6th Workshop on Current Trends in Cryptology (CTCrypt 2017)

June 5-7, 2017, Saint Petersburg, Repino, Russia.
Pre-proceedings

In cooperation

Partners/support

Program and Table Of Contents

Steering committee

Vladimir Sachkov	Academy of Cryptography, Russia
Alexey Kachalin	TC 26, Russia
Mikhail Glukhov	Academy of Cryptography, Russia
Andrei Zubkov	Steklov Mathematical Institute
Dmitry Matyukhin	TC 26, Russia
Andrei Pichkur	Educational and Methodical Association of Higher Educational Institutions of Russia on Education in Information Security, Russia
Aleksandr Shoitov	Academy of Cryptography, Russia

Program Co-chairs

Andrei Zubkov	Steklov Mathematical Institute
Dmitry Matyukhin	TC 26, Russia
Aleksandr Shoitov	Academy of Cryptography, Russia

Program Committee

Sergey Agievich	Research Institute for Applied Problems of Mathematics and Informatics, Belarus
Sergey Bezzateev	State University of Aerospace Instrumentation, Russia
Sergey Checheta	Educational and Methodical Association of Higher Educational Institutions of Russia on Education in Information Security, Russia
Claus Diem	Leipzig University, Germany
Yury Kharin	Research Institute for Applied Problems of Mathematics and Informatics, Belarus
Alla Levina	ITMO Unversity, Russia
Grigory Marshalko	TC 26, Russia
Pascal Paillier	Cryptoexperts, France
Eduard Primenko	Lomonosov Moscow State University, Russia
Markku-Juhani Olavi Saarinen	Dark Matter, UAE
Igor Semaev	University of Bergen, Norway
Vasily Shishkin	TC 26, Russia
Stanislav Smyshlyaev	TC 26, Russia
Natalya Tokareva	Sobolev Institute of Mathematics, Russia
Mikhail Tuzhilin	TC 26, Russia
Amr Youssef	Concordia University, Canada
Xiaoyun Wang	Tsinghua University, China

External Reviewers

Sergey Grebnev
Vasily Nikolaev
Grigory Karpunin
Alexey Ivanov
Ahmed Abdelkhalek
Mohamed Tolba
Viktor Markov
Dmitry Zhdanovich
Steven Galbraith
Evgeny Alekseev
Rinat Shakirov
Vladimir Drelikhov
Dmitry Pilshikov
Vladimir Mironkin

Dear colleagues!

Scientists in the field of cryptography, specialists of cryptographic information security, software manufacturers, representatives of state regulatory bodies and people interested in information security problems are going take part in the 6th Workshop on Current Trends in Cryptology.

Just like at the previous workshops this year the organizers put scientific quality and novelty of reports presented at the Workshop in the forefront. The reports you will hear today have gone through a strict review of an international program committee. There are indeed some extremely interesting and relevant research results. Workshop reports are devoted to different aspects of cryptology. There are some theoretic problems the practical importance of which is still to be implemented, as well as some issues directly related to the development and use of specific cryptographic algorithms.

Annually we invite foreign specialists, which conduct relevant research in the field of information security, to take part in the workshop. This year our guests are Jean-Philippe Aumasson and Gregor Leander – these researchers have done a lot for the development of modern cryptography.

"A day of practical cryptography" conducted under the aegis of Russian Technical Committee for Standardization TC 26 is worth mentioning. During the Day some problems of development and operation of Cryptographic information protection devices in Russia will be considered.

At the round table "Quantum cryptography: from theory to practice" questions and problems of quantum key distribution systems use are planned to be discussed.

The theoretical side of quantum cryptography has been covered within the Workshop for two years. Now we have approached the practical side of the issue. To assess the practical achievements of theoretical results comprehensively we have invited the representatives of three scientific schools which are engaged in this research trend: Lomonosov Moscow State University, Russian quantum center and ITMO University in St. Petersburg.

Another discussion will touch upon an actively debated blockchain technology which is considered by some specialists to be the future of digital

technologies. The "Blockchain: market expectations and experts' opinions" round table will be devoted to the problems of applying this technology and modern challenges for scientific community and developers to put it into practice.

Finally, the workshop delegates will hear an extraordinary report based on archival records: "Historical evidences on the origin and the establishment of the Russian cryptography (16-17 centuries)".

Dear colleagues!

In conclusion, I would like to say a few words about the Russian Academy of Cryptography, one of the organizers of all six CTCrypt workshops. Today the Academy celebrates its 25th anniversary with all Russian cryptographers. The Academy is a public academic institution carrying out fundamental and applied researches in the field of cryptography and allied sciences. The main research line and scientific assessment of the result are executed with regard to development trends and achievements of modern cryptography alongside with constant communication with the cryptographic community.

Math articles and monographs of the Academy members on algebra, probability theory, combinatorial analysis, and coding theory are published by the Russian and foreign publishers. Results of the researches of Russian mathematicians and cryptographers, including ones of the Academy of Cryptography, can be found in the "Mathematical aspects of cryptography" journal published by the Academy along with V.A. Steklov Mathematical Institute. The proceedings of CTCrypt workshop have been published in this journal since 2013.

Dear colleagues, I hope everyone will find something useful and new in this workshop while the delivered reports will give ground for new ideas and developments. At this point let me wish you efficient work, good impression, new ideas and the 6th Workshop on Current Trends in Cryptology is declared open.

Vladimir Sachkov

On cryptographic properties of the CVV and PVV parameters generation procedures in payment systems

Liliya Ahmetzyanova, Evgeny Alekseev,
Grigory Karpunin, Stanislav Smyshlyaev

Abstract

Two important mechanisms of providing security of payment systems are checks made with parameters CVV and PVV. In the current paper the provable security approach is exploited to explain the reasons of not using the two-pass decimalization procedure in the «MIR» payment system, used by VISA, for example. We propose a simple procedure that turns out to be much more secure.

The work was supported by the Russian Foundation of Basic Research, the project 16-01-00226 A.

Keywords: CVV, PVV, provable security, decimalization

1 Introduction

Cryptographic mechanisms to ensure transactions security in the Russian national payment system «MIR» are being developed. Such mechanisms include procedures for calculating the card check values and PIN-codes (CVV and PVV respectively).

The most common payment systems VISA and Mastercard use a scheme that is based upon consistent application of well-known CBCMAC [1] scheme and so-called two-pass decimalization function, see, for example, [3], for procedures of calculating CVV and PVV. For the payment system «MIR» analogues of this procedures based on Russian national cryptographic standards, in particular block cipher Magma (GOST R 34.12–2015 [6]), are intended to be used.

In the current paper we describe probability distribution on the set of strings of decimal digits, that is induced by the procedure of two-pass deci-

malization. A simple analogue of such a procedure is described — the probability distribution with it turns out to be much closer to the uniform.

In the current paper we define adversary models that are interesting from the practically-oriented point of view. We also define stronger adversary models that are traditionally used in theoretical cryptology. The security in practice-oriented models is evaluated with bounds obtained in these stronger models. We show that for the two-pass decimalization the specter of values of the resulting parameters for which security bounds can be obtained is much more poor than for the proposed analogue. The results are obtained under the assumption that the best method of distinguishing the block cipher Magma from random permutation is key recovering by bruteforce.

2 Notation

Denote by V_n the set of n–component strings on $\mathbb{F}_2$. By $\mathbb{B}_n$ we denote the set of all byte strings of length n. By V_{n*} we denote the set of all strings with elements from $\mathbb{F}_2$ with non-zero length which is multiple of n.

By INT we denote the function $V_n \to \mathbb{N}$, such that for $X = (X_1, X_2, \ldots, X_n) \in V_n$ there is $INT(X) = 2^{n-1} \cdot X_1 + 2^{n-2} \cdot X_2 + \ldots + 2^0 \cdot X_n$.

By $\mathbb{H}$ we denote the set of $\{\mathtt{0}, \mathtt{1}, \ldots, \mathtt{15}\}$. By $\mathbb{D}$ we denote the set of $\{\mathtt{0}, \mathtt{1}, \ldots, \mathtt{9}\}$. The elements $\mathtt{10}, \mathtt{11}, \ldots, \mathtt{15}$ of $\mathbb{H}$ are called *hex-symbols*, other elements of this set, which are members of $\mathbb{D}$, are called *decimal symbols*. For the hex-elements of $\mathtt{10}, \mathtt{11}, \mathtt{12}, \mathtt{13}, \mathtt{14}, \mathtt{15}$ we'll use the symbols of $\mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{D}, \mathtt{E}, \mathtt{F}$ respectively. Strings of $\mathbb{B}_n$ are interpreted as the strings from $\mathbb{H}^{2n}$ in a natural way (corresponding hex-symbols are obtained by transformation of leading and trailing 4-bit parts of bytes from left to right). Positive integers are also interpreted as strings from $\mathbb{H}^n$ or $\mathbb{D}^n$ for some n in a natural way ($123 \to \mathtt{07B} \in \mathbb{H}^3$ and $123 \to \mathtt{0123} \in \mathbb{D}^4$).

