - f_{01} - f_{10} - f_{00} \in \{(n-1) - f_{01} - f_{10}, \ldots, 1, 0\}$.

Using the statement 2 and the corollaries 1, 2, we can restore all possible transition matrices F for n steps for a given Markov chain sample according to the following algorithm:

1. For all i_1 from 0 to $(n-1)$:
2. For all i_2 from $(n-1-i_1)$ to 0:
3. If $((n-1) - i_1 - i_2) \mod 2 = 1$, then

$i_3 = ((n-1) - i_1 - i_2 - 1)/2,$

return: $\begin{pmatrix} i_1 & i_3 \\ i_3 + 1 & i_2 \end{pmatrix}, \begin{pmatrix} i_1 & i_3 + 1 \\ i_3 & i_2 \end{pmatrix};$

else

$i_3 = ((n-1) - i_1 - i_2)/2,$

return: $\begin{pmatrix} i_1 & i_3 \\ i_3 & i_2 \end{pmatrix}$.

4.3 Evaluation of a Markov chain sample probability with transition matrix F

The probability of a Markov chain sample w with transition matrix F could be evaluated according to the following relation:

$$\mathsf{Pr}(w) = \frac{1}{2} \prod_{i,j \in \{0,1\}} p_{ij}^{f_{ij}}.$$

4.4 Evaluation of a number of Markov chain samples with given transition matrix F

It was shown in [5] that

$$N_{uv}^{(n)}(F) = \frac{\prod_{i \in \{0,1\}} f_{i+}!}{\prod_{i,j \in \{0,1\}} f_{ij}!} F_{vu}^*,$$

where F_{vu}^* – is a cofactor for element (v,u) of a matrix $F^* = \{f_{ij}^*\}$:

$$f_{ij}^* = \begin{cases} \delta_{ij} - f_{ij}/f_{i+} & \text{, if } f_{i+} > 0; \\ \delta_{ij} & \text{, if } f_{i+} = 0. \end{cases}$$

Here $\delta_{ij} = \begin{cases} 1 & \text{, if } i = j \\ 0 & \text{, if } i \neq j \end{cases}$ – Kronecker symbol.

4.5 Construction of the highly probable set

After performing the previous steps, we can build a table of the following form:

$$[\ f_{00}^t \mid f_{01}^t \mid f_{10}^t \mid f_{11}^t \mid \mathsf{Pr}(F^t) \mid N^{(n)}(F^t)\ ].$$

Here the columns contain the following data:

– $f_{ij}^t, \forall i,j \in \{0,1\}$ – elements of the transition matrix of the sample for n steps;

– $\mathsf{Pr}(F^t)$ – the probability to get the sample with the transition matrix F^t;
– $N^{(n)}(F^t) = \sum_{i_1,i_n \in \{0,1\}} N^{(n)}_{i_1,i_n}(F^t)$ – the number of samples with the transition matrix F, which is equal to $N^{(n)}_{i_1,i_n}(F^t)$, where i_1 and i_n – teh initial and the final steps.

Note, that from the proposition 1 it follows that:

– $N^{(n)}(F^t) = N^{(n)}_{1,0}(F^t)$, if $f_{01} < f_{10}$;
– $N^{(n)}(F^t) = N^{(n)}_{0,1}(F^t)$, if $f_{01} > f_{10}$;
– $N^{(n)}(F^t) = N^{(n)}_{0,0}(F^t) + N^{(n)}_{1,1}(F^t)$, if $f_{01} = f_{10}$.

We can sort the resulting table by column Pr. We want to get a sample (highly probable) set of size M. To do this, select the first Z records in the list so that $\sum_{t=1}^{Z-1} N^{(n)}(F^t) < M$ and $\sum_{t=1}^{Z} N^{(n)}(F^t) \geqslant M$. On the basis of them, we will build the sample set. To do this, we need to restore $N^{(n)}_{i_1,i_n}$ vectors for each of the M records - Markov's chain samples with initial and final states i_1, i_n.

After the construction of the sample set, one can perform calculations of the average amount of work before determining the true secret string, similar to [4].

4.6 Recovery of vectors with the initial and final states and transition matrix F

We know the initial state j_1, the final state j_2 and the transition matrix F. That is, the general form of the sought vectors is as follows

$$\begin{array}{cccccccc} 1 & 2 & 3 & \dots & n-2 & n-1 & n \\ (j_1 & \_ & \_ & \cdots & \_ & \_ & j_2). \end{array}$$

Let's consider a recursive algorithm for recovering unknown bits.

Input:

l – vector length,

j_1, j_2 – the initial and the final states,

F – transition matrix for $(l-1)$ steps.

Output:

$\{w\}$ – the set of vectors of length l.

The general idea:

Consider the matrix F. Let's determine whether such matrix allows starting

from the state j_1 ($f_{j_1+} \geqslant 1$) and returning to the state j_2 ($f_{+j_2} \geqslant 1$).
If the answer is yes, we will try to add one bit to the right of the first character and to the left of the last character, reducing the corresponding matrix elements F by 1.
The number of unknown characters will thus be reduced by a factor of 2:

$$\begin{array}{cccccccc} 1 & 2 & 3 & \dots & n-2 & n-1 & n \\ \big(j_1 & j_1^1 & \_ & \cdots & \_ & j_2^2 & j_2 \big). \end{array}$$

We repeat these steps for the obtained representations until we have the whole vector length exhausted.

1. Restore all possible transition matrices F for n steps (cl. 4.2).
2. Sort the obtained set in descending order of probability to get the Markov chain sample with this transition matrix (cl. 4.3).
3. Determine possible pairs of initial and final states for each F matrix (cl. 4.1).
4. Calculate the number of $N_{ij}(F)$ vectors formed along the Markov chain and having the specified transition matrix, initial and final states (cl. 4.4).
5. Compile the set of vectors to be searched in: for each transition matrix from the obtained set and the corresponding initial and final states, restore all possible vectors $\{w\}$ (cl. 4.5, 4.6).
6. Calculate the average amount of work.

5 Comparison of estimates

Figure 5 shows the differences between the average amount of work for the model with independent non-equiprobable bits $R_{st}(M, p_0^{st})$ and the Markov chain model $R_2(M, p_{00}, p_{11})$. The corresponding tables are given in the appendix B. For the Markov model, the probabilities of outcomes in order to correspond to the first model were calculated as values of Markov chain limit probabilities.

As one would expect, the average amount of work in the case of the Markov chain model is lower than in the case of independent, non-equiprobable bits. It should be noted that for the bits of biometric vectors in practical applications, it

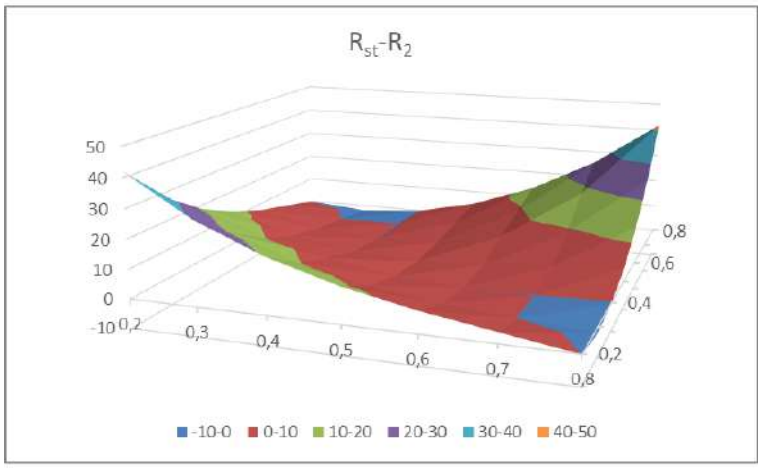


Figure 1: Difference of amounts of work for the length of a biometric vector $n = 255$ without truncating of search for the considered models (see Appendix B for details).

is the correlation type dependencies that are observed, so that the second model seems to be more adequate. However, for the real biometric vector one should expect the presence of models with different parameters for different bits.

We performed some evaluation of probablistic distribution on real biometric data. We considered biometric data extracted from AT&T face database with LBP algorithm. We tested several postprocessing techniques for removal of correlated and biased bits, and observed some remnant correlation and bias. This is supports the adequacy of the considered models.

References

[1] Y. Dodis, R. Ostrovsky, L. Reyzin, A. Smith, "Fuzzy extractor: How to generate strong keys from biometrics and other noisy data", *Advances in cryptology'04*, 2004, 523–540.

[2] Xavier Boyen, "Reusable Cryptographic Fuzzy Extractors", *ACM Conference on Computer and Communications Security—CCS 2004*, 2004, 82–91.

[3] G.T. Becker, *Robust Fuzzy Extractors and Helper Data Manipulation Attacks Revisited: Theory vs Practice*, 2017, https://eprint.iacr.org/2017/493.pdf.

[4] I. M. Arbekov, "Key security criteria", *Mat. Vopr. Kriptogr.*, **7**:1 (2016), 39–56.

[5] Patrick Billingsley, "Statistical Methods in Markov Chains", *The Annals of Mathematical Statistics*, **32**:1 (1961), 12–40.

[6] Boris Ryabko, "The Vernam cipher is robust to small deviations from randomness", *Probl. Inf. Transm.*, **51**:1, 82-86.

A Proofs of the results

A.1 Proof of Lemma 1

Proof. The proof is recursive.

Consider the class $S^0 = \{s^0\}, s^0 = q$.

Let the vector q be a code word, then $K^0 = 1 \Longrightarrow K^1 = 0$, as $\forall s_j \in S^1, \mathsf{dist}(s^0, s_j) = \mathsf{wt}(s^0 \oplus s_j) = \mathsf{wt}(w^0 \oplus w_j) = 1 < d$, where $w_j \in W^1$, i.e. any vector from S^1 lies in a sphere with the center in the code word q of a single radius and is not a code word itself.

Now suppose that we have found all $K^j, j = \overline{1, i-1}, i \geqslant 2$. We will find the number of code words K^i in the class S^i, obtained by adding various binary vectors $w_l \in W^i$ of weight i with the vector q. Note that each sphere with the center in the code word from this class includes $\binom{i}{1}$ vectors from the class S^{i-1}, and each vector belongs to only one such sphere.

$|S^{i-1}| = \binom{n}{i-1}$ – number of vectors in S^{i-1},

K^{i-1} – number of code words in the class S^{i-1},

$K^{i-2} \cdot \binom{n-(i-2)}{1}$ – number of vectors from S^{i-1} belonging to spheres with centers from S^{i-2}. Therefore, the number of vectors belonging to the spheres with centers from S^i is equal to :

$$\binom{n}{i-1} - K^{i-1} - K^{i-2} \cdot (n - (i-2)).$$

Each such sphere includes 1 vectors from S^{i-1}, and the spheres do not intersect. As a result, we get:

$$K^i = \frac{\binom{n}{i-1} - K^{i-1} - K^{i-2} \cdot (n - (i-2))}{i}, \forall i \geqslant 2.$$

If q is not a code word, then $K^0 = 0, K^1 = 1$. The formula for calculating the number of code words in each class remains unchanged. □

A.2 Proof of proposition 1

Proof. Consider all possible relations between f_{01} and f_{10}.

1. If $f_{01} = f_{10} \neq 0$, then having started the movement from the state of $i \in \{0, 1\}$, we will leave it $f_{i,1-i}$ times and we will come back to it $f_{1-i,i} =$

$f_{i,1-i}$ times again, i.e., we will eventually find ourselves in the same state from which the movement began. At $f_{01} = f_{10} = 0$ the vector completely consists of 0 or 1, depending on f_{11} and f_{00}.

2. Now let $|f_{01} - f_{10}| = 1$. Let's consider all possible options:

   (a) Let$f_{10} = f_{01} + 1$. Let's consider two cases:
   - We start from 0. Then we will exit $f_{01} - 1 = f_{10}$ times and return to f_{10} times. But we still have one more transition from 0 to 1, because $f_{01} = f_{10} + 1$, therefore, the system comes to state 1.
   - We start from 0. Then we will leave it f_{01} times and return to $f_{10} - 1 = f_{01}$ times. But we still have one more transition from 1 to 0, however, we are in the state of 0 and will not be able to implement it. Therefore, this case is impossible.

3. After considering the previous case ($|f_{01} - f_{10}| = 1$), it becomes clear that $|f_{10} - f_{01}| > 1$ cannot be (otherwise, it will not be possible to exhaust all the required transitions).

□

A.3 Proof of the proposition 2

Proof. We fix the initial state of Markov chain sample, for example, 0. Then the sample should have f_{01} "starts" from the state of 0 to 1 and f_{10} "'returns'" to the initial state of 0. It will also have f_{00} and f_{11} – as 1.

If the values are known to be f_{01}, f_{10}, then by varying the intervals between the next "start"' and "return"' to the original state, any value can be obtained from f_{00} to $(n-1) - f_{01} - f_{10}$. □

B Comparison table

- n – the length of the biometric vector w,
- $r \in \{0, \ldots, n\}$ – the parameter of the truncated key search algorithm,
- $M(r) = \sum_{j=0}^{r} \binom{n}{i}$ – the power of the sample set,
- $R_2(M, p_{00}, p_{11})$ – the average amount of work before determining the true secret string in Markov chain model,

– p_0^{st} – the component of stationary distribution of Markov chain probabilities,
– $R_{st}(M, p_0^{st})$ – the average amount of work before determining the true key in independent non-equiprobable bits model (Bernoulli scheme).

n	p_{00}	p_{11}	r	$R_2(M, p_{00}, p_{11})$	p_{st}	$R_{st}(M, p_0^{st})$
255	0.2	0.2	55	$2^{190.44...}$	0.5	2^{254}
255	0.2	0.3	55	$2^{192.32...}$	0.4667	$2^{241.77...}$
255	0.2	0.4	55	$2^{196.38...}$	0.4286	$2^{228.51...}$
255	0.2	0.5	55	$2^{202.46...}$	0.3846	$2^{215.55...}$
255	0.2	0.6	55	$2^{202.99...}$	0.3333	$2^{203.52...}$
255	0.2	0.7	55	$2^{194.47...}$	0.2727	$2^{193.70...}$
255	0.2	0.8	55	$2^{190.34...}$	0.2	$2^{186.80...}$
255	0.2	0.9	55	$2^{177.17...}$	0.1	$2^{168.82...}$
255	0.3	0.3	55	$2^{197.68...}$	0.5	2^{254}
255	0.3	0.4	55	$2^{206.65...}$	0.4615	$2^{239.88...}$
255	0.3	0.5	55	$2^{216.08...}$	0.4167	$2^{224.76...}$
255	0.3	0.6	55	$2^{212.35...}$	0.3636	$2^{210.23...}$
255	0.3	0.7	55	$2^{197.69...}$	0.3	$2^{197.52...}$
255	0.3	0.8	55	$2^{192.06...}$	0.2222	$2^{188.72...}$
255	0.3	0.9	55	$2^{180.96...}$	0.125	$2^{173.65...}$
255	0.4	0.4	55	$2^{219.93...}$	0.5	2^{254}
255	0.4	0.5	55	$2^{233.34...}$	0.4545	$2^{237.35...}$
255	0.4	0.6	55	$2^{219.93...}$	0.4	$2^{219.81...}$
255	0.4	0.7	55	$2^{200.69...}$	0.3333	$2^{203.52...}$
255	0.4	0.8	55	$2^{192.78...}$	0.25	$2^{191.20...}$
255	0.4	0.9	55	$2^{183.42...}$	0.1429	$2^{178.43...}$
255	0.5	0.5	55	2^{254}	0.5	2^{254}
255	0.5	0.6	55	$2^{224.18...}$	0.4444	$2^{233.81...}$
255	0.5	0.7	55	$2^{202.67...}$	0.375	$2^{213.04...}$
255	0.5	0.8	55	$2^{193.37...}$	0.2857	$2^{195.40...}$
255	0.5	0.9	55	$2^{185.18...}$	0.1667	$2^{182.91...}$
255	0.6	0.6	55	$2^{219.93...}$	0.5	2^{254}
255	0.6	0.7	55	$2^{202.55...}$	0.4286	$2^{228.51...}$
255	0.6	0.8	55	$2^{193.83...}$	0.3333	$2^{203.52...}$
255	0.6	0.9	55	$2^{186.20...}$	0.2	$2^{186.80...}$
255	0.7	0.7	55	$2^{197.79...}$	0.5	2^{254}
255	0.7	0.8	55	$2^{193.30...}$	0.4	$2^{219.81...}$
255	0.7	0.9	55	$2^{185.70...}$	0.25	$2^{191.20...}$
255	0.8	0.8	55	$2^{191.85...}$	0.5	2^{254}
255	0.8	0.9	55	$2^{182.96...}$	0.3333	$2^{203.52...}$
255	0.9	0.9	55	$2^{171.75...}$	0.5	2^{254}
255	0.2	0.2	255	$2^{212.03...}$	0.5	2^{254}
255	0.2	0.3	255	$2^{224.51...}$	0.4667	$2^{252.85...}$
255	0.2	0.4	255	$2^{231.88...}$	0.4286	$2^{250.76...}$
255	0.2	0.5	255	$2^{235.00...}$	0.3846	$2^{247.12...}$