For A by $Perm(A)$ we'll denote the set of all bijective transformations of A, and by $Func(A)$ we'll denote all mappings from A to A. A *block cipher* E is the family of $\{E_K | K \in V_k\} \subseteq Perm(V_n)$, the string of K is called the *key*.

If there is some probability distribution $\mathcal{D}$ defined over the set Ω, then a random selection of element ω from Ω in accordance with the distribution

$\mathcal{D}$ will be denoted as $\omega \xleftarrow{\mathcal{D}} \Omega$. $\mathcal{U}$ will denote a uniform distribution.

3 The procedure of computing check values

Definition 3.1. The function of *two-pass decimalization* is the mapping of $\mathsf{DEC}^2_{m,r} : \mathbb{H}^m \to \mathbb{D}^r$, $m \geqslant r$, which gives the output on an input $X \in \mathbb{H}^m$ by the following algorithm. If in X there are r decimal symbols, then the string $\mathsf{DEC}_{m,r}(X)$ is the concatenation of the first r of them (from left to right). If the total number s of decimal symbols is less than r, then the string of $\mathsf{DEC}_{m,r}(X)$ is the concatenation of these s symbols and $r - s$ residues of dividing first hex-symbols of X by 10.

For example, $\mathsf{DEC}^2_{5,3}(\mathtt{0}\|\mathtt{1}\|\mathtt{C}\|\mathtt{D}\|\mathtt{E}) = \mathtt{0}\|\mathtt{1}\|(12 \mod 10) = \mathtt{0}\|\mathtt{1}\|\mathtt{2}$.

Definition 3.2. The function of *modular decimalization* is the mapping of $\mathsf{DEC}^M_{m,r} : \mathbb{H}^m \to \mathbb{D}^r$, $m \geqslant r$, the output of which on the input of $X \in \mathbb{H}^m$ is equal to $\mathsf{DEC}^M_{m,r}(X) = INT(X) \mod 10^r$.

For example, $\mathsf{DEC}^M_{5,3}(\mathtt{0}\|\mathtt{1}\|\mathtt{C}\|\mathtt{D}\|\mathtt{E}) = \mathtt{0x1CDE} \mod 10^3 = 7390 \mod 10^3 = \mathtt{3}\|\mathtt{9}\|\mathtt{0}$.

Definition 3.3. For some block cipher $E = \{E_K | K \in V_k\} \subseteq Perm(V_n)$ by CBCMAC we denote the mapping of $\mathsf{CBCMAC} : V_k \times V_{n*} \to V_n$, such as

$$\mathsf{CBCMAC}_K(M) = \mathsf{CBCMAC}(K, M) = E_K(E_K(\ldots(E_K(M_1) \oplus M_2) \oplus \ldots) \oplus M_t),$$

where $M = M_1\| \ldots \|M_t$, $M_i \in V_n$.

From now on we will consider a block cipher E with block size $n = 64$ and key size of $k = 256$.

Definition 3.4. By F^C we'll denote the family of functions $F^C_K : V_{128} \to \mathbb{D}^3$, defined in the following way:

$$F^C_K(M) = \mathsf{DEC}_{16,3}(\mathsf{CBCMAC}_K(M)),$$

where $\mathsf{DEC}_{16,3}$ is one of the decimalization functions defined earlier. The choice of a particular function is clarified for a certain case.

Definition 3.5. By F^P denote the set of functions $F^P_K : V_{64} \to \mathbb{D}^4$, defined in the following way:

$$F^P_K(M) = \mathsf{DEC}_{16,4}(\mathsf{CBCMAC}_K(M)),$$

where $\mathsf{DEC}_{16,4}$ is one of decimalization functions defined earlier. The choice of a particular function is clarified for a certain case.

In the notations described above the CVV and PVV parameters for the «MIR» payment system can be calculated as follows.

The check value of a card CVV is calculated according to card number PAN (Personal Account Number, usually, 12–16 digits), expiration date $ExpDate$ (4 digits in the form YYMM) and service code SVC (3 digits, can take the only 6 values: 000, 999, 200, 201, 220, 221) using key value CVK (32 bytes). Before the main computation a 128-bit message M is generated:

$$M = (M_1 \| M_2) = (PAN \| \underbrace{0 \ldots 0}_{pad} \| ExpDate \| SVC \| 0000000000),$$

where pad is the the minimum number of zero hex characters to make the length of $(PAN \| 0 \ldots 0)$ equal to 16 hex characters.

The main CVV calculation uses F^C_{CVK} function:

$$CVV = F^C_{CVK}(M).$$

Similarly, the check values of a card PIN-code PVV is calculated according to card number PAN, key index $PVKI$ (number in range $0 \ldots 6$), PIN-code PIN (4 digits) using key value PVK (32 bytes). Before the main computation a 64-bit message M is generated:

$$M = (PAN|_{11} \| PVKI \| PIN),$$

where $PAN|_{11}$ — the first 11 digits of the card number PAN.

The main PVV calculation uses F^P_{PVK} function:

$$PVV = F^P_{PVK}(M).$$

4 Adversary models

Consider some adversary models, threats and attack scenarios, defined from the practice-oriented point of view.

4.1 Models of PRF, PRP-CPA and MAC-CPA

In the current section we provide the descriptions of standard adversary models that are used in mathematical cryptology for the analysis of various sets of functions (see [5]).

During the following considerations we assume that a certain computational model is chosen together with some way of encoding algorithms in this model. By *adversary* we'll assume any probabilistic algorithm in that computational model. By *computational resources* we'll assume the sum of the average (by random input and by the responses of his oracles) number of steps made during his work, and the length of the encoding of this adversary.

Let M be some adversary model and $\mathsf{Adv}^M(\mathcal{A})$ is some characteristic of possibilities of $\mathcal{A}$ when implementing the threat in model M. Let T be a set of resulting values of bounds on computational resources of an adversary and any other parameters, characterizing its work in the model of M. For example, T can additionally contain the values, bounding the total number of queries to the oracles, considered by the model M. The set $\mathcal{A}(T)$ of the adversaries, satisfying the limitations, which are defined by T is limited. By $\mathsf{Adv}^M(T)$ we'll denote the maximal value of $\mathsf{Adv}^M(\mathcal{A})$ for adversaries $\mathcal{A}$ in the set of $\mathcal{A}(T)$.

From now on by $\mathsf{Adv}^M(t, q, m)$ for a model M we'll denote a maximal value of $\mathsf{Adv}^M(\mathcal{A})$ for adversaries $\mathcal{A}$ in the set of $\mathcal{A}(t, q, m)$, where t denotes the computational resources limitations, q is the maximal available number of queries to the exploited oracle, m is the number of cipher blocks in queries (optionally).

Definition 4.1. The model of MAC-CPA for the family of $F = \{F_K : D \to R | K \in V_k\}$ is described in the following way. $\mathcal{A}$ can operate with an oracle $\mathcal{O}^{\text{MAC-CPA}}$, which chooses a key $K \xleftarrow{\mathcal{U}} V_k$ before starting his work. An adversary can make queries $M \in D$ to the oracle $\mathcal{O}^{\text{MAC-CPA}}$, for which

the oracle $\mathcal{O}^{\text{MAC-CPA}}$ responds with strings $T = F_K(M)$. As a result an adversary returns a pair of (M', T'), where $M' \in D$ is not equal to any of queries made to $\mathcal{O}^{\text{MAC-CPA}}$. An *advantage* $\mathsf{Adv}_F^{\text{MAC-CPA}}(\mathcal{A})$ *of the adversary* $\mathcal{A}$ *in the model* MAC-CPA is defined in the following way:

$$\mathsf{Adv}_F^{\text{MAC-CPA}}(\mathcal{A}) = \Pr\left[F_K(M') = T'\right].$$

The model of MAC-CPA is used to achieve security estimations for functions in some family regarding to a forgery threat.