255	0.2	0.6	255	$2^{233.71...}$	0.3333	$2^{240.98...}$
255	0.2	0.7	255	$2^{226.98...}$	0.2727	$2^{230.56...}$
255	0.2	0.8	255	$2^{212.16...}$	0.2	$2^{212.06...}$
255	0.3	0.3	255	$2^{235.77...}$	0.5	2^{254}
255	0.3	0.4	255	$2^{242.34...}$	0.4615	$2^{252.62...}$
255	0.3	0.5	255	$2^{244.87...}$	0.4167	$2^{249.91...}$
255	0.3	0.6	255	$2^{242.99...}$	0.3636	$2^{244.88...}$
255	0.3	0.7	255	$2^{235.77...}$	0.3	$2^{235.72...}$
255	0.3	0.8	255	$2^{220.58...}$	0.2222	$2^{218.56...}$
255	0.4	0.4	255	$2^{248.56...}$	0.5	2^{254}
255	0.4	0.5	255	$2^{250.91...}$	0.4545	$2^{252.28...}$
255	0.4	0.6	255	$2^{248.57...}$	0.4	$2^{248.56...}$
255	0.4	0.7	255	$2^{241.02...}$	0.3333	$2^{240.98...}$
255	0.4	0.8	255	$2^{225.86...}$	0.25	$2^{225.59...}$
255	0.5	0.5	255	2^{254}	0.5	2^{254}
255	0.5	0.6	255	$2^{250.91...}$	0.4444	$2^{251.74...}$
255	0.5	0.7	255	$2^{243.11...}$	0.375	$2^{246.14...}$
255	0.5	0.8	255	$2^{228.33...}$	0.2857	$2^{233.12...}$
255	0.6	0.6	255	$2^{248.57...}$	0.5	2^{254}
255	0.6	0.7	255	$2^{241.66...}$	0.4286	$2^{250.76...}$
255	0.6	0.8	255	$2^{227.69...}$	0.3333	$2^{240.98...}$
255	0.7	0.7	255	$2^{235.78...}$	0.5	2^{254}
255	0.7	0.8	255	$2^{223.05...}$	0.4	$2^{248.55...}$
255	0.8	0.8	255	$2^{212.21...}$	0.5	2^{254}

Table 1: Comparison of the results for two models.

Information Hiding

Data Embedding Based on Linear Hash Functions

Boris Ryabko[1,2] and Andrey Fionov[1,3]

[1] Institute of Computational Technologies SB RAS, Russia
[2] Novosibirsk State University, Russia
[3] Siberian State Univ. of Telecommunications and Computer Science, Russia
boris@ryabko.net, a.fionov@ieee.org

Abstract

Embedding hidden data in digital objects is usually performed by introducing some errors (distortions). If the distortion exceeds a certain bound, the methods of steganalysis can detect the presence of hidden data. We consider general class of stegosystems based on linear hash functions. The suggested stegosystems allow to transmit secret information of the amount asymptotically close to the maximum possible under a given admissible level of distortion.

Keywords: data hiding, data embedding, embedding rate, linear hash function.

1 Introduction

We consider the problem of steganography which can be formulated as the problem of transmitting messages in such a way that the very fact of transmission be concealed from any observer. To achieve this goal, the messages are embedded in various innocuous objects (digital photos, audio, video, etc.), often called cover objects or covertexts, whose transmission cannot raise any suspicion. So an alternative term for steganography is data hiding. The observer who examines transmitted objects tries to detect the presence of hidden data, which is the main problem of steganalysis. There is a lot of literature on steganography, the basics can be found in, e.g., [1].

It is usually assumed that there are two communicating parties – Alice and Bob. Alice embeds a secret message in a cover object and transmits it to Bob over an open communications channel. Bob receives the object and extracts the message. There is Eve who observes over the channel carries out steganalysis of transmitted objects. It is also usually assumed that secret messages are

encrypted prior to their embedding. So Alice and Bob must agree in advance on a stego key which can be used for encryption and decryption as well as for determining some other details of embedding and extracting algorithms.

There is a class of so-called perfect stegosystems where embedding data does not change the structure and statistical properties of cover objects and, hence, the presence of hidden information cannot be detected. In particular. efficient methods of constructing such systems for covertexts generated by sources with finite memory are suggested in [2].

However, embedding hidden data in digital images, audio and similar "natural" objects that cannot be modeled by finite memory sources is based on introducing some errors (distortion). This distortion makes digital objects "less natural" which is the main hook for steganalysis that permits to detect the presence of hidden data. So the question arises before stegosystem developers, how to construct the methods that would allow to embed maximal amount of information under a given (admissible) level of distortion. Note that this problem cannot be solved "once and for all" since with the progress in steganalysis the level of undetected distortion becomes smaller and smaller.

To explain the essence of the problem, let us consider an example. Let Alice be able to transmit innocuous N-bit messages in which she can change no more than n bits. (It is supposed that Eve can detect the fact of introducing distortion if $n+1$ bits or more have been changed). One of the possible Alice strategies is to select n positions (agreed with Bob) and replace them with the bits of a secret message (in real systems Alice and Bob often select those positions using identical pseudorandom number generators). In this case, Alice can transmit n-bit secret messages and the size of the set of all potentially possible secret messages is 2^n. As we know, the progress in steganalysis restricts the number of distorted bits n to be much less than the covertext size N. The ratio $\alpha = n/N$ is called the embedding rate and, for some covertexts, may be equal to few percent.

On the other hand, if Alice has freedom to select any n positions in the covertext, she has the set of possible messages of size not 2^n but of a rather greater value $\binom{N}{n}$ (more precisely, $\sum_{i=0}^{n}\binom{N}{i}$ as she may change not exactly n bits but any their number up to n bits). Consequently, Alice can potentially transmit secret messages of $\log\sum_{i=0}^{n}\binom{N}{i}$ bits which asymptotically, as N gets large, equals $Nh(\alpha)\,(1+o(1))$, where $h(\alpha) = -(\alpha\log\alpha + (1-\alpha)\log(1-\alpha))$ is the binary Shannon entropy, see [3]. We can see that even for small embedding

rates $\alpha = n/N$, the length of embedded message, asymptotically, can be much greater than n.

The idea of introducing distortion not in fixed n positions of covertext but in "variable" positions determined by the embedded message, was in one form or another realized in stegosystems utilizing error-correcting codes. Such systems were first suggested in [4] and then developed and generalized in [5, 6, 7] and many other works.

We consider a more general class of stegosystems that incorporates, as special cases, the methods essentially equivalent to those based on error-correcting codes. Running a few forward, we note that the stegosystems suggested in this paper allow to transmit secret information of the amount asymptotically close to the maximum possible, i.e. $\log \sum_{i=0}^{n} \binom{N}{i}$.

2 Problem Setting

Proceed to a formal description of the problem considered. Let there be given a stegosystem with the set of covertexts $\hat{C}$ and the set of admissible distortions $\hat{D}$. Denote by $c * d$ a covertext c with introduced distortion d. For instance, $\hat{C}$ may be a set of digital photos in full-color BMP format with resolution 1280×800 pixels (each pixel is encoded by three bytes, representing the intensities of red, green and blue color components (RGB)). Various admissible distortions may be accepted for various stegosystems. For example, in one stegosystem distortions of not more than 1% of least significant bits (LSB) in any of RGB components may be admitted. In this case the set of admissible distortions $\hat{D}$ may be composed of elements represented as three matrices (maps) of zeros and ones, each of size 1280×800, where ones indicate the positions which must be altered by LSB replacement, the share of ones being not greater than 1% in each matrix. An alternative demand may admit distortions of not more than 1% of LSB, as previously, but distortions in adjacent pixels are prohibited. In the third case, the distortions are admitted if in any 10×10 square there is not more than 1 bit changed. And so on. If the images are in JPEG format, different rules establishing the admissible distortions may be applied.

A natural question is related to estimation of the amount of information which can be covertly transmitted in the system $\hat{C}$, $\hat{D}$. Let us call this value

the capacity of the system and denote by γ. Then, evidently,

$$\gamma \leqslant \log |\hat{D}| \,, \tag{1}$$

where, as usually, $|\hat{D}|$ is the number of elements in $\hat{D}$. (Indeed, each distortion corresponds to one hidden message, so the number of words of length γ (which is 2^γ) cannot be greater than the number of admissible distortions $|\hat{D}|$, hence $2^\gamma \leqslant |\hat{D}|$).

3 A Solution Based on Linear Hash Functions

We could see in the examples considered that, in many cases, both covertexts and distortions may be represented as binary words of equal length and the process of applying the distortion d to the covertext c is reduced to bitwise addition modulo 2, i.e. the covertext with introduced distortion may be represented as $w = c \oplus d$. In this section, we describe a stegosystem Λ whose capacity is close to the upper bound (1). For this we need a so-called linear hash function λ defined over the set of words w with values in the set of binary words of a certain length γ_λ (for definition of hash functions and their use in cryptography and computer science in general may be found in many textbooks, see, e.g., [1]). We assume that the function λ, as any ordinary hash function, makes good mixing and for any covertext $c \in \hat{C}$ and distortion $d \in \hat{D}$ the identity is valid

$$\lambda(c \oplus d) = \lambda(c) \oplus \lambda(d) \,. \tag{2}$$

It is worth noting that the required hash function does not need to be cryptographically secure.

Describe now the sequence of actions of Alice and Bob (or the protocol) defining the system Λ.

Let Alice have a covertext $c \in \hat{C}$ and wish to send it to Bob with embedded secret message $s \in \{0,1\}^{\gamma_\lambda}$. To do that Alice computes $u = \lambda(c)$, $v = u \oplus s$, and finds the distortion $d \in \hat{D}$ satisfying the identity $\lambda(d) = v$. Then Alice forms the stegotext $w = c \oplus d$ and sends it to Bob.

Bob, having received the stegotext w, computes $\lambda(w)$.

It occurs that $\lambda(w) = s$! That is, Bob could extract the secret message. More precisely, the following simple yet important theorem is valid:

Theorem 1. *Let the stegosystem Λ be used. If for every word $v \in \{0,1\}^{\gamma_\lambda}$ there exists $d \in \hat{D}$ for which $\lambda(d) = v$, then $\lambda(w) = s$ and the system capacity equals γ_λ.*

Proof. Indeed, from linearity of λ it follows that $\lambda(w) = \lambda(c \oplus d) = \lambda(c) \oplus \lambda(d)$. Due to the system construction $\lambda(d) = v = u \oplus s$ and $\lambda(c) = u$ hence $\lambda(c \oplus d) = u \oplus (u \oplus s) = s$. The theorem is proved. □

Remark 1. *In order to fulfill the condition that $d \in \hat{D}$ for which $\lambda(d) = v$ exists, it is sufficient to require that the values of hash function λ cover entirely the set of γ_λ-bit words, i.e. the identity must hold $\{\lambda(d) : d \in D\} = \{0,1\}^{\gamma_\lambda}$. Then, evidently, for any $v \in \{0,1\}^{\gamma_\lambda}$ such $d \in \hat{D}$ can be found that $\lambda(d) = v$. Notice also that the system capacity equals γ_λ in this case.*

4 Linear Hash Functions over Binary Fields

Complexity, performance, as well as the very existence of the described above stegosystem Λ depends primarily on linear hash function λ which we shall now consider. We consider only one class of such functions based on the abstract mathematical theory of Galois fields which finds wide practical application in the systems of information transmission and storage where it is referred to as cyclic redundancy check (CRC) codes. To describe the considered class of linear hash functions we assume that empty and filled covertext objects as well as distortions are represented by binary words of length N, $N > 0$. Every word $w = w_{N-1}w_{N-2}\dots w_1 w_0 \in \{0,1\}^N$ may be seen as the polynomial

$$w(x) = w_{N-1}x^{N-1} + w_{N-2}x^{N-2} + \dots + w_1 x + w_0 \ . \tag{3}$$

Let m be an integer and

$$g(x) = x^m + g_{m-1}x^{m-1} + g_{m-2}x^{m-2} + \dots + g_1 x + g_0$$

be a degree m polynomial. We define the hash function $\lambda_G(w)$ as the remainder from devision of $w(x)$ by $g(x)$:

$$\lambda_G(w) = w(x) \bmod g(x) \ , \tag{4}$$

using this notation both for polynomial and the word formed by its coefficients. Note immediately that from elementary properties of polynomials it follows that

$\lambda_G(w_1 \oplus w_2) = \lambda_G(w_1) \oplus \lambda_G(w_2)$, i.e. λ_G is a linear hash function.

Denote by Λ_G the described above stegosystem Λ in the case when the hash function λ_G is employed.

It is known that if $g(x)$ is an irreducible polynomial (i.e. the one that cannot be factored), then the set of all possible polynomials $\lambda_G(w)$ constitutes a binary field $\mathbb{F}_{2^m}$ (or a Galois field $GF(2^m)$ in alternative notation) whose definition and main properties can be found in many textbooks, see e.g. [8].

Consider an example. Let the covertexts be binary words of length $N = 2^m - 1$, $m \geqslant 1$, and the admissible distortion be 1 bit (in other words, Alice gets an N-bit word w in which she may change not more than 1 bit for hidden data transmission). Formally, the set of admissible distortions $\hat{D}$ may be represented as $\hat{D} = \{s_0, s_1, \ldots, s_N\}$, where s_i is a word having single 1 at the i-th position and zeros at the remaining positions (s_0 consisting of zeros only).

To construct the stegosystem Λ_G choose a primitive polynomial $g(x)$ of degree m that constitutes a binary field $\mathbb{F}_{2^m}$ and let for a word $w \in \{0,1\}^N$ hash function $\lambda_G(w)$ be defined by identities (3) and (4). (Note that the length of hash values is m bits.)