Definition 4.2. The model of PRF for the family of $F = \{F_K : D \to R | K \in V_k\}$ is described in the following way. $\mathcal{A}$ can operate with an oracle $\mathcal{O}^{\text{PRF}}$, which chooses a bit $b \xleftarrow{\mathcal{U}} \{0,1\}$ and if $b = 1$, the oracle chooses a key $K \xleftarrow{\mathcal{U}} V_k$, assuming later $F' = F_K(\cdot)$. If $b = 0$, the oracle chooses $F' \xleftarrow{\mathcal{U}} Func(D, R)$. The adversary can make to $\mathcal{O}^{\text{PRF}}$ queries of $M \in D$, for which the oracle $\mathcal{O}^{\text{PRF}}$ responds with strings $T = F'(M)$. As a result an adversary returns a bit $a \in \{0,1\}$. An *advantage* $\mathsf{Adv}_F^{\text{PRF}}(\mathcal{A})$ *of the adversary* $\mathcal{A}$ *in the model of* PRF is defined in the following way:

$$\mathsf{Adv}_F^{\text{PRF}}(\mathcal{A}) = \Pr\left[a = 1 | b = 1\right] - \Pr\left[a = 1 | b = 0\right].$$

A reduction to the abilities of an adversary in the model of PRF allows to obtain a majority of security bounds in practice oriented adversary models. Informally, an advantage of an adversary in the model of PRF reflects abilities of an adversary in the sense of applying methods making use of special properties of transformations made by studied functions.

Definition 4.3. The model of PRP-CPA for a cipher of $E = \{E_K | K \in V_k\} \subseteq Perm(V_n)$ is described in the following way. $\mathcal{A}$ can operate with an oracle $\mathcal{O}^{\text{PRP-CPA}}$, which chooses a bit $b \xleftarrow{\mathcal{U}} \{0,1\}$ and, if $b = 1$, the oracle chooses a key $K \xleftarrow{\mathcal{U}} V_k$, assuming later $F = E_K(\cdot)$. If $b = 0$, the oracle chooses $F \xleftarrow{\mathcal{U}} Perm(V_n)$. The adversary can make to $\mathcal{O}^{\text{PRP-CPA}}$ queries of $M \in V_n$, for which the oracle $\mathcal{O}^{\text{PRP-CPA}}$ responds with strings $C = F(M)$. As a result an adversary returns a bit $a \in \{0,1\}$. An *advantage* $\mathsf{Adv}_E^{\text{PRP-CPA}}(\mathcal{A})$ *of the adversary* $\mathcal{A}$ *in the model of* PRP-CPA is defined in the following way:

$$\mathsf{Adv}_E^{\text{PRP-CPA}}(\mathcal{A}) = \Pr\left[a = 1 | b = 1\right] - \Pr\left[a = 1 | b = 0\right].$$

The model PRP-CPA is close to the model of PRF, but is usually used to evaluate security of families of bijective mappings.

4.2 The search of the CVV value for a certain attacked card

The relevant adversary model in the task of searching of the CVV value for a certain attacked card is defined as follows. The adversary knows the parameters of q cards that have been issued by the issuer using the same key CVK unknown to the adversary, i.e., q sets $(PAN, ExpDate, SVC)$ and corresponding CVV values.

In practice, the values known for the adversary are able to be obtained by cards being stolen or spying on the parameters (including the CVV). Also some cards can be issued for the adversary with the actual key value of CVK.

The threat in this model is defined in the following way: the adversary finds the correct value of CVV for a certain (attacked) card with known parameters $(PAN, ExpDate, SVC)$, for which the corresponding value of CVV remained unknown.

This would make the adversary possible to do CNP (Card-Not-Present) transactions (e.g. for e-commerce).

4.3 The search of the (PIN, PVV) values for a certain attacked card

The relevant adversary model in the task of searching of the correct (PIN, PVV) pair for a certain attacked card is defined as follows. The adversary knows the parameters of q cards that have been issued by the issuer using the same key PVK unknown to the adversary, i.e., q sets $(PAN, PVKI)$ and corresponding (PIN, PVV) values.

In practice, the values known for the adversary are able to be obtained by cards being stolen together with PIN-codes. Also some cards can be issued for the adversary with the actual key value of PVK. Also for an adversary to obtain the PVV values we assume that PVV are written on cards and the adversary can read them, and/or the PVV are stored in the issuer database and the adversary has access to it.

The threat in this model is defined in the following way: the adversary finds the correct pair of values (PIN, PVV) for a certain (attacked) card with known parameters $(PAN, PVKI)$, for which the corresponding value of correct pair remained unknown.

This would make the adversary possible to clone an existing card and make transactions with it in ATMs.

During the cryptographic analysis of the properties of the computation function of the check values CVV and PVV a standard MAC-CPA model for function families F^C and F^P is used. Thus model is a significantly stronger variant of practice-oriented models. The strengthening is in allowing the adversary to obtain the values of CVV and (PIN, PVV) for any tuples of parameters of a card, and in case of PVV also to choose PIN-values by himself PIN. Also the strengthening of the model is in replacing the selective forgery threat (where an adversary has to obtain a correct value of CVV and (PIN, PVV) for a certain parameter values), with the existential forgery threat. In the former case an adversary has to obtain a correct pair of $((PAN, ExpDate, SV), CVV)$ (in case of CVV) or $((PAN, PVKI), (PIN, PVV))$ (in case of PVV).

4.4 Search of the correct value of PIN for an attacked card with known PVV

The relevant adversary model in the task of searching of the correct PIN value for a certain attacked card in those cases where PVV value is known is defined as follows. The adversary knows the parameters of q cards that have been issued by the issuer using the same key CVK unknown to the adversary, i.e., q sets $(PAN_1, PVKI_1), \ldots, (PAN_q, PVKI_q)$ and corresponding pairs of values $(PIN_1, PVV_1), \ldots, (PIN_q, PVV_q)$. Moreover, for a certain attacked card with a known parameters $(PAN, PVKI)$ the adversary knows the only correct value PVV.

In practice, the values known for the adversary are able to be obtained by cards being stolen together with PIN-codes. Also some cards can be issued for the adversary with the actual key value of PVK. Also for an adversary to

obtain the PVV_i values we assume that PVV_i are written on cards and the adversary can read them, and/or the PVV_i are stored in the issuer database and the adversary has access to it.

The threat in this model is defined in the following way: the adversary finds the correct value of PIN for a certain (attacked) card with known parameters $(PAN, PVKI)$ and known PVV.

This would make the adversary possible to clone an existing (possibly stolen) card and make transactions with it in ATMs.

Denote the considered model by PR (**P**IN **R**ecovery). For formalization of the model consider an oracle $\mathcal{O}_{F^P}$, which implements a randomly selected function $F_K^P \xleftarrow{\mathcal{U}} F^P$. The values of $(PAN, PVKI)$ are fed to the input of the oracle $\mathcal{O}_{F^P}$ together with the value of $res \in \{\texttt{FULL}, \texttt{REDUCED}\}$. The oracle $\mathcal{O}_{F^P}$ chooses PIN randomly (according to the uniform distribution) and calculates the value of $PVV = F_K^P(PAN \| PVKI \| PIN)$. If the flag $res = \texttt{FULL}$, then the oracle outputs the pair (PIN, PVV), if $res = \texttt{REDUCED}$, then only the PVV. Assume that the adversary $\mathcal{A}^{\mathcal{O}_{F^P}}$ makes q random queries $(PAN_1, PVKI_1), \ldots, (PAN_q, PVKI_q)$ with the flag $res = \texttt{FULL}$ to this oracle and obtains $(PIN_1, PVV_1), \ldots, (PIN_q, PVV_q)$; and also makes one query $(PAN, PVKI)$ with the flag $res = \texttt{REDUCED}$ and obtains PVV as a result. Then the adversary tries to find a PIN', such that $PVV = F_K^P(PAN \| PVKI \| PIN')$.

Adversary success regarding this threat is estimated by the advantage

$$\mathsf{Adv}_{F^P}^{\mathrm{PR}}\left(\mathcal{A}^{\mathcal{O}_{F^P}}\right) = \Pr\left[\mathcal{A}^{\mathcal{O}_{F^P}} \Rightarrow PIN' : \; PVV = F_K^P(PAN \| PVKI \| PIN')\right].$$

5 Distributions of the decimalization functions

In the current section we describe the distributions of the random variables $\mathsf{DEC}_{m,r}^2(H)$ and $\mathsf{DEC}_{m,r}^M(H)$ in case when H is chosen from $\mathbb{H}^m$ according to the uniform distribution, for particular cases corresponding to the functions for calculation of CVV and PVV. The distributions of the random variables defines the resulting security bounds of the procedures that have been analyzed in the current paper.

The function $\mathsf{DEC}_{m,r}^2$ output distribution is computed in Appendix, Sec-

CVV case ($m = 16$, $r = 3$)		PVV case ($m = 16$, $r = 4$)	
# values $V \in \mathbb{D}^3$	$\Pr\left[\mathsf{DEC}^2_{16,3}(H) = V\right]$	# values $V \in \mathbb{D}^4$	$\Pr\left[\mathsf{DEC}^2_{16,4}(H) = V\right]$
240	$\approx 10^{-3} + 2.975 \cdot 10^{-8}$	864	$\approx 10^{-4} + 3.696 \cdot 10^{-8}$
144	$\approx 10^{-3} + 4.108 \cdot 10^{-8}$	2400	$\approx 10^{-4} + 2.091 \cdot 10^{-8}$
216	$\approx 10^{-3} + 4.179 \cdot 10^{-8}$	1440	$\approx 10^{-4} + 3.507 \cdot 10^{-8}$
400	$\approx 10^{-3} - 5.520 \cdot 10^{-8}$	1296	$\approx 10^{-4} + 3.707 \cdot 10^{-8}$
		4000	$\approx 10^{-4} - 4.517 \cdot 10^{-8}$

Table 1: Distributions of $\mathsf{DEC}^2_{m,r}$ for CVV and PVV cases.

CVV case ($m = 16$, $r = 3$)		PVV case ($m = 16$, $r = 4$)	
# values $V \in \mathbb{D}^3$	$\Pr\left[\mathsf{DEC}^M_{16,3}(H) = V\right]$	# values $V \in \mathbb{D}^4$	$\Pr\left[\mathsf{DEC}^M_{16,4}(H) = V\right]$
616	$\approx 10^{-3} + 2.082 \cdot 10^{-20}$	1616	$\approx 10^{-4} + 4.545 \cdot 10^{-20}$
384	$\approx 10^{-3} - 3.339 \cdot 10^{-20}$	8384	$\approx 10^{-4} - 8.760 \cdot 10^{-21}$

Table 2: Distributions of $\mathsf{DEC}^M_{m,r}$ for CVV and PVV cases.

tion 8.1, formula (7). For both cases CVV and PVV these distributions are presented in Table 1.

The function $\mathsf{DEC}^M_{m,r}$ output distribution is computed in Appendix, Section 8.2, formulas (8) and (9). For both cases CVV and PVV these distributions are presented in Table 2.

For each case in Tables 1, 2 by "# values $V \in \mathbb{D}^r$" we mean a number of $V \in \mathbb{D}^r$ such that $\Pr\left[\mathsf{DEC}_{m,r}(H) = V\right]$ equals a given probability.

6 Cryptographic analysis of the check values evaluation function

6.1 Security evaluation regarding the threat of check values forgery

In the current section the security of F^C and F^P regarding the threat of check values (generated with them) forgery is evaluated, i.e. in the model of MAC-CPA .

Using the final estimates of Section 8.4 (formulas (17)–(20)) and estimate (12), we obtain the following relations which are included in the main results of the paper:

$$
\begin{aligned}
&\mathsf{Adv}^{\text{MAC-CPA}}_{F^C}(t,q) \leqslant 10^{-3} + \frac{t+2q+2qn}{2^k} + \frac{4q^2}{2^{n-1}} + \frac{3q}{10^5} \text{ (for } \mathsf{DEC}^2_{16,3}\text{)};\\
&\mathsf{Adv}^{\text{MAC-CPA}}_{F^C}(t,q) \leqslant 10^{-3} + \frac{t+2q+2qn}{2^k} + \frac{4q^2}{2^{n-1}} + \frac{2q}{10^{17}} \text{ (for } \mathsf{DEC}^M_{16,3}\text{)};\\
&\mathsf{Adv}^{\text{MAC-CPA}}_{F^P}(t,q) \leqslant 10^{-4} + \frac{t+2q+qn}{2^k} + \frac{q^2}{2^{n-1}} + \frac{2.3q}{10^4} \text{ (for } \mathsf{DEC}^2_{16,4}\text{)};\\
&\mathsf{Adv}^{\text{MAC-CPA}}_{F^P}(t,q) \leqslant 10^{-4} + \frac{t+2q+qn}{2^k} + \frac{q^2}{2^{n-1}} + \frac{2.3q}{10^{16}} \text{ (for } \mathsf{DEC}^M_{16,4}\text{)}.
\end{aligned}
\tag{1}
$$

Example 6.1. Let us consider estimates (1) for the payment system «MIR». Since block cipher Magma is intended to be used in «MIR», we have $n = 64$, $k = 256$. Also, suppose that adversary's computational resources t are bounded by 2^{128}; the number of queries q is bounded by 10^7. The last value is an upper bound for the number of payment cards emitted by a bank during the lifetime of the key CVK and/or PVK. Under these settings we obtain the following relations for the function DEC^M:

$$
\mathsf{Adv}^{\text{MAC-CPA}}_{F^C}(t,q) \leqslant 10^{-3} + 10^{-4.36} + 10^{-9.69}; \tag{2}
$$

$$
\mathsf{Adv}^{\text{MAC-CPA}}_{F^P}(t,q) \leqslant 10^{-4} + 10^{-4.96} + 10^{-8.63}. \tag{3}
$$

Note that for the case of DEC^2 similar bounds with the same parameter values are trivial — the right side is greater than 1: $\mathsf{Adv}^{\text{MAC-CPA}}_{F^C}(t,q) \leqslant 300$ and $\mathsf{Adv}^{\text{MAC-CPA}}_{F^P}(t,q) \leqslant 230$. Moreover, the bounds for DEC^2 becomes trivial approximately as $q \geqslant 3 \cdot 10^4$.

Remark 6.1. Triviality of the estimates for DEC^2 does not necessarily mean that the functions F^C and F^P with DEC^2 are insecure. It means that even assuming the security (in the model of PRP-CPA) of the cipher E the known for the current moment methods of cryptanalysis do not allow to prove inexistence of adversaries with parameters bounded by t and q, which are able to make forgeries with probabilities significantly greater than a priori bound for guessing. Note also that there are a lot of examples that show that the bounds obtained by this technique are close to tight. By «tightness» in this case we mean the existence of a particular adversary with the limited

(accordingly) resources, who gains the obtained success probability bound, (cf. [4]). To support the statement of the fact that the usage of DEC^2 does not lead to the decrease in security we note that during the process of security reduction we use the following statement: if an adversary cannot distinguish the outputs of F^C and F^P and a random value from a corresponding set then he is now able to make an existential forgery. In case when the success probability of an adversary in the distinguishing task is greater than the a priori estimation of guessing, it seems impossible to obtain proven estimation of success probability of an adversary in the forgery.

6.2 Security evaluation regarding the threat of obtaining PIN

In the current section the security of F^P regarding the threat of obtaining PIN in the model PR is considered.

Combining the final estimates of Section 8.4 (formulas (17)–(20)) and Theorem 8.8, we obtain the following relations for the function F^P. These relations with estimates (1) are the main results of the paper:

$$\mathsf{Adv}^{\mathrm{PR}}_{F^P}(t,q) \leqslant \frac{t+q+2+qn}{2^k}+\frac{(q+2)^2}{2^{n-1}}+\frac{2.3(q+2)}{10^4}+\frac{2}{10^4}-\frac{1}{10^8} \qquad (\text{for } \mathsf{DEC}^2_{16,4}); \tag{4}$$

$$\mathsf{Adv}^{\mathrm{PR}}_{F^P}(t,q) \leqslant \frac{t+q+2+qn}{2^k}+\frac{(q+2)^2}{2^{n-1}}+\frac{2.3(q+2)}{10^{16}}+\frac{2}{10^4}-\frac{1}{10^8} \qquad (\text{for } \mathsf{DEC}^M_{16,4}). \tag{5}$$

Example 6.2. Consider the obtained estimates for the settings of Example 6.1: $n = 64$, $k = 256$, $t = 2^{128}$, $q \leq 10^7$. Under these settings we obtain the following bounds for the case of usage of DEC^M:

$$\mathsf{Adv}^{\mathrm{PR}}_{F^P}(t,q) \leqslant 2 \cdot 10^{-4} + 10^{-4.96} + 10^{-8.63}. \tag{6}$$

Here, similar to the model of MAC-CPA, bound (4) for DEC^2 under the given settings is trivial: $\mathsf{Adv}^{\mathrm{RP}}_{F^P}(t,q) \leqslant 2300$. Moreover, the bound for DEC^2 becomes trivial approximately as $q \geqslant 0.5 \cdot 10^4$.