Proposition 1. *The capacity of the stegosystem Λ_G equals m bits which is the maximum possible value.*

Proof. Recall that $\hat{D} = \{s_0, s_1, \ldots, s_N\}$, hence the capacity of this stegosystem cannot exceed $\log|\hat{D}| = \log(N+1) = m$. Let us show now that the capacity equals m. Indeed, the capacity of stegosystem Λ_G is determined in the remark to Theorem 1. As it follows from the remark, it suffices to show that the values of hash function λ_G are different at all possible distortions $d \in \hat{D}$. We defined the set $\hat{D}$ so that any element s_i contains a single 1 bit in the i-th position (s_0 is all zeros). Consequently, $\lambda_G(s_i) = x^{i-1} \bmod g(x)$ for all $i = 1, 2, \ldots, N$, see (4). By definition, the polynomial $g(x)$ is primitive and the property of binary field dictates that $x^{i-1} \bmod g(x)$ are non-zero and different for all $i = 1, 2, \ldots, N = 2^m - 1$, (they generate $2^m - 1$ different non-zero elements of the field). Hence all $\lambda_G(s_i)$ are different for $i = 1, 2, \ldots, N$ and also differ from $\lambda_G(s_0) = 0$. $\square$

Consider a more specific example. Let $m = 2$ and thus $N = 3$. In this case, the set of admissible distortions $\hat{D} = \{000, 001, 010, 100\}$ (we write the words in big-endian format, i.e. the least significant bit is the rightmost, for ease of association with polynomials). Assume that Alice and Bob choose the primitive

polynomial $g(x) = x^2 + x + 1$. Then the hash function values for the elements of $\hat{D}$ are:

$$\begin{aligned}
\lambda_G(000) &= 00\ , \\
\lambda_G(001) &= 1 \bmod g(x) = 01\ , \\
\lambda_G(010) &= x \bmod g(x) = x = 10\ , \\
\lambda_G(100) &= x^2 \bmod g(x) = x + 1 = 11\ .
\end{aligned}$$

Suppose Alice has the covertext $c = 101$ and wishes to transmit the secret message $s = 11$. By the protocol defining stegosystem Λ, Alice computes $u = \lambda_G(c) = \lambda_G(101) = (x^2 + 1) \bmod (x^2 + x + 1) = x = 10$. Then Alice finds $v = u \oplus s = 10 \oplus 11 = 01$ and determines $d \in \hat{D}$ for which $\lambda_G(d) = 01$: $d = 001$. Alice introduces distortion d in covertext c: $w = 101 \oplus 001 = 100$, and sends it to Bob. Bob, having received the distorted covertext w, computes $\lambda_G(100) = 11$ and gets the secret message $s = 11$.

Remark that with the admissible distortion 1 bit, the considered stegosystem Λ_G has the same capacity as a stegosystem based on Hamming codes, see [7].

5 Potential Capacity of Stegosystems Based on Linear Hash Functions

In this section, we show that "almost any" stegosystem based on linear hash function, generally speaking, has the capacity asymptotically close to the maximum possible. To do this, we go back to considering the general system Λ. In this system, Alice transmits in one covertext object γ_λ bits of secret information which means, by definition, that the capacity equals γ_λ so the question of its evaluation plays an important role. Let us proceed to answering this question.

We start with clarifying the concept of hash function mixing property. Let λ be a function defined over the binary words of length N and taking values in the set of binary words of length m, moreover, $N \geqslant m \geqslant 1$. We refer to this function as *mixing* if for any $v \in \{0,1\}^m$

$$P\{\lambda(w) = v\} = 2^{-m}\ , \tag{5}$$

if different w are picked from the set of words $\{0,1\}^N$ uniformly at random (with equal probabilities).

Assume now that the sets of covertexts $\hat{C}$ and admissible distortions $\hat{D}$ are

given, their elements are represented by binary words of length N, and there is a hash function λ, of which is only known that it possesses the mixing property (5). Let us estimate the capacity γ_λ of this system. First, note that with any (non-zero) value of m and any hash function λ the situation is possible when the values of hash function $\lambda(d)$, $d \in \hat{D}$, do not completely cover the set $\{0,1\}^m$, i.e.

$$\{v : \lambda(d) = v,\ d \in \hat{D}\} \neq \{0,1\}^m\ .$$

Therefore the described system may have some (non-zero) capacity only with certain probability. Obviously, only those systems are practically interesting which have this probability close to 1, say, $1 - 10^{-8}$.

It occurs that asymptotically under any arbitrarily small $\delta > 0$ the capacity γ_λ is close to the maximum possible. More formally, the following holds:

Theorem 2. *Let the stegosystem Λ be defined on the set of covertexts $\hat{C}$, the set of randomly selected admissible distortions $\hat{D}$ and uses a hash function λ which possesses the mixing property (5). Then for large $|\hat{D}|$ and any $\delta > 0$ the inequality*

$$\gamma_\lambda \geqslant \log|\hat{D}| - \log\ln(|\hat{D}|/\delta) \tag{6}$$

holds with probability $1-\delta$ (here log *denotes binary and* ln *natural logarithms).*

Proof. The proof is based on known solutions of the problem of distributing balls into boxes. The problem is formulated as follows. There are M boxes into which K balls are to be distributed uniformly at random, besides, each box can stow an arbitrary number of balls. A random variable μ_0 is defined to be the number of boxes that remain empty after finishing the distribution of balls. It is shown in [9] that

$$E(\mu_0) \leqslant M\mathrm{e}^{-K/M}\ . \tag{7}$$

With respect to the stegosystem Λ, we may consider every word from the set $\{0,1\}^m$ as a box, and the elements of $\hat{D}$ as the balls. Besides, assume that the ball d is placed in the box $v \in \{0,1\}^m-$, if $\lambda(d) = v$. Note that the mixing property (5) ensures uniformity of distribution of balls into boxes. So

$$M = 2^m\ ,\ \ K = |\hat{D}|\ . \tag{8}$$

Random variable μ_0 being not equal to zero, means that the values of hash function $\lambda(d)$ do not cover entirely the set $\{0,1\}^m$ under the given $\hat{D}$. By the condition of the theorem, it is required that the probability of this event be

equal to $1-\delta$, i.e.

$$P\{\mu_0 = 0\} = 1 - \delta \; . \tag{9}$$

Notice now that, by definition,

$$E(\mu_0) = \sum_{j=1}^{\infty} j \times P\{\mu_0 = j\} \; .$$

It is plain that

$$E(\mu_0) \geqslant \sum_{j=1}^{\infty} 1 \times P\{\mu_0 = j\} = 1 - P\{\mu_0 = 0\} \; .$$

From this inequality and (7) we obtain

$$M\mathrm{e}^{-K/M} \geqslant 1 - P\{\mu_0 = 0\} \; ,$$

consequently,

$$P\{\mu_0 = 0\} \geqslant 1 - M\mathrm{e}^{-K/M} \; .$$

By substitution of (9) in the last inequality we obtain

$$1 - \delta \geqslant 1 - M\mathrm{e}^{-K/M} \; .$$

Hence

$$K/M - \ln M \leqslant \ln(1/\delta) \; .$$

Taking into account that the number of boxes M is less than the number of balls K (since, according to the theorem condition, we consider large $K = |\hat{D}|$), from the last inequality we obtain

$$K/M - \ln K \leqslant \ln(1/\delta) \; .$$

By rearranging the last inequality we can see that

$$M \geqslant K/\ln(K/\delta) \; .$$

Considering that this inequality holds with probability $1-\delta$ and Eq. (8) is valid, taking logarithms we obtain

$$\gamma_\lambda \geqslant \log|\hat{D}| - \log\ln(|\hat{D}/\delta)| \; .$$

This completes the proof. □

6 Methods of Constructing Stegosystems Based on Linear Hash Functions

The problem of constructing the system with capacity r close to the maximum γ_λ given by (1) essentially depends on the size of the set $\hat{D}$. If this quantity is small and all elements of $\hat{D}$ may be counted in relatively short time, we can define a hash function λ' whose values may be recorded for all admissible distortions $d \in \hat{D}$. Then the maximal value of r can be found such that certain (e.g. initial) r bits of values of $\lambda'(d)$, $d \in \hat{D}$, cover entirely the set $\{0,1\}^r$. In this case, we can slightly modify the hash function by making its value consisted of those r bits. (It is easy to see that in the modified hash function the property of linearity is preserved.) The capacity of this stegosystem is obviously r bits. If Alice and Bob have enough time, they can exhaustively search through several hash functions to select one with maximal capacity and then use the selected hash function for stegosystem construction.

However, in many situations in practice the set of admissible distortions $\hat{D}$ is of very large size so the exhaustive search through all $d \in \hat{D}$ to find experimentally a hash function of maximal capacity is infeasible. Let, for instance, the placeholders for embedding are the least significant bits of relatively small images, represented as matrices of zeros and ones of dimensions 100×100 and the admissible distortions amount to 2% of symbols. Then the number of admissible distortions equals $\sum_{i=0}^{200} \binom{10000}{i} \gg 2^{100}$ which precludes the exhaustive search.

We consider a well-known approach to solving this problem: divide the covertext in small pieces and use each piece as a separate covertext. Then the total amount of admissible distortions is distributed among the pieces. Thus for the example above the initial covertext of dimensions 100×100 can be divided in 200 equal-size pieces with admissible distortion 1 bit per piece. In this case the stegosystem is greatly simplified and we can use the method based on binary fields. Unfortunately, the capacity of the system with split covertexts may be essentially lower than that of initial system. For the example considered, the potential capacity of initial system is $\log \sum_{i=0}^{200} \binom{10000}{i} \sim 1400$ (see (1)), while the capacity of the system with split covertexts is only $200 \times \lfloor \log 51 \rfloor = 1000$.

Let us derive asymptotic estimates for both stegosystems that allow to judge

about their capacities in general case. Let the covertexts and admissible distortions be the sets of N-bit words and each word of the admissible distortion set contains not more than n ones (in other words, distortions in not more than n symbols of covertext are admitted). So the rate of embedding $\alpha = n/N$. Then from (1) we can see that for any stegosystem Λ its capacity γ_λ does not exceed $\log \sum_{i=0}^{n} \binom{N}{i}$. Using known asymptotic estimates for large N and fixed α, see [3], we may write

$$\gamma_\lambda \leqslant Nh(\alpha)\,(1 + o(1))\ , \tag{10}$$

where $h(\alpha) = -(\alpha \log \alpha + (1-\alpha)\log(1-\alpha))$.

Consider now a simple stegosystem λ^1 where each covertext is divided in n subwords of equal length N/n with admissible distortion 1 bit. We have seen that such a system can be built using binary fields and its capacity $\gamma_\lambda^1 = n \log(N/n)$, see Proposition 1. (We do not count for the necessity of rounding N/n since we are interested in asymptotic estimates.) The last equation can be presented as

$$\gamma_\lambda^1 = N(-\alpha \log \alpha)\,(1 + o(1))\ .$$

Comparing it with (10) we can see that, asymptotically, when implementing the stegosystem λ^1, the loss in capacity is

$$N(-(1-\alpha)\log(1-\alpha))\,(1 + o(1))\ .$$

Upon small α, this value tends to zero hence the loss in implementing the simple stegosystem gets negligible.

Acknowledgments

The authors gratefully acknowledges the support of Russian Foundation for Basic Research, grant no. 18-29-03005.

References

[1] Ryabko B. Ya., Fionov A. N., *Cryptography in Information World*, Goryachaya Liniya – Telekom, Moscow, 2018, 300 pp., in Russian.

[2] Ryabko B. Ya., Ryabko D. B., "Constructing perfect steganographic systems", *Information and Computation*, **209** (2011), 1223 — 1230.

[3] Cover T. M., Thomas J. A., *Elements of Information Theory*, Wiley-Interscience, New York, NY, USA, 2006.

[4] Crandall R., *Some notes on steganography*, Posted on steganography mailing list, 1998, http://dde.binghamton.edu/download/Crandall_matrix.pdf.

[5] Westfeld A., "High capacity despite better steganalysis (F5–A steganographic algorithm)", *LNCS*, Information Hiding. 4th International Workshop, **2137**, eds. I. S. Moskowitz, Springer-Verlag, Berlin, Heidelberg, 2001, 289 — 302.

[6] Galand F., Kabatiansky G., "Information hiding by coverings", IEEE Information Theory Workshop, 2004, 151 — 154.

[7] Bierbrauer J., Fridrich J., "Constructing good covering codes for applications in steganography", *Transactions on data hiding and multimedia security III*, 2008, 1 — 22.

[8] Menezes A., van Oorschot P., Vanstone S., *Handbook of Applied Cryptography*, CRC Press, 1996.

[9] Kolchin V. F., Sevastianov B. A., Chistiakov V. P., *Random Allocations*, distributed solely by Halsted Press, V. H. Winston, Washington, New York, 1978.

Public key cryptography

Application of Non-associative Structures for Construction of Homomorphic Cryptosystems

Sergey Katyshev[1], Andrey Zyazin[1], and Anton Baryshnikov[2]

[1] Russian Technological University (MIREA), Russia
[2] Certification Research Center LLC, Russia
avziazin@mail.ru

Abstract

Homomorphic encoding allows to perform certain mathematical operations with the encoded text and to get the encoded outcome that corresponds to the operations' results processed with a plaintext. There exist both fully homomorphic and partially homographic options (with respect to one or more operations). For practical use of such an encoding it is necessary to have a homomorphism with respect for at least one operation. Using non-associative operations, we construct in this paper an example of a cryptosystem based on the El-Gamal system that is homomorphic with respect to two on-going operations: a group one and a quasi-group one.

Keywords: public-key cryptosystem, homomorphic encryption, non-associative algebraic structures.

1 Introduction

Fully Homomorphic Encryption (FHE) is a specific cryptographic primitive that allows to compute the ciphertexts $Enc(f(m_1, \ldots, m_t))$ without knowing the decryption key and the original input data $m_1, \ldots, m_t$ only with the use of the encrypted data $Enc(m_1), Enc(m_2), \ldots, Enc(m_t)$ and of the arbitrary function f. If the encrypted data $Enc(m_1), Enc(m_2), \ldots, Enc(m_t)$ allows to compute the ciphertext $Enc(f(m_1, \ldots, m_t))$ only for certain functions f, the primitive is called a partially homomorphic encryption.

Cryptographically strong FHE systems may provide new opportunities to maintain information security in such areas like cloud computing, medical and financial data processing because they allow to operate data without its preliminary deciphering in an untrusted environment. The very idea of the FHE was firstly proposed in 1978 in [12] but found no practical implementation.

The first successful attempt to create a FHE system was the system proposed by Goldwasser-Micali in 1982 [7]. However, this cryptosystem was homomorphic only for one operation – modulo 2 addition.

Furthermore, there was a number of systems presented that were homomorphic only in one operation, such as widely-known El-Gamal and RSA cryptosystems. The question on existence of FHE remained unsolved. The first homomorphism was developed by D. Boneh, E.-J. Goh and K. Nissim in the system elaborated in 2005 [5] allowing the computation of any number of additions and one multiplication of the encrypted messages.

An encryption system homomorphic with respect to addition and multiplication in the residue ring was firstly proposed in the paper by Gentry [6] in 2009. Afterwards, there appeared more than a dozen of papers providing alternative options for encrypting systems to be fully homomorphic over residue rings. A detailed overview of these works can be found in [2].

Despite the use of a certain number of the cryptosystems proposed in real-world system, it is already proved that the FHE systems over the residue rings are vulnerable to the known-plaintext attacks [2]. In this regard, a particular interest may be payed to the homomorphic systems based on non-associative structures. Such systems shouldn't be obligatory homomorphic over the residue rings but they still allow to solve a number of practical issues.

It was already proposed in [8] to use some classes of the groupoids for encryption systems which are homomorphic with respect to a unique operation. This paper considers the extension of the El-Gamal [9] cryptosystem to quasi-groups linear over the Abelian group. In this case the resulting system possesses the homomorphic property with respect to two operations at once: group and quasi-group.

2 Basic definitions

Let Enc, Dec be an encrypting and decrypting function of a cryptosystem A. Such a cryptosystem is called *homomorphic with respect to an n-ary operation* $*(a_1, ..., a_n)$ [13] if there exists an effective algorithm M which transforms (for any set of plaintexts $(m_1, ...m_n)$) the input set $(Enc(m_1), ..., Enc(m_n))$ into output C such that

$$Dec(C) = *(m_1, ..., m_n).$$

A particular case of such encryption systems are systems for which

$$Dec(*(Enc(m_1), ..., Enc(m_n))) = *(m_1, ...m_n).$$

In other words, homomorphic encryption allows to produce certain mathematical operations with encrypted texts and obtain an encrypted result that corresponds to the results performed with plaintexts.

Following [10], for an element g of a groupoid $(\Omega, *)$ and given $r, l \in \mathbb{N}$, we define the *right r-th and left l-th powers* respectively by the equalities:

$$g^{[r]} = \underbrace{(...((g * g) * g)...)}_{r\ factors}, \qquad {}^{[l]}g = \underbrace{(...(g * (g * g))...)}_{l\ factors}.$$

We say that g has *commuting right powers*, or that g is a *CRP-element*, if

$$\forall m, n \in \mathbb{N}: \quad g^{[m][n]} = g^{[n][m]}. \tag{1}$$

If this identity is valid for any element $g \in \Omega$, then we say that $(\Omega, *)$ is a *CRP-groupoid*.