7 Conclusion

In the current paper we obtain the security bounds for the functions used for generation of check values CVV and PVV regarding the forgery threat. We show that the usage of the function DEC^M leads to negligible difference between a priori guess probability and the probability of success of an adversary obtaining not more than 10^7 values of CVV and PVV. It is also shown that the function F^P is secure regarding to the threat of obtaining PIN value when used with DEC^M; i.e. its probability of success for an adversary with typically bounded resources is extremely close to the a priori one.

We show that for the same limitations, similar bounds for the functions, using DEC^2, degenerate. Degeneration of the bounds lead to impossibility of statements about provable security in this case.

The authors are very grateful to I. M. Goldovskii for fruitful discussions and to E. S. Smyshlyaeva for her valuable help comments and suggestions concerning the text of the paper.

References

[1] Bellare M., Kilian J., Rogaway P. The security of the cipher block chaining message authentication code. Journal of Computer and System Sciences (JCSS), vol. 61, no. 3, pp. 362–399, 2000. Earlier version in Crypto'94.

[2] Nandi M. A Simple and Unified Method of Proving Unpredictability. Cryptology eprint archive 2006/264. 2006.

[3] Application Programmer's Guide, Appendix F. Cryptographic Algorithms and Processes, PIN Formats and Algorithms, VISA PIN Algorithms// IBM Knowledge Center, 2011 (`http://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.1.0/com.ibm.zos.v2r1.csfb400/csfb4za2598.htm`)

[4] Iwata T. Comments on «On the security of XCBC, TMAC and OMAC» by Mitchell.
https://pdfs.semanticscholar.org/21b0/c40d3a08ffce60b11721b3fdd2516f37dce8.pdf — 2003.

[5] Bellare M., Rogaway P. Introduction to modern cryptography: Lecture Notes, 2001. http://www.cs.ucsd.edu/users/mihir/cse207/classnotes.html

[6] Dolmatov V. RFC 5830. GOST 28147-89: Encryption, Decryption, and Message Authentication Code (MAC) Algorithms.

8 Appendix

8.1 Calculating the distribution of $\mathsf{DEC}^2_{m,r}$ outcomes

In this section we consider the following problem: calculate the probability distribution of the random variable $\mathsf{DEC}^2_{m,r}(H)$, where H is taken from $\mathbb{H}^m$ according to a uniform distribution. To be definite we introduce the notations $H = h_1\|h_2\|\ldots\|h_m$ and $\mathsf{DEC}^2_{m,r}(H) = d_1\|\ldots\|d_r$.

Let us fix some decimal digits d_1, d_2, $\ldots$, $d_r \in \mathbb{D}$. Now we divide the event $\{H \in \mathbb{H}^m :\ \mathsf{DEC}^2_{m,r}(H) = d_1\|\ldots\|d_r\}$ into three disjoint events:

A. All decimal digits d_1, d_2, $\ldots$, d_r outcomes during the first pass of the two-pass decimalization.

B. All decimal digits d_1, d_2, $\ldots$, d_r outcomes during only the second pass of the two-pass decimalization. Note that in this case all digits d_1, d_2, $\ldots$, d_r must belong to $\{\mathtt{0},\ldots,\mathtt{5}\}$.

C. The first s decimal digits d_1, $\ldots$, d_s outcomes during the first pass of the two-pass decimalization but the other $r-s$ digits d_{s+1}, $\ldots$, d_r outcomes during the second pass. Note that in this case the first s decimal digits d_1, $\ldots$, d_s can be arbitrary but other digits d_{s+1}, $\ldots$, d_r must belong to $\{\mathtt{0},\ldots,\mathtt{5}\}$.

Further, we calculate the probabilities of these events.

A. For convenience we divide the event $\mathbf{A}$ into events $\mathbf{A}_i$, where $\mathbf{A}_i = \{H \in \mathbf{A} : d_r = h_i\}$ is an event of occurring the last output digit d_r on the i-th position of the input. Then,

$$\Pr_{\mathbf{A}}[d_1,\ldots,d_r] = \frac{1}{16^m}\cdot\sum_{i=r}^{m}\binom{i-1}{r-1}\cdot(16-10)^{i-r}\cdot 16^{m-i}.$$

In this formula each summand is the cardinality of $\mathbf{A}_i$. The first factor $\binom{i-1}{r-1}$ of the summand is the number of ways to put other digits d_1, $\ldots$, d_{r-1} on the first $i-1$ input positions; the next factor $(16-10)^{i-r}$ is the number of ways to fill input positions $1,\ldots,i$ by hexadecimal

digits not belonging to $\{d_1, \ldots, d_{r-1}\}$; finally, the factor 16^{m-i} is the number of ways to fill other input positions $i+1, \ldots, m$ by arbitrary hexadecimal digits.

B. In this event the output decimal digits $d_1, d_2, \ldots, d_r$ determine completely the first r input digits: $h_i = d_i + 10$, $i = 1, \ldots, r$. Other input hexadecimal digits $h_{r+1}, \ldots, h_m$ can be arbitrary. Thus we have

$$\Pr_{\mathbf{B}}[d_1, \ldots, d_r] = \frac{1}{16^m} \cdot (16-10)^{m-r}.$$

C. As above, for convenience we divide the event $\mathbf{C}$ into events $\mathbf{C}_i$, where $\mathbf{C}_i = \{H \in \mathbf{C} : d_s = h_i\}$ is an event of occurring the s-th output digit d_s on the i-th input position. Then,

$$\Pr_{\mathbf{C}}[d_1, \ldots, d_r; s] = \frac{1}{16^m} \cdot \sum_{i=s}^{m} \binom{i-1}{s-1} \cdot (16-10)^{m-r}.$$

In this formula each summand is the cardinality of $\mathbf{C}_i$. The first factor $\binom{i-1}{s-1}$ of the summand is the number of ways to put the first $s-1$ output decimal digits $d_1, \ldots, d_{s-1}$ on the first $i-1$ input positions. The first $r-s$ positions of other $m-s$ input positions are completely determined by the digits $d_{s+1}, \ldots, d_r$, and the last $m-r$ positions can be elements of $\{\mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{D}, \mathtt{E}, \mathtt{F}\}$. This reasons the presence of the second factor $(16-10)^{m-r} = (16-10)^{(m-s)-(r-s)}$ in the summand.

By definition, put $t = \max(\{0\} \cup \{i = 1, \ldots, r |\ d_i \in \{\mathtt{6}, \mathtt{7}, \mathtt{8}, \mathtt{9}\}\})$. It is readily seen that all digits $d_1, \ldots, d_t$ outcomes during the first pass of the two-pass decimalization. Now, using the introduced notations, we obtain the final formula for calculating the probability of DEC^2 output:

$$\Pr[d_1, \ldots, d_r] = \\ = \begin{cases} \Pr_{\mathbf{A}}[d_1, \ldots, d_r] + \sum\limits_{s=t}^{r-1} \Pr_{\mathbf{C}}[d_1, \ldots, d_r; s], & t > 0; \\ \Pr_{\mathbf{A}}[d_1, \ldots, d_r] + \Pr_{\mathbf{B}}[d_1, \ldots, d_r] + \sum\limits_{s=1}^{r-1} \Pr_{\mathbf{C}}[d_1, \ldots, d_r; s], & t = 0. \end{cases} \tag{7}$$

8.2 Calculating the distribution of $\mathsf{DEC}^M_{m,r}$ outcomes

In this section we consider the following problem: calculate the probability distribution of the random variable $\mathsf{DEC}^M_{m,r}(H)$, where H is taken from $\mathbb{H}^m$ according to a uniform distribution.

Let d and w be integers numbers such that $16^m = d \cdot 10^r + w$, where $0 \leqslant w < 10^r$. Then for all w values $V \in \{0, \ldots, w-1\}$ the cardinality of a preimage of V under $\mathsf{DEC}^M_{m,r}$ is equal to $d+1 = (16^m + 10^r - w)/10^r$. For all other $10^r - w$ values $V \in \{w, \ldots, 10^r - 1\}$ the cardinality of a preimage of V under $\mathsf{DEC}^M_{m,r}$ is equal to $d = (16^m - w)/10^r$. If H is taken of $\mathbb{H}^m$ according to a uniform distribution, then for $V \in \{0, \ldots, w-1\}$ the following equality holds

$$\Pr[V] = \frac{16^m + 10^r - w}{10^r \cdot 16^m} = \frac{1}{10^r} + \frac{1 - w/10^r}{16^m}, \tag{8}$$

and for $V \in \{w, \ldots, 10^r - 1\}$ the following equality holds

$$\Pr[V] = \frac{16^m - w}{10^r \cdot 16^m} = \frac{1}{10^r} - \frac{w/10^r}{16^m}. \tag{9}$$

8.3 Classical models and known results about their relationship

In the current section we provide known results about security bounds in the models defined in Section 4.1.