Similarly, using the identity

$$\forall m, n \in \mathbb{N}: \quad {}^{[m][n]}g = {}^{[n][m]}g,$$

we define *elements and groupoids with commuting left powers, CLP-elements* and *CLP-groupoids*, respectively.

Let us say that a groupoid $(\Omega, *)$ is a *groupoid with commuting powers* (*CP-groupoid*) if it is a CLP- and CRP-groupoid and moreover, for any $g \in \Omega$ and any $l, r \in \mathbb{N}$, the following equality holds:

$${}^{[l]}(g^{[r]}) = ({}^{[l]}g)^{[r]}.$$

Example. Let $(\Omega, +)$ be an Abelian group. Fixing two commuting automorphisms $\sigma, \tau \in \mathrm{Aut}(\Omega)$ ($\sigma\tau = \tau\sigma$) we define a new operation $*$ on Ω by the following condition:

$$\forall\ x, y \in \Omega \quad x * y = \sigma(x) + \tau(y). \tag{2}$$

So we obtain a groupoid $(\Omega, *)$, which is a quasi-group. Such quasi-groups form a class of *medial quasi-groups* i.e. quasi-groups with the identity $(x*y)*(u*v) = (x * u) * (y * v)$ [1]. According to [10] quasi-group $(\Omega, *)$ is a CP-groupoid.

Some medial quasi-groups have already been used in [8] to create encryption system for homomorphic quasi-group operations $*$.

3 Cryptosystem description

Let $(\Omega, +)$ be an Abelian group, and $(\Omega, *)$ be a medial quasi-group with operation (2). The commuting powers allow to extend El-Gamal cryptosystems on quasi-group $(\Omega, *)$.

Cryptosystem 1.

1. Public key generation. Selecting an element $g \in \Omega$, user A generates an arbitrary (secret) number r_A and computes $g_A = g^{[r_A]}$. The pair (g, g_A) is the public key of user A.

2. Encryption. In order to encrypt a message $m \in \Omega$ user B generates an arbitrary (secret) number r_B and computes $g_B = g^{[r_B]}$. Then he computes $g_A^{[r_B]}$ and $m_{AB} = m + g_A^{[r_B]}$. His ciphertext is the pair (g_B, m_{AB}).

3. Decrypting. In order to decrypt a ciphertext (g_B, m_{AB}) user A computes $g_B^{[r_A]}$ and finds $x = m_{AB} - g_B^{[r_A]}$.

Denote by $AUT(\sigma, t)$ the complexity of calculation of $\sigma^t(a)$ for an arbitrary element a of the semigroup $(\Omega, +)$.

Theorem 1. *The Cryptosystem 1 operates correctly, the complexity of a message encryption is estimated by $O(AUT(\sigma, |\Omega|)\log_2(|\Omega|))$ operations in the group $(\Omega, +)$.*

Proof. Let (g, g_A) be a public key for user A, and (g_B, m_{AB}) be the result of encoding of an arbitrary message $m \in \Omega$ with this key. To decrypt the message user A computes $x = m_{AB} - g_B^{[r_A]} = m + g_A^{[r_B]} - g_B^{[r_A]}$. Due to commutation of the right powers in the quasi-group $(\Omega, *)$, we have

$$g_A^{[r_B]} = g^{[r_A][r_B]} = g_B^{[r_A]}.$$

Therefore, $x = m + g_A^{[r_B]} - g_B^{[r_A]} = m$, and the decoding is accomplished correctly.

The estimation in Theorem 1 follows from the fact that complexity of computation of the element $g^{[k]}$ in the quasi-group $(\Omega, *)$, according to [10], is estimated by $O(AUT(\sigma, k)\log_2(k))$ operations in the group $(\Omega, +)$. □

Remark 1. *Conventionally, automorphisms are defined by the action on generators of the group. In this case, according to [10], the complexity of expo-*

nentiation, and hence the encryption in the Cryptosystem 1 is estimated as $O(\log|\Omega|)$*. If automorphisms* σ*,* τ *have small order in comparison with the order of the group, this valuation equals the estimation of the complexity due to the classical El-Gamal scheme.*

Note that the complexity of revelation of the secret key by an observer having access to the open information g, $g^{[r_A]}$, $g^{[r_B]}$ does not exceed the complexity of the *right discrete logarithm in the groupoid*, i.e. the complexity of solving the equation

$$g^{[x]} = h. \tag{3}$$

Remark 2. *The complexity of the right discrete logarithm in the groupoid depends both on the groupoid structure and on its representation. The solution to the problem within different structures is researched in [10, 11, 3, 4].*

A natural generalization of Cryptosystem 1 consists in combination of right and left powers.

Cryptosystem 2.

1. Public key generation. User A chooses an element $g \in \Omega$, a natural numbers $r \leqslant n$, an ordered set of numbers $a_1, ..., a_n \in \mathbb{N}$, and computes

$$g_A = {}^{[a_1]\dots[a_r]}g^{[a_{r+1}]\dots[a_n]}.$$

User's A public key is the pair (g, g_A).

2. Encryption. In order to encrypt a message $m \in \Omega$, user B generates natural numbers $t \leqslant k$, an ordered set of numbers $b_1, ..., b_k \in \mathbb{N}$, and computes

$$g_B = {}^{[b_1]\dots[b_t]}g^{[b_{t+1}]\dots[b_k]}.$$

Then he computes $g_{AB} = {}^{[b_1]\dots[b_t]}g_A^{[b_{t+1}]\dots[b_k]}$ and $m_{AB} = m + g_{AB}$. The ciphertext is the pair (g_B, m_{AB}).

3. Decrypting. In order to decrypt a ciphertext (g_B, m_{AB}), user A computes $g_{BA} = {}^{[a_1]\dots[a_r]}g_B^{[a_{r+1}]\dots[a_n]}$ and finds $x = m_{AB} - g_{BA}$.

Theorem 2. *The Cryptosystem 2 operates correctly. The complexity of message encryption is estimated by* $O(N \cdot AUT(\sigma, |\Omega|)\log_2(|\Omega|))$ *operations in the group* $(\Omega, +)$*, where* $N = \max(n, k)$*.*

Proof. The same as for the Theorem 1. □

Remark 3. *It is natural to choose n and k not exceeding the logarithm of the size of the set Ω.*

When using the Cryptosystem 2, the recovery of a message by the user from the ciphertext is not harder than the problem of *generalized discrete logarithm* [10], i.e. of finding some pair of positive integers u, v and some set $(x_1, ..., x_v)$ $x_i \in \mathbb{N}$, $i \in \overline{1, v}$, satisfying the equation:

$$^{[x_1]...[x_u]}g^{[x_{u+1}]...[x_v]} = h, \tag{4}$$

if such numbers exist. At the moment, common approaches to a solution of the problem are not described yet. Some approaches are developed in [4].

It is well-known that El-Gamal scheme is homomorphic in regard to group operations. Systems constructed with the help of medial quasi-groups also are homomorphic in regard to group operations. For further we need an additional lemma.

Lemma 1. *For any elements a and b of the medial quasi-group $(\Omega, *)$, and for any natural numbers n and k the following equations are true:*

1. $(a + b)^{[n]} = a^{[n]} + b^{[n]}$,
2. $^{[k]}(a + b) = {}^{[k]}a + {}^{[k]}b$,
3. $^{[k]}(a + b)^{[n]} = {}^{[k]}a^{[n]} + {}^{[k]}b^{[n]}$.

Proof. According to [10], for any element g of the quasi-group $(\Omega, *)$, the following equation is satisfied:

$$g^{[n]} = \sigma^{n-1}(g) + \sigma^{n-2}(\tau(g)) + ... + \sigma(\tau(g)) + \tau(g). \tag{5}$$

If we take $g = a + b$, we get

$$(a + b)^{[n]} = \sigma^{n-1}(a + b) + \sigma^{n-2}(\tau(a + b)) + ... + \sigma(\tau(a + b)) + \tau(a + b).$$

As σ and τ are homomorphisms of the group $(\Omega, +)$, for any natural number k, we have

$$\sigma^k(\tau(a + b)) = \sigma^k(\tau(a) + \tau(b)) = \sigma^k(\tau(a)) + \sigma^k(\tau(b)).$$

Using this equality we get

$$(a+b)^{[n]} = \Big(\sigma^{n-1}(a) + \sigma^{n-2}(\tau(a)) + ... + \tau(a)\Big) + \\ + \Big(\sigma^{n-1}(b) + \sigma^{n-2}(\tau(b)) + ... + \tau(b)\Big).$$

The proof of the first equality is finished with the implementation of the identity (5) for elements a and b.

The second equality is proved in the same way, using the described in [10] identity:

$$^{[k]}g = \tau^{k-1}(g) + \tau^{k-2}(\sigma(g)) + ... + \tau(\sigma(g)) + \sigma(g). \quad (6)$$

The last equality follows from the first two ones. □

Theorem 3. *The cryptosystems 1 and 2 are homomorphic with respect to the group operation $+$.*

Proof. Since Cryptosystem 1 is a particular case of the system 2, we need to consider only the Cryptosystem 2.

Let $Enc(m), Dec(c)$ be the algorithms of the encryption and decryption, defined in the points 2 and 3 of the Cryptosystem 2. Let us check that they are homomorphic with respect to the operation $+$.

Let $c_1 = (g_1, s_1), c_2 = (g_2, s_2)$ be ciphertexts, got as a result of encryption of the messages m_1 and m_2 under a public key $k = (g, g_A)$.

Apply the algorithm of the decryption Dec to $c_1 + c_2 = (g_1 + g_2, s_1 + s_2)$:

$$Dec(c_1 + c_2) = (s_1 + s_2) - {}^{[a_1]...[a_r]}(g_1 + g_2)^{[a_{r+1}]...[a_n]}.$$

Using Lemma 1 several times, we get

$${}^{[a_1]...[a_r]}(g_1 + g_2)^{[a_{r+1}]...[a_n]} = {}^{[a_1]...[a_r]}g_1^{[a_{r+1}]...[a_n]} + {}^{[a_1]...[a_r]}g_2^{[a_{r+1}]...[a_n]}.$$

According to the definition of Dec

$$s_1 - {}^{[a_1]...[a_r]}g_1^{[a_{r+1}]...[a_n]} = Dec(c_1),\ s_2 - {}^{[a_1]...[a_r]}g_2^{[a_{r+1}]...[a_n]} = Dec(c_2).$$

We obtain

$$Dec(c_1 + c_2) = Dec(c_1) + Dec(c_2).$$

□

Let us move to consideration of the quasi-group operation $*$.

Lemma 2. *For any elements $a, b, c, d \in \Omega$ and natural numbers n and k, the following equalities hold:*

$$1.{}^{[k]}(a * b)^{[n]} = {}^{[k]}a^{[n]} * {}^{[k]}b^{[n]},$$
$$2.(a + b) * (c + d) = a * c + b * d.$$

Proof. According to the definition of the operation $*$, $a * b = \sigma(a) + \tau(b)$. Therefore,

$$\sigma(a * b) = \sigma(\sigma(a) + \tau(b)) = \sigma^2(a) + \sigma(\tau(b)) = \sigma^2(a) + \tau(\sigma(b)) = \sigma(a) * \sigma(b).$$

In particular, $\sigma(a^{[k]}) = (\sigma(a))^{[k]}$ and, similarly, $\tau(a^{[k]}) = (\tau(a))^{[k]}$. To obtain the first equality we need to apply Lemma 1 to the elements $\sigma(a)$ and $\tau(b)$.

Let us move to the second equality.

$$(a + b) * (c + d) = \sigma(a + b) + \tau(c + d).$$

As σ and τ are homomorphisms of the group $(\Omega, +)$,

$$\sigma(a + b) = \sigma(a) + \sigma(b), \quad \tau(c + d) = \tau(c) + \tau(d).$$

If we rearrange the summands, we get

$$(a + b) * (c + d) = (\sigma(a) + \tau(c)) + (\sigma(b) + \tau(d)).$$

Using the definition of the operation $*$ once again, we get the required equality. □

Theorem 4. *Cryptosystems 1 and 2 are homomorphic with respect to quasi-group operation $*$.*

Proof. We shall use notations from Theorem 3.

Let's have a look at the element $W = Dec(c_1) * Dec(c_2)$. Using the definition of the operation Dec, we get:

$$W = \left(s_1 - {}^{[a_1]\ldots[a_r]}g_1^{[a_{r+1}]\ldots[a_n]}\right) * \left(s_2 - {}^{[a_1]\ldots[a_r]}g_2^{[a_{r+1}]\ldots[a_n]}\right).$$

Let's apply the second point of the lemma 2:

$$W = \left(s_1 * s_2\right) - \left({}^{[a_1]\ldots[a_r]}g_1^{[a_{r+1}]\ldots[a_n]} * {}^{[a_1]\ldots[a_r]}g_2^{[a_{r+1}]\ldots[a_n]}\right).$$

If we apply the first point from lemma 2 several times, we get:

$$W = \big(s_1 * s_2\big) - \big({}^{[a_1]\dots[a_r]}(g_1 * g_2)^{[a_{r+1}]\dots[a_n]}\big).$$

We see that W coincides with the result of the decryption of the message $c_1 * c_2 = (g_1 * g_2, s_1 * s_2)$. Thus, $Dec(c_1 * c_2) = Dec(c_1) * Dec(c_2)$. □

One can show that the built-up systems are not fully homomorphic. Yet they are homomorphic for many classes of functions.

For example, if we take the cyclic group $(\mathbb{Z}_p, +)$ $(p > 2)$, the identical automorphism σ, and $\tau(x) = -x$, then we can prove that the proposed cryptosystems are homomorphic for an n-ary operation $*(a_1, ..., a_n)$ if and only if the operation $*$ is given by a linear function of $a_1, ..., a_n$ from $(\mathbb{Z}_p, +)$.

References

[1] Belyavskaya G. B., Tabarov A. Kh., "Identities with permutations leading to linearity of quasigroups", *Discr. Math.*, **21**:1 (2009), 36–51, in Russian.

[2] Babenko L. K., Byrtika F.B., Makarevitch O.B., Trapacheva A.V., "A fuuly homomorphic encoding (review)", *Problems of information security*, **3** (2016), 3–25, in Russian.

[3] Baryshnikov A.V., Katyshev S.Yu., "Key agreement schemes based on linear groupoids", *Mat. Vopr. Kriptogr.*, **8**:1 (2017), 7–12.

[4] Baryshnikov A.V., Katyshev S.Yu., "Application of non-associative structures for the construction of public key distribution algorithms", *Mat. Vopr. Kriptogr.*, **9**:4 (2018), 5–30, in Russian.

[5] Boneh D., Goh E.-J., Nissim K., "Evaluating 2-DNF formulas on ciphertexts", *Theory of Cryptography - TCC'05, ser. Lecture Notes in Computer Science. Springer*, **3378** (2005), 325–341.

[6] Gentry C., *A Fully Homomorphic Encryption Scheme*, Ph.D. Thesis. – Stanford Univ., 2009.

[7] Goldwasser S., Micali S., "Probabilistic encryption", *Journal of Computer and System Sciences*, **28**:2 (1984), 270–299.

[8] Gribov A.V. , "Some Homomorphic Cryptosystem Based on Nonassociative Structures", *J Math Sci*, **223**:5 (2017), 581–586.

[9] El-Gamal T., "A public-key cryptosystem and a signature scheme based on discrete logarithms", *IEEE Trans. Inform. Theory*, **4** (1985), 469–472.

[10] Katyshev S. Yu., Markov V. T., Nechaev A. A, "Using non associative groupoids for realization of an open key distribution procedure", *Diskretnaya matematika*, **26**:3 (2014), 45–64, in Russian.

[11] Katyshev S. Yu., "Discrete logarithm problem in finite dimensional algebras over field", *Prikladnaya diskretnaya matematika*, **26**:4 (2014), 21–28, in Russian.

[12] Rivest R., Adleman L., Dertouzos M., "On data banks and privacy homomorphisms", *Foundations of Secure Computation. Academic Press*, 1978, 169–177.

[13] Varnovski N. P., Shokurov A.V., "Homomorphic Encryption", *Papers Of the Institute of System programming*, **12** (2006), 27–36.

Modified Gaudry-Schost Algorithm for the Two-Dimensional Discrete Logarithm Problem

Maksim Nikolaev

Academy of Cryptography of the Russian Federation, Russia
max.abstract@gmail.com

Abstract

In this article we consider the two-dimensional discrete logarithm problem for the subgroup G of an elliptic curve over a finite prime field that has an efficient automorphism of order 6 (for given $P_1, P_2, Q \in G, 0 < N_1, N_2 < \sqrt{|G|}$ find n_1, n_2 such that $Q = n_1P_1 + n_2P_2, -N_1 \leqslant n_1 \leqslant N_1, -N_2 \leqslant n_2 \leqslant N_2$). For this problem, modification of the Gaudry-Schost algorithm is suggested, such that for any $\varepsilon > 0$ there exists an algorithm which average complexity does not exceed $(1+\varepsilon)0.847\sqrt{N} + O_\varepsilon(N^{1/4})$ group operations for $N = 4N_1N_2, N \to \infty$.

Keywords: Gaudry-Schost algorithm, two-dimensional discrete logarithm problem, efficient automorphism.

1 Two-dimensional discrete logarithm problem

Definition 1. *Discrete logarithm problem.*
Given: group $G = \langle P \rangle$, $Q \in G$.
Find: $n \in \{0, \ldots, |G| - 1\}$ *such that* $Q = nP$.

Definition 2. *Two-dimensional discrete logarithm problem.*
Given: group G; $P_1, P_2, Q \in G$, $N_1, N_2 \in \mathbb{N}$, $Q = n_1P_1 + n_2P_2$ *for some (unknown)* $n_1 \in \{-N_1, \ldots, N_1\}$, $n_2 \in \{-N_2, \ldots, N_2\}$.
Find: $n_1', n_2' \in \mathbb{Z}$ *such that* $Q = n_1'P_1 + n_2'P_2$.

Two-dimensional discrete logarithm problem arises in a number of contexts, for example, computation of the group order of the Jacobian of curves [9], analysis of the complexity of solving the discrete logarithm problem for exponents of bounded height [2]. In general case, the Gaudry-Schost algorithm [9] is the most efficient algorithm for solving the two-dimensional discrete logarithm problem.

The basic idea of the algorithm can be described as follows. First, we define the so-called "tame" and "wild" sets:

$$T = \{-N_1, \ldots, N_1\} \times \{-N_2, \ldots, N_2\},$$

$$W = \{-N_1 + n_1, \ldots, N_1 + n_1\} \times \{-N_2 + n_2, \ldots, N_2 + n_2\}.^1$$

Then, we compute in parallel pseudo-random sequences.

$$x_i P_1 + y_i P_2, (x_i, y_i) \in T, i = 1, 2, \ldots, \tag{1}$$

$$Q + z_j P_1 + w_j P_2, (n_1 + z_j, n_2 + w_j) \in W, j = 1, 2, \ldots \tag{2}$$

until we get two identical elements in them:

$$x_k P_1 + y_k P_2 = Q + z_l P_1 + w_l P_2, \tag{3}$$

then we can find $n_1' = x_k - z_l, n_2' = y_k - w_l$.

The average complexity of the Gaudry-Schost algorithm is determined using the following result of Galbraith and Holmes.

Theorem 1. *[6, Theorem 1] Suppose that the following conditions are satisfied.*

1. *We assume that there are C different colours of balls. The j-th ball sampled has probability $r_{j,c}$ of being colour c (independent of all previous selections) where $c \in \{1, \ldots, C\}$.*

$$p_c = \lim_{n \to \infty} n^{-1} \sum_{k=1}^{n} r_{j,c}$$

exists, and $p_1 \geqslant p_2 \geqslant \cdots \geqslant p_C > 0$. Let $b_{n,c} = p_c - n^{-1} \sum_{k=1}^{n} r_{j,c}$. We assume that there is a constant K such that $|b_{n,c}| \leqslant K/n$ for all c.

2. *There are $N' \in \mathbb{N}$ distinct urns. If the j-th ball has color c then the probability that it is put in urn i is $q_{c,i}(N')$ (i.e., independent of previous colour and urn selections and of k). There exists $d > 0$ such that for every $c = 1, \ldots, C$ and $i = 1, \ldots, N'$,*

$$0 \leqslant q_{c,i} \leqslant d/N'.$$

There exist constants $\alpha, \mu > 0$ such that

$$|\{i \in \{1, \ldots, N'\} : q_{1,i}, q_{2,i} \geqslant \mu/N'\}| \geqslant \alpha N'.$$

[1] despite the fact that the n_1, n_2 are unknown, we can choose elements from W, as it will be shown later

Let $Z_{N'}$ be the first time that there are two balls of different colours in the same urn. Then

$$\mathbf{M}(Z_{N'}) = \sqrt{\frac{\pi}{2A_{N'}}} + O(N'^{1/4}),$$

Where

$$A_{N'} = \sum_{c=1}^{C} p_c \left(\sum_{c'=1, c \neq c'}^{C} p_{c'} \left(\sum_{i=1}^{N'} q_{c,i} q_{c',i} \right) \right)$$

and the constant in O depends on $C, p_c, d, K, \alpha, \mu$, but does not depend on N' and $q_{c,i}$.

The average complexity of the Gaudry-Schost algorithm is calculated by Galbraith and Ruprai in [7] and equal $(2.43 + o(1))\sqrt{N}$, where $N = 4N_1N_2$, $o(1) \to 0$, $N_1, N_2 \to \infty$. In the same article an improved version of the algorithm was proposed with average complexity $(2.36 + o(1))\sqrt{N}$.

One can use the automorphism of group G for which the orbit of any element of the group is calculated much faster than a group operation (this automorphism is called efficient) to speed up algorithm for solving the two-dimensional problem. Let G be a cyclic prime-order group and have an efficient automorphism φ acting in the group G as a multiplication by $\lambda \in \{1, \ldots, |G|-1\}$. Then, just as it is done for the classical discrete logarithm problem [5, 14], you can speed up the algorithm if you look not for the same elements of the sequences (1) and (2), but for the same equivalence classes of these elements. Indeed, in this case, instead of equality (3), we have the equality

$$\varphi^s(x_k P_1 + y_k P_2) = Q + z_l P_1 + w_l P_2, \tag{4}$$

for some s, whence

$$Q = (\lambda^s x_k - z_l)P_1 + (\lambda^s y_k - w_l)P_2,$$

i.e. $n'_1 = \lambda^s x_k - z_l, n'_2 = \lambda^s y_k - w_l$.

Efficient automorphisms are widely used in software implementations of cryptographic mechanisms based on elliptic curves [4, 3] in order to obtain speedup of scalar multiplications. The performance gains can be obtained by rewriting of kP as $k_1P + k_2\varphi(P)$, where φ – is an automorphism of an elliptic curve, and $k_1, k_2 \leqslant C_{decomp}\sqrt{|G|}$.

To compute kP for a random $0 < k < |G|$ (GLV[10], GLS [11] methods)

one of the following approaches [1] is used.

1. The parameters $k_1, k_2 \in_R [0, \sqrt{|G|}) \cap \mathbb{Z}$ are selected, and then point $kP = k_1P + k_2\varphi(P)$ is calculated, where $k \equiv k_1 + \lambda k_2 \bmod |G|$. This approach is called *recomposition.*

2. A random parameter k is selected, and then corresponding k_1, k_2 are determined to calculate a point. This approach *decomposition* is more resource intensive.

In case of using recomposition or decomposition, the problem of finding the discrete logarithm k can be reduced to finding values of k_1, k_2 that satisfy

$$kP = k_1P + k_2\varphi(P),$$

i. e. solving the two-dimensional discrete logarithm problem with $P_2 = \varphi(P_1)$.

There are several modifications of Gaudry-Schost algorithm for the cases of elliptic curves with efficient automorphism.

1. In the case of subgroup of an elliptic curve $y^2 = x^3 + Ax + B$ over a finite prime field of $p > 3$ elements (it has an efficient automorphism of order 2) with $N_1 = N_2$ there is algorithm with average complexity $(1 + \varepsilon)1.2533\sqrt{N} + O_\varepsilon(N^{\frac{1}{4}})$ group operations [13], where $N = 4N_1N_2$;

2. In the case of a subgroup of an elliptic curve $y^2 = x^3 + Ax$ over a finite prime field of $p \equiv 1 \bmod 4$ elements with $P_2 = \varphi(P_1)$ and $N_1 = N_2$, where φ is an efficient automorphism of order 4, there is algorithm with average complexity $(1 + \varepsilon)0.8862\sqrt{N} + O_\varepsilon(N^{\frac{1}{4}})$ group operations [13], where $N = 4N_1N_2$;

3. In the case of a subgroup of an elliptic curve $y^2 = x^3 + B$ over a finite prime field of $p \equiv 1 \bmod 3$ elements with $P_2 = \varphi(P_1)$ and $N_1 = N_2$, where φ is an efficient automorphism of order 6, there is algorithm with average complexity $(1+\varepsilon)0.8862\sqrt{N}+O_\varepsilon(N^{\frac{1}{4}})$ group operations [13], where $N = 4N_1N_2$.

In the latter case the automorphism of order 6 of an elliptic curve $y^2 = x^3 + B$ does not give a corresponding performance gain of $\sqrt{6}$ times. The purpose of this article is to obtain an optimized version of Gaudry-Schost algorithm for this case. The main idea of modification of Gaudry-Schost algorithm is to decompose the tame set into smaller ones and use notuniform choice of elements from these subsets.

2 Modified Gaudry-Schost algorithm

The prime-order-q subgroup G of an elliptic curve E defined over $GF(p)$ by the equation $y^2 = x^3 + B$ with $p \equiv 1 \pmod 3$, $q^2 \nmid \#E$ has an efficient automorphism φ of order 6, $\varphi(x, y) = (\beta x, -y)$, where $\beta \neq 1$ – the cube root of 1 modulo p, λ is the root of the equation $\lambda^2 - \lambda + 1 \equiv 0 \pmod q$. For the point $P = (x, y)$, the equivalence class is represented by six elements:

$$\{(x, y), (\beta x, -y), (\beta^2 x, y), (x, -y), (\beta x, y), (\beta^2 x, -y)\}.$$

If $P_2 = \varphi(P_1)$, then

$$\varphi(aP_1 + bP_2) = a(\lambda P_1) + b(\lambda - 1)P_1 = -bP_1 + (a + b)P_2,$$

from where

$$\begin{gathered}\varphi^2(aP_1 + bP_2) = -(a + b)P_1 + aP_2,\\ \varphi^3(aP_1 + bP_2) = -aP_1 - bP_2,\\ \varphi^4(aP_1 + bP_2) = bP_1 - (a + b)P_2,\\ \varphi^5(aP_1 + bP_2) = (a + b)P_1 - aP_2,\end{gathered}$$

i. e. the equivalence class of the point $aP_1 + bP_2$ under the action of the group $\langle\varphi\rangle$ also includes the specified five points. Each such equivalence class corresponds to a set (class) of pairs

$$\begin{aligned}C(a, b) = \{&(a, b), (-b, a + b), (-(a + b), a),\\ &(-a, -b), (b, -(a + b)), (a + b, -a)\}.\end{aligned} \quad (5)$$

Theorem 2. *Let G be a prime-order-q subgroup of an elliptic curve E defined over a finite prime field $GF(p)$ by the equation $y^2 = x^3 + B$ with $p \equiv 1 \pmod 3$, $q^2 \nmid \#E$; φ is an automorphism of the group G, $\varphi(x, y) = (\beta x, -y)$, where $\beta \neq 1$ is the cube root of 1 modulo p; λ is the root of the equation $\lambda^2 - \lambda + 1 \equiv 0 \pmod q$ such that $\varphi(x, y) = \lambda(x, y)$. Then for any $\varepsilon > 0$ there exists an algorithm for solving the two-dimensional discrete logarithm problem in G with average complexity $(1+\varepsilon)0.847\sqrt{N} + O_\varepsilon(N^{\frac{1}{4}})$ group operations (with $N_1 = N_2$, $P_2 = \varphi(P_1)$ and (n_1, n_2), chosen uniformly at random), where $N = 4N_1N_2$, $N \to \infty$.*

Proof. We define the "tame" set T and the set $\widetilde{T}$ of representatives of the classes

T in the classical way:

$$T = \{C(a,b) : -N_1 \leqslant a \leqslant N_1, -N_1 \leqslant b \leqslant N_1\}.$$

$$\widetilde{T} = \{0, \ldots, N_1\} \times \{1, \ldots, N_1\} \cup \{(0,0)\}.$$

We define the sets $T_i^j,\ j = 0, \ldots, 5,\ i = 1, 2$ as follows (graphically represented at the figure 1):

$$T_1^0 = \{(a;b) : 0 \leqslant a \leqslant N_1,\ 1 \leqslant b \leqslant N_1 - a\},$$

$$T_2^0 = \{(a;b) : 0 \leqslant a \leqslant N_1,\ N_1 - a + 1 \leqslant b \leqslant N_1\},$$

$$T_i^j = \varphi(T_i^{j-1}) = \{\varphi(a;b) : (a;b) \in T_i^{j-1}\},\ j = 1, \ldots, 5,\ i = 1, 2.$$

Then

$$T_1^i \cap T_2^j = \varnothing,\ i, j = 0, \ldots, 5,\ i \neq j.$$

$$\widetilde{T} = T_1^0 \cup T_2^0.$$

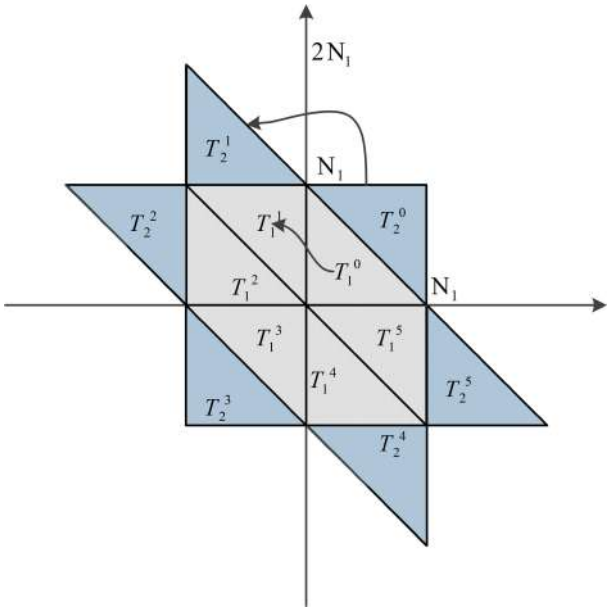


Figure 1: figure
Structure of tame set

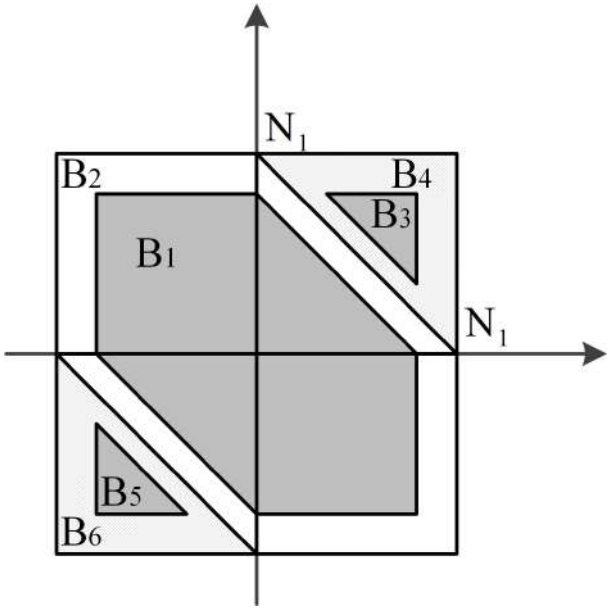


Figure 2: figure
Values of x, y

Denote by T_0 the union of classes from the set T.