Let $\mathcal{A}'$ be an adversary working in conditions defined by some model M' and using some adversary $\mathcal{A}$, working in conditions defined by some model M, as a subroutine («a black box»). Let

$$\mathsf{Adv}^{M'}(\mathcal{A}') \geqslant \mathsf{Adv}^{M}(\mathcal{A}) - \theta, \tag{10}$$

where θ is some value that depends on the limitations of $\mathcal{A}$.

If $\mathcal{A}$ is a member of $\mathcal{A}(T)$ for some set of limitations T, then the inequality $\mathsf{Adv}^{M'}(\mathcal{A}') \geqslant \mathsf{Adv}^{M}(T) - \theta(T)$ holds, where θ is some value depending on T. Let the parameters of $\mathcal{A}'$ such as computational resources and the total number of oracle queries, assuming limitations of $\mathcal{A}$ by values in T, are limited by some values, that we denote as T'. Then the following inequality

stays: $\mathsf{Adv}^{M'}(T') \geqslant \mathsf{Adv}^{M'}(\mathcal{A}')$, hence we obtain that

$$\mathsf{Adv}^{M'}(T') \geqslant \mathsf{Adv}^{M}(T) - \theta(T),$$

which is equivalent to

$$\mathsf{Adv}^{M}(T) \leqslant \mathsf{Adv}^{M'}(T') + \theta(T). \tag{11}$$

To gain absolute formal correctness in research anywhere where it is possible without any harm to practical applications, usually statements connected with estimations like (10), correct in any computational models are formulated in theorems. The estimations like (11) remain correct for any fixed computational model. In that case a full description of adversary algorithm $\mathcal{A}'$ is needed, together with estimation of his computational resources.

The value of absolute accuracy in such estimations is questionable. This is because the results of this kind are closer to practical than to theoretical area — the obtained estimations do not prove that some algorithms are polynomial or that they belong to certain classes. Their main purpose is to let the parameters of concrete applications and protocols using some constructions to be estimated. In the vast majority of cases that are interesting from the practical applications point of view, the computational resources that are available for an adversary are much larger the complexity of the reduction, that is the part of the resources of $\mathcal{A}'$, which is necessary to implement the threat in his model excluding the resources of «a black box» $\mathcal{A}$. Exactly this part leads to the difference between the computational resources in T' and T. Also it must be noticed that other parameters in T' can often be evaluated accurately.

Assumption 8.1. Based on these considerations in the current work we consider the RAM (Random Access Memory) computational model, which is pretty close to real computers. Also we will assume that the operations that require negligible memory and computational costs are presented as a single computational resource.

8.3.1 The «PRF-as-a-MAC» principle

The following statement demonstrates the known principle «PRF-as-a-MAC paradigm» (see [5]).

Theorem 8.1 ([5]). *Let* $F = \{F_K : D \to R | K \in V_k\}$. *If* $\mathcal{A}$ *is an adversary in a model of* MAC-CPA, *making not more than* q *queries to an oracle of* $\mathcal{O}_F^{\text{MAC-CPA}}$, *then an adversary* $\mathcal{B}$ *exists, who makes to the oracle of* $\mathcal{O}_F^{\text{PRF}}$ *not more than* $q+1$ *queries, and he is such that*

$$\mathsf{Adv}_F^{\text{PRF}}(\mathcal{B}) \geqslant \mathsf{Adv}_F^{\text{MAC-CPA}}(\mathcal{A}) - \frac{1}{|R|}.$$

The adversary $\mathcal{B}$, which is constructed in the proof of Theorem 8.1, makes some additional actions for intercepting queries of the adversary $\mathcal{A}$ to its oracle $\mathcal{O}_F^{\text{MAC-CPA}}$ and for retranslating responses of the $\mathcal{B}$'s oracle $\mathcal{O}_F^{\text{PRF}}$ to $\mathcal{A}$. We assume that the complexity of these operations is proportional to the number of queries with the proportionality constant to be equal to 1. Taking into account Assumption 8.1, we obtain the approximate inequality:

$$\mathsf{Adv}_F^{\text{MAC-CPA}}(t, q) \leqslant \mathsf{Adv}_F^{\text{PRF}}(t+q, q+1) + \frac{1}{|R|}. \tag{12}$$

8.3.2 Security estimation of CBCMAC

It is known that CBCMAC is not secure in the model of MAC-CPA in the case of variable-length queries. In [1] the following theorem was obtained:

Theorem 8.2 ([1]). *If* $\mathcal{A}$ *makes not more than* q *queries of a fixed length* m *to* $\mathcal{O}_{\mathsf{CBCMAC}}^{\text{PRF}}$, *then* $\mathcal{B}$, *making not more than* qm *queries to* $\mathcal{O}^{\text{PRP-CPA}}$, *exists, and he is such that*

$$\mathsf{Adv}_E^{\text{PRP-CPA}}(\mathcal{B}) \geqslant \mathsf{Adv}_{\mathsf{CBCMAC}}^{\text{PRF}}(\mathcal{A}) - \frac{q^2 m^2}{2^{n-1}}.$$

The computational resources that are additionally needed for $\mathcal{B}$ are approximately equal to mqn. Taking into account Assumption 8.1, we obtain the approximate inequality:

$$\mathsf{Adv}_{\mathsf{CBCMAC}}^{\text{PRF}}(t, q, m) \leqslant \mathsf{Adv}_E^{\text{PRP-CPA}}(t + mqn, qm) + \frac{q^2 m^2}{2^{n-1}}. \tag{13}$$

8.3.3 Security estimation of a block cipher in the model of PRP-CPA

Assumption 8.2. For a block cipher, for which no special methods of security decrease are known, the value of $\mathsf{Adv}_E^{\text{PRP-CPA}}(t,q)$ is estimated by the general method that can solve the corresponding task — the exhaustive search. If the volume of queries doesn't exceed the unicity distance of the block cipher E, i.e. as $qn \geq k$, then we can assume that for such a cipher the following approximate equality holds:

$$\mathsf{Adv}_E^{\text{PRP-CPA}}(t,q) \approx \frac{t}{2^k}. \tag{14}$$

Thus for the considered cipher Magma with parameters $n = 64$ and $k = 256$ we can assume:

$$\mathsf{Adv}_E^{\text{PRP-CPA}}(t,q) \approx \frac{t}{2^{256}}. \tag{15}$$

8.3.4 Two random variables distinguishing task

For a random variable X let $X[q]$ be a random vector of length q, components of which are pairwise independent random variables distributed as X.

Let $d_{stat}(X,Y)$ be a *statistical distance between random variables* X *and* Y, which takes values in a finite set S. By definition, put

$$d_{stat}(X,Y) = \frac{1}{2} \cdot \sum_{s \in S} |\Pr[X \to s] - \Pr[Y \to s]|,$$

where $\Pr[X \to s]$ and $\Pr[Y \to s]$ are the probabilities of outcome $s \in S$ for the random variables X and Y respectively.

Now suppose F and G are random variables, the adversary $\mathcal{A}$ is trying to distinguish F and G. Suppose $\mathcal{A}$ has an access to the oracle $\mathcal{O}$ that implements during one session either the random variable F or the random variable G. We will denote by $\mathcal{O}_F$ ($\mathcal{O}_G$) the case when the oracle $\mathcal{O}$ implements the random variable F (G) during the session. Let $\mathcal{A}$ during one session gets q independent outcomes $a_1, \ldots, a_q$ for the random variable F or the random variable G from the oracle $\mathcal{O}$.

As a result of the work the adversary outputs 0 (an assumption that the oracle implements the random variable F) or 1 (an assumption that the oracle implements the random variable G). We will denote by $\mathcal{A} \Rightarrow 0$ or $\mathcal{A} \Rightarrow 1$ the case when the adversary outputs 0 or 1 respectively. *The advantage of the adversary* in this task is the following value:

$$\mathsf{Adv}_{F,G}(\mathcal{A}) = \left|\Pr\left[\mathcal{A}^{\mathcal{O}_F} \Rightarrow 1\right] - \Pr\left[\mathcal{A}^{\mathcal{O}_G} \Rightarrow 1\right]\right|.$$

The following inequality was proved in [2]:

$$\mathsf{Adv}_{F,G}(\mathcal{A}) \leqslant d_{stat}(F[q], G[q]). \tag{16}$$

8.4 Security evaluation regarding distinguishing against a random function

Let H be a random variable uniformly distributed over $\mathbb{H}^m$, D be a random variable uniformly distributed over $\mathbb{D}^r$. Consider a problem of distinguishing between $\mathsf{DEC}_{m,r}(H)$ and D, where $\mathsf{DEC}_{m,r}(H)$ is one of the two decimalization functions defined in Section 3. From the inequality (16) we get immediately the following result.