Define

$$W_k = \{C(n_1 + a, n_2 + b) : -\frac{kN_1}{2} \leqslant a \leqslant \frac{kN_1}{2}, -\frac{kN_1}{2} \leqslant b \leqslant \frac{kN_1}{2}\}$$

and

$$\widetilde{W}_k = \{-\frac{kN_1}{2} + n_1, \ldots, \frac{kN_1}{2} + n_1\} \times \{-\frac{kN_1}{2} + n_2, \ldots, \frac{kN_1}{2} + n_2\}.$$

We will compute tame and wild sequences as follows

$$x_i P_1 + y_i P_2, \begin{cases} (x_i, y_i) \in T_1^0, & \text{with probability } p_3 \\ (x_i, y_i) \in T_2^0, & \text{with probability } p_4 = 1 - p_3 \end{cases}, \ i = 1, 2, \ldots,$$

$$Q + z_j P_1 + w_j P_2, (z_j, w_j) \in \widetilde{W}_k, j = 1, 2, \ldots$$

Then average complexity measured in group operations does not exceed the total number $Z_{N'}$ of values (x_i, y_i) and (z_j, w_j), chosen before the appearance of (x_l, y_l) and (z_m, w_m) such that $C(x_l, y_l) = C(z_m, w_m)$. Further we will estimate average number of steps of algorithm $Z_{N'}$ as it done in [13]. Under the conditions of the Galbraith-Holmes theorem, we have the following.

$$p_1 = p_2 = \frac{1}{2},$$

$$|T| = |\widetilde{T}| = |T_1^0| + |T_2^0| = \frac{N}{4},$$

$$|T_1^0| = |T_2^0| = \frac{N}{8},$$

$$|\widetilde{W}_k| = k^2 \frac{N}{4}.$$

However, balls of colour 1 (tame) will fall into urns from the sets T_1^0 and T_2^0 with probabilities p_3 and p_4, respectively ($p_3 + p_4 = 1$). Then

$$q_{1,i} = \begin{cases} p_3 \cdot \frac{8}{N}, & \text{if } i \in T_1^0 \\ p_4 \cdot \frac{8}{N}, & \text{if } i \in T_2^0 \\ 0, & \text{otherwise} \end{cases}.$$

Since each class $C(a, b)$ contains no more than 6 elements, $T \cap W_k$ is divided into 6 disjoint subsets $U_j, j = 1, \ldots, 6$, such that each class in U_j gets exactly j elements from $\widetilde{W}_k$, i. e.

$$q_{2,i} = \frac{4j}{N}, i \in U_j.$$

Besides,

$$U = U_1 \cup U_2 = (U \cap T_1^0) \cup (U \cap T_2^0).$$

We get:

$$A_{N'} = \frac{1}{2}(\frac{8p_3}{N}\frac{4}{k^2N}|U_1| + \frac{8p_4}{N}\frac{4}{k^2N}|U_2|) =$$
$$= \frac{16}{k^2N^2}(p_3|U_1| + p_4|U_2|),$$

$$\mathbf{M}(Z_{N'}|(n_1,n_2)) \sim \sqrt{\frac{\pi}{2A_{N'}}} = \sqrt{\frac{\pi k^2N^2}{32(p_3|U_1| + p_4|U_2|)}} =$$
$$= \sqrt{\frac{\pi}{32}}kN\sqrt{\frac{1}{p_3|U_1| + p_4|U_2|}}.$$

Following the articles [8, 12], we set $n_1 = xN_1, n_2 = yN_1, |x|, |y| \leqslant 1$. Let estimate the cardinality of the set U depending on the values of x and y (see figure 2).

1. $(n_1,n_2) \in B_1 = \{(xN_1, yN_1) : \ -1 + \frac{k}{2} \leqslant x \leqslant 1 - \frac{k}{2}, \ \max(-1 - x + k, -1 + \frac{k}{2}) \leqslant y \leqslant \min(1 - x - k, 1 - \frac{k}{2})\}$. Probability of $(n_1,n_2) \in B_1$ is equal $\frac{3}{4} - k + \frac{k^2}{4}$. In this case, the set $\widetilde{W}_k$ is completely contained in $\cup T_1^j, \ j = 0, \ldots 5$, i. e.
$$|U_1| = |\widetilde{W}_k|$$

2. $(n_1,n_2) \in B_2 = \{(xN_1, yN_1) : \ -1 \leqslant x \leqslant 1, \ \max(-1 - x, -1) \leqslant y \leqslant \min(1 - x, 1)\}\backslash B_1$. Probability of $(n_1,n_2) \in B_2$ is equal $k - \frac{k^2}{4}$. In this case we can estimate
$$|U_1| \geqslant \frac{|\widetilde{W}_k|}{4}$$

3. $(n_1,n_2) \in B_3 = \{(xN_1, yN_1) : \ k + \frac{k}{2} \leqslant x \leqslant 1 - \frac{k}{2}, 1 - x + k \leqslant y \leqslant 1 - \frac{k}{2}\}$. Probability of $(n_1,n_2) \in B_3$ is equal $\frac{k^2}{2} - \frac{k}{2} + \frac{1}{8}$. In this case, the set $\widetilde{W}_k$ is completely contained in T_0, i. e.
$$|U_2| = |\widetilde{W}_k|$$

4. $(n_1,n_2) \in B_4 = \{(xN_1, yN_1) : \ 0 \leqslant x \leqslant 1, 1 - x \leqslant y \leqslant 1\}\backslash B_3$. Probability

of $(n_1, n_2) \in B_4$ is equal $-\frac{k^2}{2} + \frac{k}{2}$. In this case we can estimate

$$|U_2| \geqslant \frac{|\widetilde{W}_k|}{8}$$

5. $(n_1, n_2) \in B_5 = \{(xN_1, yN_1) : \ k + \frac{k}{2} \leqslant x \leqslant 1 - \frac{k}{2}, 1 - x + k \leqslant y \leqslant 1 - \frac{k}{2}\}$. Probability of $(n_1, n_2) \in B_3$ is equal $\frac{k^2}{2} - \frac{k}{2} + \frac{1}{8}$. In this case, the set $\widetilde{W}_k$ is completely contained in T_0, i. e.

$$|U_2| = |\widetilde{W}_k|$$

6. $(n_1, n_2) \in B_6 = \{(xN_1, yN_1) : \ 0 \leqslant x \leqslant 1, 1 - x \leqslant y \leqslant 1\} \backslash B_5$. Probability of $(n_1, n_2) \in B_4$ is equal $-\frac{k^2}{2} + \frac{k}{2}$. In this case we can estimate

$$|U_2| \geqslant \frac{|\widetilde{W}_k|}{8}$$

Now we can estimate $\mathbf{M}(Z_{N'})$.

$$\begin{aligned}
&\mathbf{M}(Z_{N'}) = \\
&= \left(\frac{3}{4} - k + \frac{k^2}{4}\right) \mathbf{M}(Z_{N'} | (n_1, n_2) \in B_1) + \left(k - \frac{k^2}{4}\right) \mathbf{M}(Z_{N'} | (n_1, n_2) \in B_2) + \\
&+ 2 \cdot \left(\frac{1}{4} - k + k^2\right) \mathbf{M}(Z_{N'} | (n_1, n_2) \in B_3) + 2 \cdot \left(k - k^2\right) \mathbf{M}(Z_{N'} | (n_1, n_2) \in B_4) \leqslant \\
&\leqslant \left(\frac{3}{4} - k + \frac{k^2}{4}\right) \sqrt{\frac{\pi}{32}} kN \sqrt{\frac{4}{p_3 k^2 N}} + \left(k - \frac{k^2}{4}\right) \sqrt{\frac{\pi}{32}} kN \sqrt{\frac{16}{p_3 k^2 N}} + \\
&+ 2 \cdot \left(\frac{1}{4} - k + k^2\right) \sqrt{\frac{\pi}{32}} kN \sqrt{\frac{4}{p_4 k^2 N}} + 2 \cdot \left(k - k^2\right) \sqrt{\frac{\pi}{32}} kN \sqrt{\frac{16}{p_4 k^2 N}} + \\
&= \sqrt{\frac{\pi N}{32}} \left(\frac{3}{2} - 2k + \frac{k^2}{2} + 4k - k^2\right) \sqrt{\frac{1}{p_3}} + \\
&+ \sqrt{\frac{\pi N}{32}} \left(\frac{1}{2} - 2k + 2k^2 + 4k - 4k^2\right) \sqrt{\frac{1}{p_4}} \\
&= \sqrt{\frac{\pi N}{32}} \left(\left(\frac{3}{2\sqrt{p_3}} + \frac{1}{2\sqrt{p_4}}\right) + 2k \left(\frac{1}{\sqrt{p_3}} + \frac{1}{\sqrt{p_4}}\right) - k^2 \left(\frac{3}{4\sqrt{p_3}} + \frac{2}{\sqrt{p_4}}\right)\right)
\end{aligned}$$

Let

$$\varepsilon(k) = \left(2k\left(\frac{1}{\sqrt{p_3}} + \frac{1}{\sqrt{p_4}}\right) - k^2\left(\frac{3}{4\sqrt{p_3}} + \frac{2}{\sqrt{p_4}}\right)\right)\left(\frac{3}{2\sqrt{p_3}} + \frac{1}{2\sqrt{p_4}}\right)^{-1},$$

then

$$\mathbf{M}(Z_{N'}) \leqslant (1+\varepsilon(k))\frac{1}{8}\sqrt{\frac{\pi}{2}}\left(\frac{3}{\sqrt{p_3}} + \frac{1}{\sqrt{p_4}}\right)\sqrt{N}.$$

The minimum of the function $f(x) = \left(\frac{3}{\sqrt{x}} + \frac{1}{\sqrt{1-x}}\right)$ at $(0;1)$ is reached at $x_0 = \frac{9}{10} - \frac{3\sqrt[3]{3}}{10} + \frac{3^{2/3}}{10} \approx 0.67533$. Then for $p_3 = 0.67533,\ p_4 = 0.32467$

$$\mathbf{M}(Z_{N'}) \leqslant (1+\varepsilon)0.847\sqrt{N} + O_\varepsilon(N^{\frac{1}{4}})$$

□

3 Conclusion

The modification of the Gaudry-Schost algorithm is proposed, which is the most effective for solving the two-dimensional discrete logarithm problem for an elliptic curve $y^2 = x^3 + B$. The performance gain was obtained by non-uniformly selecting elements from various subsets of the Tame set.

References

[1] Aranha D. F., Fouque P., Gérard B., Kammerer J., Tibouchi M., Zapalowicz J.-C., "GLV/GLS Decomposition, Power Analysis, and Attacks on ECDSA Signatures with Single-Bit Nonce Bias", *Advances in Cryptology – ASIACRYPT 2014: 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I*, **17** (2014), 262–281.

[2] Blackburn S.R., Scott S., "The discrete logarithm problem for exponents of bounded height", *LMS Journal of Computation and Mathematics*, **17** (2014), 148–156.

[3] Bos J. W., Costello C., Hisil H., Lauter K., "High-Performance Scalar Multiplication Using 8-Dimensional GLV/GLS Decomposition", *Cryptographic Hardware and Embedded Systems - CHES 2013: 15th International Workshop, Santa Barbara, CA, USA, August 20-23, 2013. Proceedings*, 2013, 331–348.

[4] Costello C., Hisil H., Smith B., "Faster Compact Diffie–Hellman: Endomorphisms on the x-line", *Advances in Cryptology – EUROCRYPT 2014: 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, 2014, 183–200.

[5] Duursma I. M., Gaudry P., Morain F., "Speeding up the discrete log computation on curves with automorphisms", *Lecture Notes in Computer Science*, **1716** (1999), 103–121.

[6] Galbraith S. D., Holmes M., "A non-uniform birthday problem with applications to discrete logarithms", *Discrete Applied Mathematics*, **160**:10-11 (2012), 1547–1560.

[7] Galbraith S. D., Ruprai R. S., "An improvement to the Gaudry-Schost algorithm for multidimensional discrete logarithm problems", *Lecture Notes in Computer Science*, **5921**:10-11 (2009), 368–382.

[8] Galbraith S. D., Ruprai R. S., "Using Equivalence Classes to Accelerate Solving the Discrete Logarithm Problem in a Short Interval", *Lecture Notes in Computer Science*, **6056** (2010), 368–383.

[9] Gaudry P., Schost E., "A low-memory parallel version of Matsuo, Chao and Tsujii's algorithm", *Lecture Notes in Computer Science*, **3076** (2004), 208–222.

[10] Gallant R., Lambert R., Vanstone S., "Faster Point Multiplication on Elliptic Curves with Efficient Endomorphisms", *Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology*, 2001, 190–200.

[11] Galbraith S. D., Lin X., Scott M., "Endomorphisms for Faster Elliptic Curve Cryptography on a Large Class of Curves", *Journal of Cryptology*, **24 3** (2011), 446–469.

[12] Liu W., "Improved algorithms for the 2-dimensional discrete logarithm problem with equivalence classes", 2010, http://www.math.auckland.ac.nz/ sgal018/Wei-Liu-MSc.pdf.

[13] Nikolaev M. V., "On the complexity of two-dimensional discrete logarithm problem in a finite cyclic group with efficient automorphism", *Mathematical Aspects of Cryptography*, **6**:2 (2015), 45–57.

[14] Wiener M. J., Zuccherato R. J., "Faster Attacks on Elliptic Curve Cryptosystems", *Lecture Notes in Computer Science*, **1556** (1999), 190–200.

Division Polynomials for Hyperelliptic Curves Defined by Dickson Polynomials

Ekaterina Malygina and Semyon Novoselov

Immanuel Kant Baltic Federal University, Russia
{emalygina, snovoselov}@kantiana.ru

Abstract

In this paper, we investigate division polynomials for hyperelliptic curves of genus 2 defined by the Dickson polynomial. For the case of $l = 3$, we obtain explicit formulae.

Keywords: hyperelliptic curve, division polynomials, Mumford-Cantor's coordinates, l-torsion, Dickson polynomials.

1 Introduction

Let $\mathbb{F}_q$ be a finite field of size $q = p^n$, where $p > 2$. A hyperelliptic curve C of genus g is a nonsingular curve defined by equation

$$y^2 = f(x),$$

where f is a monic polynomial of degree $2g + 1$ or $2g + 2$.

Hyperelliptic curves were first proposed for use in cryptography by Koblitz [2] for building cryptosystems based on discrete logarithm problem (DLP). At the present time only curves of genus 2 and 3 are considered for building of cryprosystems on DLP due to index-calculus attacks [7].

On the other side, in the post-quantum cryptography on isogenies there is no limitation on genus. The main problem in this field is an absence of effective formulas for a computation of isogenies in general case of degree l. Recently, Flynn and Yan Bo Ti [10] proposed a first post-quantum isogeny-based scheme on genus 2 curves. The authors used the Richelot isogenies for a computing of degree 2 isogenies and the Kummer surfaces for degree 3 case.

In this work, we develop a direct approach for degree 3 case to remove a dependency on the Kummer surface in scheme. It consist of two steps. The first

step is to give an explicit formulas for the division polynomials which describe kernels of degree 3 isogenies. The second step is a computation of an isogeny from the division polynomial. The division polynomials provide explicit formulas for a scalar multiplication in the Jacobian of hyperelliptic curve. Since it is known that $[l] = \psi \circ \hat{\psi}$ for any isogeny ψ of degree l, the computation of an isogeny can be done by factoring or decomposing the division polynomials.

As the first step, in this paper, we give explicit formulas for the division polynomials which describe kernels of degree 3 isogenies. We do this for interesting class of curves defined by the Dickson polynomials.

The division polynomials were first introduced for elliptic curves ($g = 1$) and later were described for hyperelliptic curves by Cantor [4]. The division polynomials are used in Schoof-Pila-like [3] algorithms for counting points on the Jacobian of the curve and in computation of the modular equations [6]. By theorem of Tate [1], point-counting allows us to determine whether two hyperelliptic curves have the isogenous Jacobians or not. Counting points on the Jacobian of curve involves a computation modulo division ideal generated by the division polynomials. Because of that, the degrees and the form of the division polynomials directly affects the complexity of point-counting algorithms.