Lemma 8.3. *Let $\mathcal{A}$ be an adversary which solves the distinguishing task between $\mathsf{DEC}_{m,r}(H)$ and D. Suppose $\mathcal{A}$ gets q independent outcomes of the random variable $\mathsf{DEC}_{m,r}(H)$ or the random variable D. Then the following inequality holds:*

$$\mathsf{Adv}_{\mathsf{DEC}_{m,r}(H),D}(\mathcal{A}) \leqslant d_{stat}(\mathsf{DEC}_{m,r}(H)[q], D[q]).$$

Lemma 8.4. *Let $\mathcal{A}$ be an adversary which solves the distinguishing task between $\mathsf{DEC}_{16,3}(H)$ and D. Suppose $\mathcal{A}$ gets q independent outcomes of the random variable $\mathsf{DEC}_{16,3}(H)$ or the random variable D. Then the following inequalities holds:*

$$\mathsf{Adv}_{\mathsf{DEC}^2_{16,3}(H),D}(\mathcal{A}) \leqslant \frac{3q}{10^5},$$

$$\mathsf{Adv}_{\mathsf{DEC}^M_{16,3}(H),D}(\mathcal{A}) \leqslant \frac{2q}{10^{17}}.$$

Proof. By Lemma 8.3, it is sufficient to estimate the values $d_{stat}(\mathsf{DEC}^2_{16,3}(H)[q], D[q])$ and $d_{stat}(\mathsf{DEC}^M_{16,3}(H)[q], D[q])$.

Let us estimate the value $1/2 \sum_{d\in\mathbb{D}^{3q}} |\Pr\left[\mathsf{DEC}^2_{16,3}(H)[q] = d\right] - \Pr\left[D[q] = d\right]|$. From Table 1, we get

$$\frac{1}{2}\sum_{d\in\mathbb{D}^{3q}} |\Pr\left[\mathsf{DEC}^2_{16,3}(H)[q] = d\right] - \Pr\left[D[q] = d\right]| \leqslant$$

$$\leqslant \frac{1}{2}\cdot 10^{3q}\cdot\left(10^{-3q} - \left(10^{-3} - 5.521\cdot 10^{-8}\right)^q\right) =$$

$$= \frac{1}{2}\cdot\left(1 - \left(1 - 5.521\cdot 10^{-5}\right)^q\right) \leqslant \frac{1}{2}\cdot\left(\frac{5.521\cdot q}{10^5}\right) \leqslant \frac{3q}{10^5}.$$

Similarly, from Table 2, we get the estimation in the case of $\mathsf{DEC}^M_{16,3}$:

$$\frac{1}{2}\sum_{d\in\mathbb{D}^{3q}} |\Pr\left[\mathsf{DEC}^M_{16,3}(H)[q] = d\right] - \Pr\left[D[q] = d\right]| \leqslant \frac{2q}{10^{17}}.$$

□

Likewise, next Lemma 8.5 is proved in the case of $\mathsf{DEC}_{16,4}$.

Lemma 8.5. *Let $\mathcal{A}$ be an adversary which solves the distinguishing task between $\mathsf{DEC}_{16,4}(H)$ and D. Suppose $\mathcal{A}$ gets q independent outcomes of the random variable $\mathsf{DEC}_{16,4}(H)$ or the random variable D. Then the following inequalities holds:*

$$\mathsf{Adv}_{\mathsf{DEC}^2_{16,4}(H),D}(\mathcal{A}) \leqslant \frac{2.3q}{10^4},$$

$$\mathsf{Adv}_{\mathsf{DEC}^M_{16,4}(H),D}(\mathcal{A}) \leqslant \frac{2.3q}{10^{16}}.$$

Theorem 8.6. *If $\mathcal{A}$ is an adversary, making not more than q queries to the oracle $\mathcal{O}^{\mathrm{PRF}}_{F^C}$, then there exists another adversary $\mathcal{B}$, making not more than q queries to the oracle $\mathcal{O}^{\mathrm{PRF}}_{\mathsf{CBCMAC}}$, such that:*

$$\mathsf{Adv}^{\mathrm{PRF}}_{\mathsf{CBCMAC}}(\mathcal{B}) \geqslant \mathsf{Adv}^{\mathrm{PRF}}_{F^C}(\mathcal{A}) - \frac{3q}{10^5}, \quad \textit{for } \mathsf{DEC}^2_{16,3};$$

$$\mathsf{Adv}^{\mathrm{PRF}}_{\mathsf{CBCMAC}}(\mathcal{B}) \geqslant \mathsf{Adv}^{\mathrm{PRF}}_{F^C}(\mathcal{A}) - \frac{2q}{10^{17}}, \quad \textit{for } \mathsf{DEC}^M_{16,3}.$$

The length of each query equals two blocks.

Proof. Let us construct the adversary $\mathcal{B}$. The adversary $\mathcal{B}$ uses the adversary $\mathcal{A}$ as a black box (or a subprogram) and intercepts all queries of $\mathcal{A}$ to its oracle. Let M be a query of $\mathcal{A}$ to its oracle. Then the adversary $\mathcal{B}$ treats this query by the following way: makes a query M to its own oracle $\mathcal{O}^{\mathrm{PRF}}_{\mathsf{CBCMAC}}$; gets the oracle response T, and returns the value $\mathsf{DEC}_{16,3}(T)$ to $\mathcal{A}$. Obtaining $\mathsf{DEC}_{16,3}(T)$ from $\mathcal{B}$, the adversary $\mathcal{A}$ does its own work and returns a bit. So now the adversary $\mathcal{B}$ outputs this bit.

Denote by b the bit which sets the behavior of the oracle $\mathcal{O}^{\mathrm{PRF}}_{\mathsf{CBCMAC}}$, by b' the bit which sets behavior of the oracle $\mathcal{O}^{\mathrm{PRF}}_{F^C}$. Note that $\mathcal{B}$ simulates the oracle $\mathcal{O}^{\mathrm{PRF}}_{F^C}$ for $\mathcal{A}$.

Let us estimate advantage of $\mathcal{B}$.

$$\begin{aligned}
\mathsf{Adv}^{\mathrm{PRF}}_{\mathsf{CBCMAC}}(\mathcal{B}) &= \Pr[\mathcal{B} \Rightarrow 1 | b = 1] - \Pr[\mathcal{B} \Rightarrow 1 | b = 0] = \\
&= \Pr[\mathcal{A} \Rightarrow 1 | b' = 1] - \Pr[\mathcal{A} \Rightarrow 1 | b = 0] = \\
&= (\Pr[\mathcal{A} \Rightarrow 1 | b' = 1] - \Pr[\mathcal{A} \Rightarrow 1 | b' = 0]) - \\
&- (\Pr[\mathcal{A} \Rightarrow 1 | b = 0] - \Pr[\mathcal{A} \Rightarrow 1 | b' = 0]) = \\
&= \mathsf{Adv}^{\mathrm{PRF}}_{F^C}(\mathcal{A}) - (\Pr[\mathcal{A} \Rightarrow 1 | b = 0] - \Pr[\mathcal{A} \Rightarrow 1 | b' = 0]).
\end{aligned}$$

Let us consider the probability $\Pr[\mathcal{A} \Rightarrow 1 | b = 0] - \Pr[\mathcal{A} \Rightarrow 1 | b' = 0]$. Note that if $b = 0$, then $\mathcal{O}^{\mathrm{PRF}}_{\mathsf{CBCMAC}}$ behaves as a random variable D defined at the beginning of this section. If $b' = 0$, then $\mathcal{O}^{\mathrm{PRF}}_{F^C}$ behaves as a random variable $\mathsf{DEC}_{m,r}(H)$.

Therefore the following equality holds

$$|\Pr[\mathcal{A} \Rightarrow 1 | b = 0] - \Pr[\mathcal{A} \Rightarrow 1 | b' = 0]| = \mathsf{Adv}_{\mathsf{DEC}_{16,3}(H),D}(\mathcal{A}).$$

Finally, combining this equality and Lemma 8.4, we get the desired result. □

In the same way, an analogous result for the function F^P can be proved.