In this work, we investigate the division polynomials for special classes of curves, which defined by the Dickson polynomials. These classes arise in the decomposition of Jacobians of curves with equation $X : y^2 = x^{2g+1} + ax^{g+1} + bx$.

Theorem 1 ([8]). *If genus g of X is odd then*

$$J_X(\mathbb{F}_q[\sqrt[g]{b}]) \sim J_{X_1}(\mathbb{F}_q[\sqrt[g]{b}]) \times J_{X_2}(\mathbb{F}_q[\sqrt[g]{b}])$$

where

$$X_1 : y^2 = D_g(x, \sqrt[g]{b}) + a$$

and

$$X_2 : y^2 = (x^2 - 4\sqrt[g]{b})(D_g(x, \sqrt[g]{b}) + a).$$

If genus g of X is even then

$$J_X(\mathbb{F}_q[\sqrt[2g]{b}]) \sim J_{X_3}(\mathbb{F}_q[\sqrt[2g]{b}]) \times J_{\tilde{X}_3}(\mathbb{F}_q[\sqrt[2g]{b}])$$

where

$$X_3 : y^2 = (x + 2\sqrt[2g]{b})(D_g(x, \sqrt[g]{b}) + a)$$

and

$$\tilde{X}_3 : y^2 = (x - 2\sqrt[2g]{b})(D_g(x, \sqrt[g]{b}) + a).$$

Here, $D_g(x, \alpha)$ for some constant α denotes the Dickson polynomial of degree g. If $\alpha = 1$, we simply write $D_g(x)$ instead of $D_g(x, 1)$. We refer to [5] for definition and properties of the Dickson polynomials.

Because of the theorem, the computation of the number of points on the $J_X(\mathbb{F}_q)$ can be reduced to the computation of the number of points on $J_{X_1}, J_{X_2}, J_{X_3}, J_{\tilde{X}_3}$.

2 Background and Notations

Let k be a perfect field, $\mathrm{char}(k) \neq 2$ and a hyperelliptic curve C/k of genus g is defined by the equation

$$C : Y^2 = f(X) = \sum_{i=0}^{2g+1} f_i X^i; \quad f_{2g+1} = 1.$$

We denote the l-torsion subgroup of elements from the Jacobian $\mathrm{Jac}_{\bar{k}}(C)$ of curve C by $\mathrm{Jac}_{\bar{k}}(C)[l]$, where l is a prime and $l \neq \mathrm{char}(k)$.

Let τ be a hyperelliptic involution and $\sigma : C/\bar{k} \to \mathrm{Jac}_{\bar{k}}(C)$ be a canonical injection, such that a point P corresponds to a divisor class $[P - \infty]$. Any element of $\mathrm{Jac}_{\bar{k}}(C)$ can be uniquely represented by a divisor $D = \sum_{i=1}^{r} \sigma(P_i)$ and the following holds:

- $P_i \in C(\bar{k})$ and $P_i \neq \infty$;
- $P_i \neq \tau(P_{i'})$ with $i \neq i'$;
- $r \leqslant g$.

Let $P_i = (x_i, y_i) \in C(k)$. Then the Mumford-Cantor's representation of the divisor D has following form

$$D = (d(X), e(X)) = \left(X^r + d_{r-1}X^{r-1} + \ldots + d_0, e_{r-1}X^{r-1} + \ldots + e_0\right),$$

where $d(X) = \prod_{i=1}^{r}(X - x_i)$, $e(x_i) = y_i$, $\deg e(X) < \deg d(X) \leqslant g$ and $d(X)|(e^2(X) - f(X))$.

In fact, all generic non-zero divisors $D \in \mathrm{Jac}_k(C)[l]$ have a weight g. In this work, we will consider the case $g = 2$ and then for the divisor $D = P_1 + P_2 - 2\infty$ we have

$$[l]D = 0 \;\Leftrightarrow\; [l](P_1 - \infty) = -[l](P_2 - \infty).$$

Set $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$. So, the Mumford-Cantor's coordinates for divisors $[l](P_i - \infty)$, $i = 1, 2$, can be represented by the pair of polynomials

$$\left(\delta_l\left(\frac{x_i - X}{4y_i^2}\right), \epsilon_l\left(\frac{x_i - X}{4y_i^2}\right)\right).$$

We remark that $\delta_l\left(\frac{x_i - X}{4y_i^2}\right)$ is not necessarily a monic polynomial. Thus, we need to divide by the leading coefficient to obtain the Mumford-Cantor's representation.

The main result of this paper is explicit formulae for δ_3 and ϵ_3 in the case of genus 2 hyperelliptic curve defined by the Dickson polynomials.

3 Padé Approximation

To find the polynomials δ_l and ϵ_l, we can use an algorithm for group law from [9] for adding divisors in Mumford-Cantor's representation. But there is a more efficient way due to Cantor [4] which use Padé approximation. We specialize this formulas to the case of curves defined by the Dickson polynomials and $l = 3$ and obtain them in explicit form.

Throughout this paper, we assume $g = 2$. We consider a hyperelliptic curve $C/\mathbb{F}_q$ defined by the equation

$$Y^2 = (X \pm 2)(D_4(X, \alpha) + c),$$

where $D_4(X, \alpha) = X^4 - 4X^2\alpha + 2\alpha^2$ is the Dickson polynomial and $\alpha, c \in \mathbb{F}_q$.

Let $P = (x, y) \in C(\mathbb{F}_q)$. We make a following change of variables

$$P = (x, y) \mapsto \tilde{P} = (0, -y).$$

Then by setting

$$X = x - Z, \;\; \tilde{f}(Z) = f(x - Z),$$

the original curve

$$C : Y^2 = X^5 \pm 2X^4 - 4\alpha X^3 \mp 8\alpha X^2 + (2\alpha^2 + c)X \pm (4\alpha^2 + 2c),$$

passing through the point P, is replaced by the curve

$$\tilde{C} : \tilde{Y}^2 = (x - Z)^5 \pm 2(x - Z)^4 - 4\alpha(x - Z)^3 \mp 8\alpha(x - Z)^2 + \\ + (2\alpha^2 + c)(x - Z) \pm (4\alpha^2 + 2c),$$

passing through the point $\tilde{P}$. Denote the right part of the equation of the new curve by $\tilde{f}(Z)$.

Expand the expression $\sqrt{\tilde{f}(Z)}$ in a Taylor series around $Z = 0$:

$$S(Z) := \sqrt{\tilde{f}(Z)} = \sum_{i=1}^{\infty} s_i(x) Z^i.$$

with constant term $s_0 = -y$.

If we assume $m_r = \left\lfloor \frac{r+g}{2} \right\rfloor$ and $n_r = \left\lfloor \frac{r-g-1}{2} \right\rfloor$ with additional conditions $r \geqslant g+1$. Let $A_r(Z)$ and $B_r(Z)$ be non-zero polynomials, such that the formal power series $A_r(Z) - B_r(Z)S(Z)$ is divided by $Z^{m_r+n_r+1}$ and $\deg A_r \leqslant m_r$, $\deg B_r \leqslant n_r$, then a pair (A_r, B_r) is (m_r, n_r)-Padé approximants of series $S(Z)$, namely $\frac{A_r(Z)}{B_r(Z)} = S(Z)$ up to order $m_r + n_r$. So, the solution of Padé approximation problem can be reduced to the finding of polynomials $A_r(Z)$ and $B_r(Z)$.

For next section we need following notations:

$$C_r(Z) = -\frac{A_r(Z) - B_r(Z)S(Z)}{Z^r},$$

$$D_r(Z) = -(A_r(Z) + B_r(Z)S(Z))C_r(Z).$$

Here $C_r(Z)$ is an error value showing how far $\frac{A_r(Z)}{B_r(Z)}$ is from approximating $S(Z)$. The zeros of the polynomial $D_r(Z)$ correspond Z-coordinates of divisor representation $[r](\tilde{P} - \infty)$.

4 Explicit Formulae

In this section we obtain the explicit formulae for the division polynomials ψ_r and as a consequence we obtain explicit formulae for the Mumford-Cantor's

coordinates via coefficients of the series $S(Z)$, the division polynomials ψ_r, the Padé approximants $A_r(Z), B_r(Z)$ and the values $C_r(Z)$, $D_r(Z)$ for $g = 2$.

Let $P = (x, y) \in C(\mathbb{F}_p)$, where

$$y = \pm\sqrt{x^5 \pm 2x^4 - 4\alpha x^3 \mp 8\alpha x^2 + (2\alpha^2 + c)x \pm (4\alpha^2 + 2c)}.$$

and as above we have

$$S(Z) = \sum_{i=1}^{\infty} s_i(x) Z^i.$$

Denote

$$\det(S)_{mn} = \begin{vmatrix} s_{m-n+1} & s_{m-n+2} & \dots & s_m \\ s_{m-n+2} & s_{m-n+3} & \dots & s_{m+1} \\ \vdots & \vdots & \dots & \vdots \\ s_{m-1} & s_m & \dots & s_{m+n-2} \\ s_m & s_{m+1} & \dots & s_{m+n-1} \end{vmatrix}$$

The first four values for the division polynomials are

$$\psi_1 = 0,$$

$$\psi_2 = 1,$$

$$\psi_3 = (2y)^2,$$

$$\psi_4 = (2y)^5 s_3.$$

As above

$$m_r = \left\lfloor \frac{r+g}{2} \right\rfloor, \;\; n_r = \left\lfloor \frac{r-g-1}{2} \right\rfloor.$$

and for $r \geqslant 5$ we can express ψ_r by terms $\det(S)_{m_r n_r}$:

$$\psi_r = (2y)^{\frac{r^2-r-2}{2}} \cdot \det(S)_{m_{r+1} n_{r+1}}.$$

However, the value ψ_r can be calculated by the recursion formula with $s \geqslant r$ [4]:

$$\psi_s \cdot \psi_r \cdot \psi_{s+r} \cdot \psi_{s-r} = \begin{vmatrix} \psi_{s-2} & \psi_{s-1} \cdot \psi_{r+1} & \psi_s \cdot \psi_{r+2} \\ \psi_{s-1} \cdot \psi_{r-1} & \psi_r \psi_s & \psi_{s+1} \cdot \psi_{r+1} \\ \psi_s \cdot \psi_{r-2} & \psi_{s+1} \cdot \psi_{r-1} & \psi_{s+2} \cdot \psi_r \end{vmatrix}$$

Also according to [4] the first non-trivial values for the Padé approximants

A_r, B_r and the error value C_r with $r = 0, \ldots 4$ are

$$A_0(Z) = -1,$$

$$A_1(Z) = -Z, \;\; A_2(Z) = -Z^2,$$

$$A_3(Z) = \sum_{i=0}^{2} s_i Z^i, \;\; A_4(Z) = \sum_{i=0}^{3} s_i Z^i,$$

$$B_0(Z) = 0,$$

$$B_1(Z) = 0, \;\; B_2(Z) = 0,$$

$$B_3(Z) = 1, \;\; B_4(Z) = 1,$$

$$C_0(Z) = 1,$$

$$C_1(Z) = 1, \;\; C_2(Z) = 1,$$

$$C_3(Z) = \sum_{i=3}^{\infty} s_i Z^{i-3}, \;\; C_4(Z) = \sum_{i=4}^{\infty} s_i Z^{i-4},$$

Knowing only ψ_r and the initial values given above, we get recursion formulae for A_r, B_r, C_r with $r \geqslant 5$:

$$A_r(Z) = (2y)^{-r+2} \cdot \frac{\psi_{r-1}}{\psi_{r-2}} \cdot A_{r-1}(Z) - (2y)^{-2r+3} \cdot \frac{\psi_r}{\psi_{r-2}} \cdot A_{r-2}(Z) \cdot Z.$$

$$B_r(Z) = (2y)^{-r+2} \cdot \frac{\psi_{r-1}}{\psi_{r-2}} \cdot B_{r-1}(Z) - (2y)^{-2r+3} \cdot \frac{\psi_r}{\psi_{r-2}} \cdot B_{r-2}(Z) \cdot Z.$$

$$C_r(Z) = \left((2y)^{-r+2} \cdot \frac{\psi_{r-1}}{\psi_{r-2}} \cdot C_{r-1}(Z) - (2y)^{-2r+3} \cdot \frac{\psi_r}{\psi_{r-2}} \cdot C_{r-2}(Z) \right) / Z.$$

By similar way we define the initial values for $D_r(Z)$ with $r = 2, 3, 4$:

$$D_2(Z) = -Z^2,$$

$$D_3(Z) = 2(s_0 s_5 + s_1 s_4 + s_2 s_3) Z^2 + 2(s_0 s_4 + s_1 s_3) Z + 2 s_0 s_3,$$

$$D_4(Z) = 2(s_0 s_6 + s_1 s_5 + s_2 s_4) Z^2 + 2(s_0 s_5 + s_1 s_4) Z + 2 s_0 s_4.$$

And recursion formula for $D_r(Z)$ with $r \geqslant 5$ is

$$D_r(Z) = 2\Bigg(\frac{(2y)^{-2r+4} \cdot \frac{\psi_{r-1}^2}{\psi_{r-2}^2} \cdot A_{r-1}(Z) \cdot C_{r-1}(Z)}{Z} - \\ - \frac{(2y)^{-3r+5} \cdot \frac{\psi_{r-1}\psi_r}{\psi_{r-2}^2} \cdot A_{r-1}(Z) \cdot C_{r-2}(Z)}{Z} + \\ + (2y)^{-4r+6} \cdot \frac{\psi_{r-1}^2}{\psi_{r-2}^2} \cdot A_{r-2}(Z) \cdot C_{r-2}(Z) - \\ - (2y)^{-3r+5} \cdot \frac{\psi_{r-1}\psi_r}{\psi_{r-2}^2} \cdot A_{r-2}(Z) \cdot C_{r-1}(Z) \Bigg) \{i_2\} = \\ = 2(A_r(Z) \cdot C_r(Z))\{i_2\},$$

where a symbol $\{i_2\}$ denotes omitting all terms in branches of degree more than 2.

Finally, we reformulate one of the central Cantor's theorem in our notations:

Theorem 2 ([4]). *If $r \geqslant 3$ then element $[r](\tilde{P} - \infty)$ in the Jacobian $\mathrm{Jac}_{\bar{k}}(\tilde{C})$ of the curve $\tilde{C}$ can be represented by the pair $(D_r(Z), E_r(Z))$, where*

$$E_r(Z) = \\ = 2y \cdot \frac{\psi_{r-1}\psi_{r+1}}{\psi_r^2} \cdot Z \cdot \left(\frac{(2y)^{r^2+r-2} D_{r+1}(Z)}{\psi_{r+1}^2} - \frac{(2y)^{r^2-3r} D_{r-1}(Z)}{\psi_{r-1}^2} \right) \pmod{D_r(Z)}.$$

Returning to the original curve C and divisor $[r](P - \infty)$ with $r \geqslant 3$, the Mumford-Cantor's coordinates are the polynomials of following form:

$$\delta_r(Z) = (2y)^{r^2-r-2} \cdot D_r(4y^2 Z),$$

$$\epsilon_r(Z) = \frac{y \cdot (\psi_{r-1}^2 \cdot \delta_{r+1}(Z) - \psi_{r+1}^2 \cdot \delta_{r-1}(Z)) \cdot Z}{\psi_{r-1} \cdot \psi_r^2 \cdot \psi_{r+1}} \pmod{\delta_r(Z)}.$$

4.1 Case of $l = 3$

Consider a hyperelliptic curve $C/\mathbb{F}_p$ of genus $g = 2$ with the equation

$$Y^2 = \\ = (X-2)(D_4(X, \alpha)+c) = X^5 - 2X^4 - 4\alpha X^3 + 8\alpha X^2 + (2\alpha^2 + c)X - 4\alpha^2 - 2c,$$

where $D_4(X,\alpha) = X^4 - 4\alpha X^2 + 2\alpha^2$ is the Dickson polynomial. Set $\alpha = 1$, then our equation has a form:

$$Y^2 = X^5 - 2X^4 - 4X^3 + 8X^2 + (c+2)X + (-2c-4).$$

For the divisor $D = P_1 + P_2 - 2\infty$ with points $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$, we have a relation

$$[l]D = 0 \Leftrightarrow [l](P_1 - \infty) = -[l](P_2 - \infty).$$

For $l = 3$ we denote the Mumford-Cantor's representation as

$$[3](P_1 - \infty) = \left(\delta_3\left(\frac{x_1 - X}{4y_1^2}\right), \epsilon_3\left(\frac{x_1 - X}{4y_1^2}\right)\right),$$

$$[3](P_2 - \infty) = \left(\delta_3\left(\frac{x_2 - X}{4y_2^2}\right), \epsilon_3\left(\frac{x_2 - X}{4y_2^2}\right)\right).$$

For simplicity we set

$$[3](P_1 - \infty) = \\ = \left(d_2(x_1,y_1)X^2 + d_1(x_1,y_1)X + d_0(x_1,y_1), e_1(x_1,y_1)X + e_0(x_1,y_1)\right),$$

$$[3](P_2 - \infty) = \\ = \left(d_2(x_2,y_2)X^2 + d_1(x_2,y_2)X + d_0(x_2,y_2), e_1(x_2,y_2)X + e_0(x_2,y_2)\right),$$

where d_2, d_1, d_0, e_1, e_0 are rational functions of variables x_i, y_i, $i = 1, 2$.

$$
\begin{aligned}
d_2 = -\frac{1}{4y^4}\Big(& 64x^{20} - 512x^{19} + 512x^{18} + 6144x^{17} + (256c - 16896)x^{16} + (-2048c - 20480)x^{15} + \\
& + (120832 + 3072c)x^{14} + (-32768 + 16384c)x^{13} + (-391680 - 55808c + 384c^2)x^{12} + \\
& + (-3072c^2 - 12288c + 380928)x^{11} + (6144c^2 + 253952c + 614400)x^{10} + \\
& + (12288c^2 - 212992c - 999424)x^9 + (-439296c + 256c^3 - 59904c^2 - 374784)x^8 + \\
& + (1228800 + 36864c^2 + 696320c - 2048c^3)x^7 + \\
& + (129024c^2 + 5120c^3 + 192512c - 90112)x^6 + \\
& + (-786432 - 786432c - 196608c^2)x^5 + \\
& + (-19968c^3 - 23040c^2 + 230400 + 64c^4 + 149504c)x^4 + \\
& + (253952 - 512c^4 + 28672c^3 + 184320c^2 + 376832c)x^3 + \\
& + (-4096c^3 - 106496 - 147456c + 1536c^4 - 61440c^2)x^2 + \\
& + (-65536c - 49152c^2 - 2048c^4 - 16384c^3 - 32768)x + \\
& + 16384 + 8192c^3 + 1024c^4 + 24576c^2 + 32768c \Big),
\end{aligned}
$$

$$
\begin{aligned}
d_1 = -\frac{1}{4y^4}\Big(& -128x^{21} + 1024x^{20} - 1024x^{19} - 12288x^{18} + (33792 - 512c)x^{17} + \\
& + (4096c + 95y^2 + 40960)x^{16} + (-608y^2 - 241664 - 6144c)x^{15} + \\
& + (144y^2 + 65536 - 32768c)x^{14} + (783360 - 768c^2 + 111616c + 6464y^2)x^{13} + \\
& + (24576c - 761856 + 6144c^2 + 60cy^2 - 9960y^2)x^{12} + \\
& + (-1228800 - 12288c^2 - 26688y^2 - 507904c + 96cy^2)x^{11} + \\
& + (70240y^2 - 3408cy^2 + 425984c - 24576c^2 + 1998848)x^{10} + \\
& + (25216y^2 + 119808c^2 + 878592c + 11072cy^2 - 512c^3 + 749568)x^9 + \\
& + (90c^2y^2 + 552cy^2 + 4096c^3 - 2457600 - 192024y^2 - 73728c^2 - 1392640c)x^8 + \\
& + (-59520cy^2 - 10240c^3 + 180224 - 258048c^2 - 385024c + 104320y^2 + 480c^2y^2)x^7 + \\
& + (142528y^2 + 393216c^2 + 98240cy^2 + 1572864c + 1572864 - 6992c^2y^2)x^6 + \\
& + (-128c^4 + 23232c^2y^2 - 460800 + 39936c^3 - \\
& - 134400y^2 - 20736cy^2 - 299008c + 46080c^2)x^5 + \\
& + (-368640c^2 + 380c^3y^2 + 1024c^4 - 507904 - 69552cy^2 - 753664c - \\
& - 15776y^2 - 57344c^3 - 30072c^2y^2)x^4 + \\
& + (39296cy^2 - 3072c^4 + 122880c^2 - 2272c^3y^2 + 8192c^3 + 294912c + \\
& + 32000y^2 + 212992 + 7104c^2y^2)x^3 + \\
& + (65536 + 4096c^4 + 32768c^3 + 131072c + 4752c^3y^2 + \\
& + 7872cy^2 + 16224c^2y^2 + 98304c^2 - 11136y^2)x^2 + \\
& + (-65536c - 11136c^2y^2 + 2304cy^2 - 16384c^3 - \\
& - 2048c^4 - 32768 - 49152c^2 - 3904c^3y^2 + 17920y^2)x + \\
& + 888c^3y^2 - c^4y^2 - 9232y^2 - 5664cy^2 + 1256c^2y^2\Big)
\end{aligned}
$$

$$
\begin{aligned}
d_0 = -\frac{1}{4y^4}\Big(&64x^{22} - 512x^{21} + 512x^{20} + 6144x^{19} + (-16896 + 256c)x^{18} + \\
&+ (-95y^2 - 20480 - 2048c)x^{17} + \\
&+ (120832 + 608y^2 + 3072c)x^{16} + (-32768 - 144y^2 + 16384c)x^{15} + \\
&+ (-391680 - 55808c - 6464y^2 + 384c^2)x^{14} + \\
&+ (-60cy^2 + 9960y^2 - 12288c - 3072c^2 + 380928)x^{13} + \\
&+ (6144c^2 + 40y^4 + 253952c + 26688y^2 - 96cy^2 + 614400)x^{12} + \\
&+ (-192y^4 - 999424 - 70240y^2 + 12288c^2 + 3408cy^2 - 212992c)x^{11} + \\
&+ (-11072cy^2 - 374784 + 256c^3 - 439296c - 160y^4 - 25216y^2 - 59904c^2)x^{10} + \\
&+ (-90c^2y^2 + 1920y^4 + 696320c - 2048c^3 - 552cy^2 + 36864c^2 + 192024y^2 + 1228800)x^9 + \\
&+ (5120c^3 + 192512c - 480c^2y^2 + 59520cy^2 - 40cy^4 - \\
&- 90112 + 129024c^2 - 720y^4 - 104320y^2)x^8 + \\
&+ (-786432 - 98240cy^2 - 8960y^4 - 196608c^2 - \\
&- 142528y^2 + 640cy^4 - 786432c + 6992c^2y^2)x^7 + \\
&+ (-19968c^3 + 149504c + 20736cy^2 - 23232c^2y^2 + 230400 - 23040c^2 + \\
&+ 64c^4 + 12672y^4 - 3136cy^4 + 134400y^2)x^6 + \\
&+ (376832c + 253952 + 184320c^2 - 380c^3y^2 + 15776y^2 + 2560y^4 + 69552cy^2 + \\
&+ 28672c^3 + 6400cy^4 - 512c^4 + 30072c^2y^2)x^5 + \\
&+ (2272c^3y^2 - 106496 - 147456c - 61440c^2 - 32000y^2 - 9760y^4 + 440c^2y^4 - \\
&- 39296cy^2 - 7104c^2y^2 - 4000cy^4 + 1536c^4 - 4096c^3)x^4 + \\
&+ (-2240c^2y^4 - 16384c^3 - 4752c^3y^2 - 49152c^2 - 32768 + 11136y^2 - 16224c^2y^2 - \\
&- 3840cy^4 + 1280y^4 - 2048c^4 - 65536c - 7872cy^2)x^3 + \\
&+ (1024c^4 + 11136c^2y^2 - 2304cy^2 + 32768c + 16384 + 24576c^2 + 3680c^2y^4 + \\
&+ 8192c^3 - 17920y^2 + 4480cy^4 - 5760y^4 + 3904c^3y^2)x^2 + \\
&+ (-1664c^2y^4 + 1536cy^4 + c^4y^2 + 5664cy^2 + 9232y^2 + 9728y^4 - 888c^3y^2 - 1256c^2y^2)x - \\
&- 1952cy^4 + 8c^3y^4 - 464c^2y^4 - 1984y^4\Big),
\end{aligned}
$$

$$
\begin{aligned}
e_1 = \frac{1}{(2y)^9}\Big(& 145x^{24} - 1392x^{23} + 2472x^{22} + 16288x^{21} + (-1626c - 59460)x^{20} + \\
& + (20880c - 89952)x^{19} + \\
& + (689296 - 96568c)x^{18} + (116064c - 459072)x^{17} + \\
& + (-5649c^2 + 597564c - 3226692)x^{16} + \\
& + (81568c^2 - 2566528c + 7090816)x^{15} + (-456816c^2 + 3031872c + 659520)x^{14} + \\
& + (1116480c^2 + 3700992c - 20189952)x^{13} + \\
& + (1684c^3 - 194792c^2 - 15568016c + 25595680)x^{12} + \\
& + (13344c^3 - 5577792c^2 + 18238848c + 2370816)x^{11} + \\
& + (-274800c^3 + 13458528c^2 - 5592384c - 38119296)x^{10} + \\
& + (1462208c^3 - 11625856c^2 - 8844032c + 34221568)x^9 + \\
& + 9(c+2)(999c^3 - 413782c^2 + 505556c + 757304)x^8 + \\
& + (-48(c+2)(1889c^3 - 98362c^2 - 70260c + 481032))x^7 + \\
& + 8(c+2)(45865c^3 - 229418c^2 - 450196c + 1301704)x^6 + \\
& + (-96(c+2)^2(7693c^2 + 8724c - 52236))x^5 + \\
& + (-6(c+2)^2(111c^3 - 115790c^2 - 250476c + 978904))x^4 + \\
& + 16(c+2)^2(277c^3 - 4298c^2 - 33828c + 50056)x^3 + \\
& + (-24(c+2)^3(457c^2 + 16676c - 38108))x^2 + \\
& + 96(c+2)^3(125c^2 + 3188c - 6412)x + \\
& + (c+2)^3(c^3 - 4922c^2 - 72948c + 135944)\Big)
\end{aligned}
$$

$$
\begin{aligned}
e_0 = \frac{1}{(2y^9)}\Big(& -145x^2 51392x^{24} - 2472x^{23} - 16288x^{22} + (1626c + 59460)x^{21} + \\
& + (88y^2 - 20880c + 89952)x^{20} + (-704y^2 + 96568c - 689296)x^{19} + \\
& + (480y^2 - 116064 * c + 459072)x^{18} + \\
& + (9600y^2 + 5649c^2 - 597564c + 3226692)x^{17} + \\
& + (-1448cy^2 - 17488y^2 - 81568c^2 + 2566528c - 7090816)x^{16} + \\
& + (17664cy^2 - 87552y^2 + 456816c^2 - 3031872c - 659520)x^{15} + \\
& + (-80512cy^2 + 333568y^2 - 1116480c^2 - 3700992c + 20189952)x^{14} + \\
& + (133632cy^2 - 138240y^2 - 1684c^3 + 194792c^2 + 15568016c - 25595680)x^{13} + \\
& + 16(143c^2y^2 + 10828cy^2 - 69156y^2 - 834c^3 + 348612c^2 - 1139928c - 148176)x^{12} + \\
& + (-6272c^2y^2 - 1120768cy^2 + 2223616y^2 + 274800c^3 - \\
& - 13458528c^2 + 5592384c + 38119296)x^{11} + \\
& + (-64(1243c^2y^2 - 28852cy^2 + 21932y^2 + 22847c^3 - 181654c^2 - 138188c + 534712))x^{10} + \\
& + (535808c^2y^2 - 961536cy^2 - 822272y^2 - 8991c^4 + 3706056c^3 + \\
& + 2898072c^2 - 15915744c - 13631472)x^9 + \\
& + 16(707c^3y^2 - 83358c^2y^2 - 41500cy^2 + 206936y^2 + 5667c^4 - 283752c^3 - 800952c^2 + \\
& + 1021536c + 2886192)x^8 + \\
& + (-8(c+2)(12640c^2y^2 - 196736cy^2 + 289152y^2 + \\
& + 45865c^3 - 229418c^2 - 450196c + 1301704))x^7 + \\
& + 32(c+2)(11084c^2y^2 - 17520cy^2 + 16880y^2 + 23079c^3 + 72330c^2 - 104364c - 313416)x^6 + \\
& + (-6(c+2)(97536c^2y^2 + 56320cy^2 - 408576y^2 - 111c^4 + 115568c^3 + \\
& + 482056c^2 - 477952c - 1957808))x^5 + \\
& + (-8(c+2)(89c^3y^2 - 44618c^2y^2 - 17748cy^2 + 274760y^2 + \\
& + 554c^4 - 7488c^3 - 84848c^2 - 35200c + 200224))x^4 + \\
& + 8(c+2)^2(680c^2y^2 + 23456cy^2 - 37728y^2 + 1371c^3 + 52770c^2 - 14268c - 228648)x^3 + \\
& + (-32(c+2)^2(481c^2y^2 + 11012cy^2 - 24956y^2 + 375c^3 + 10314c^2 - 108c - 38472))x^2 + \\
& + (c+2)^2(18816c^2y^2 + 124416cy^2 - 350720y^2 - c^4 + \\
& + 4920c^3 + 82792c^2 + 9952c - 271888)x + \\
& + (-8(c+2)^3(c^2 + 1028c - 2044))y^2\Big).
\end{aligned}
$$

Conclusion

In this paper, we obtained the explicit formulae for 3-division polynomial for class of curves, defined by the Dickson polynomial. This gives an explicit

description of 3-torsion in the Jacobian of the curve and this allow us to find the points of order 3. Therefore, we can find $\#J_C(\mathbb{F}_q) \pmod 3$.

In further works, we will use this formulae to describe $(3,3)$-isogenies for our class of curves with applications to scheme of Flynn and Yan Bo Ti.

Acknowledgments

Semyon Novoselov gratefully acknowledges the support of Russian Foundation for Basic Research, research project №18-31-00244.

References

[1] Tate J., "Endomorphisms of abelian varieties over finite fields", *Inventiones mathematicae*, **2**:2 (1966), 134—144..

[2] Koblitz N., "Hyperelliptic cryptosystems", *Journal of cryptology*, **1**:3 (1989), 139—150.

[3] Pila J., "Frobenius maps of abelian varieties and finding roots of unity in finite fields", *Mathematics of Computation*, **55**:192 (1990), 745—763.

[4] Cantor D. G., "On the analogue of the division polynomials for hyperelliptic curves", *Journal fur die reine und angewandte Mathematik*, **447** (1994), 91—146.

[5] Lidl R., Mullen G. L., Turnwald G., *Dickson polynomials*, **65**, Chapman & Hall/CRC, 1993.

[6] Gaudry P., Schost É., "Modular equations for hyperelliptic curves", *Mathematics of Computation*, **74**:249 (2005), 429—454.

[7] Gaudry P., Thomé E., Thériault N., Diem C., "A double large prime variation for small genus hyperelliptic index calculus", *Mathematics of Computation*, **76**:257 (2007), 475—492.

[8] Novoselov S.A., "Counting points on hyperelliptic curves of type $y^2 = x^{2g+1} + ax^{g+1} + bx$", 2019, arXiv: 1902.05992.

[9] Cantor D. G., "Computing in the Jacobian of a hyperelliptic curve", *Mathematics of Computation*, **48**:177 (1987), 95—101.

[10] Flynn E.V., Yan Bo Ti, *Genus Two Isogeny Cryptography*, Cryptology ePrint Archive, Report 2019/177, https://eprint.iacr.org/2019/177.