Theorem 8.7. *If $\mathcal{A}$ is an adversary, making not more than q queries to the oracle $\mathcal{O}^{\mathrm{PRF}}_{F^P}$, then there exists another adversary $\mathcal{B}$, making not more than q queries to the oracle $\mathcal{O}^{\mathrm{PRF}}_{\mathsf{CBCMAC}}$, such that:*

$$\mathsf{Adv}^{\mathrm{PRF}}_{\mathsf{CBCMAC}}(\mathcal{B}) \geqslant \mathsf{Adv}^{\mathrm{PRF}}_{F^P}(\mathcal{A}) - \frac{2.3q}{10^4}, \quad \textit{for } \mathsf{DEC}^2_{16,4};$$

$$\mathsf{Adv}^{\mathrm{PRF}}_{\mathsf{CBCMAC}}(\mathcal{B}) \geqslant \mathsf{Adv}^{\mathrm{PRF}}_{F^P}(\mathcal{A}) - \frac{2.3q}{10^{16}}, \quad \textit{for } \mathsf{DEC}^{M}_{16,4}.$$

The length of each query equals two blocks.

Note that additional computational resources when using the $\mathcal{A}$ by $\mathcal{B}$ are needed to independent calculation of the functions of $\mathsf{DEC}_{m,r}$. Since the complexity of calculation of these functions is negligible, the volume of these additional resources can be estimated by q. Using bound (13) for $\mathsf{Adv}^{\mathrm{PRF}}_{\mathsf{CBCMAC}}(t+q, q, 2)$ and bound (14) for security of an exploited cipher in the model of PRP-CPA, taking into account the considerations of Section 8.3, we obtain the following inequalities for the function of F^C:

$$\mathsf{Adv}^{\mathrm{PRF}}_{F^C}(t, q) \leqslant \frac{t + q + 2qn}{2^k} + \frac{4q^2}{2^{n-1}} + \frac{3q}{10^5} \text{ (for } \mathsf{DEC}^{2}_{16,3}\text{)} \tag{17}$$

$$\mathsf{Adv}^{\mathrm{PRF}}_{F^C}(t, q) \leqslant \frac{t + q + 2qn}{2^k} + \frac{4q^2}{2^{n-1}} + \frac{2q}{10^{17}} \text{ (for } \mathsf{DEC}^{M}_{16,3}\text{)}. \tag{18}$$

For F^P we have the following inequalities:

$$\mathsf{Adv}^{\mathrm{PRF}}_{F^P}(t, q) \leqslant \frac{t + q + qn}{2^k} + \frac{q^2}{2^{n-1}} + \frac{2.3q}{10^4} \text{ (for } \mathsf{DEC}^{2}_{16,4}\text{)}; \tag{19}$$

$$\mathsf{Adv}^{\mathrm{PRF}}_{F^P}(t, q) \leqslant \frac{t + q + qn}{2^k} + \frac{q^2}{2^{n-1}} + \frac{2.3q}{10^{16}} \text{ (for } \mathsf{DEC}^{M}_{16,4}\text{)}. \tag{20}$$

8.5 Reduction of the PIN recovery problem to the PRF distinguishing problem

Theorem 8.8. *If $\mathcal{A}$ is an adversary, making not more than q queries to the oracle $\mathcal{O}^{\mathrm{PR}}_{F^P}$, then there exists another adversary $\mathcal{B}$, making not more than $q+2$ queries to the oracle $\mathcal{O}^{\mathrm{PRF}}_{F^P}$, such that:*

$$\mathsf{Adv}^{\mathrm{PRF}}_{F^P}(\mathcal{B}) \geqslant \mathsf{Adv}^{\mathrm{PR}}_{F^P}(\mathcal{A}) - \frac{2}{10^4} + \frac{1}{10^8}.$$

Proof. Let us construct the adversary $\mathcal{B}$. The adversary $\mathcal{B}$ uses the adversary $\mathcal{A}$ as a black box (or a subprogram) and intercepts all queries of $\mathcal{A}$ to its oracle. Suppose $(PAN, PVKI)$ is a query of $\mathcal{A}$ to its oracle such that flag

of this query equals `FULL`. Then the adversary $\mathcal{B}$ treats this query by the following way: choses $PIN \xleftarrow{\mathcal{U}} \mathbb{D}^4$, makes a query $M = (PAN, PVKI, PIN)$ to its own oracle $\mathcal{O}_{F^P}^{\mathrm{PRF}}$, gets the oracle response PVV, and returns the pair (PIN, PVV) to $\mathcal{A}$. Now suppose $(PAN', PVKI')$ is a unique special query of $\mathcal{A}$ to its oracle such that flag of this query equals `REDUCED`. Then the adversary $\mathcal{B}$ treats this query by the following way: choses $PIN' \xleftarrow{\mathcal{U}} \mathbb{D}^4$, makes a query $M = (PAN', PVKI', PIN')$ to its own oracle $\mathcal{O}_{F^P}^{\mathrm{PRF}}$, gets the oracle response PVV', memorizes it, and returns only PVV' to $\mathcal{A}$. Obtaining PVV' from $\mathcal{B}$, the adversary $\mathcal{A}$ does its own work and returns a value PIN''. So now the adversary $\mathcal{B}$ makes a query $M' = (PAN', PVKI', PIN'')$ to its oracle $\mathcal{O}_{F^P}^{\mathrm{PRF}}$, gets the response PVV'', and outputs bit 1 if $PVV' = PVV''$, 0 otherwise.

Denote by b the bit which sets the behavior of the oracle $\mathcal{O}_{F^P}^{\mathrm{PRF}}$. Let us estimate advantage of $\mathcal{B}$.

$$\begin{aligned}\mathsf{Adv}_{F^P}^{\mathrm{PRF}}(\mathcal{B}) &= \Pr\left[\mathcal{B} \Rightarrow 1 | b = 1\right] - \Pr\left[\mathcal{B} \Rightarrow 1 | b = 0\right] = \\ &= \Pr\left[\mathcal{A} \Rightarrow PIN'',\ PVV' = PVV'' | b = 1\right] - \\ &- \Pr\left[\mathcal{A} \Rightarrow PIN'',\ PVV' = PVV'' | b = 0\right] = \\ &= \mathsf{Adv}_{F^P}^{\mathrm{PR}}(\mathcal{A}) - \Pr\left[\mathcal{A} \Rightarrow PIN'',\ PVV' = PVV'' | b = 0\right]. \quad (21)\end{aligned}$$

Let us consider the probability $\Pr\left[\mathcal{A} \Rightarrow PIN'',\ PVV'' = PVV' | b = 0\right]$. Note that if $b = 0$ then $\mathcal{O}_{F^P}^{\mathrm{PRF}}$ behaves as a random function $F \xleftarrow{\mathcal{U}} Func(V_{64}, \mathbb{D}^4)$. Obviously, for any fixed pair $(PAN', PVKI')$ we have a random function

$$f \in Func(\mathbb{D}^4, \mathbb{D}^4) : f = F(PAN', PVKI', \cdot).$$

Hence for any special query $(PAN', PVKI')$ the following relations hold:

$$\begin{aligned}
&\Pr\left[\mathcal{A} \Rightarrow PIN'',\ PVV'' = PVV' | b = 0\right] = \\
&\qquad = \Pr\left[\mathcal{A} \Rightarrow PIN'', f(PIN') = f(PIN'')\right] = \\
&= \sum_{PIN' \in \mathbb{D}^4} \Pr\left[\mathcal{A} \Rightarrow PIN'', f(PIN') = f(PIN'') | PIN'\right] \cdot \Pr\left[PIN'\right] = \\
&\quad = \frac{1}{10^4} \cdot \sum_{PIN' \in \mathbb{D}^4} \Pr\left[\mathcal{A} \Rightarrow PIN'', f(PIN') = f(PIN'') | PIN'\right] = \\
&\qquad\qquad = \frac{1}{10^4} \cdot \sum_{PIN' \in \mathbb{D}^4} \Pr\left[\mathcal{A} \Rightarrow PIN'', f(PIN'') = PVV'\right].
\end{aligned}$$

Note that

$$\begin{aligned}
&\Pr\left[\mathcal{A} \Rightarrow PIN'', f(PIN'') = PVV'\right] = \\
&= \underbrace{\Pr\left[\mathcal{A} \Rightarrow PIN'', f(PIN'') = PVV' | PIN' = PIN''\right]}_{\leqslant 1} \cdot \Pr\left[PIN' = PIN''\right] + \\
&+ \underbrace{\Pr\left[\mathcal{A} \Rightarrow PIN'', f(PIN'') = PVV' | PIN' \neq PIN''\right]}_{= \frac{1}{10^4}} \cdot \Pr\left[PIN' \neq PIN''\right] \leqslant \\
&\qquad\qquad \leqslant \frac{1}{10^4} + \frac{1}{10^4} \cdot \left(1 - \frac{1}{10^4}\right) = \frac{2}{10^4} - \frac{1}{10^8}.
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\Pr\left[\mathcal{A} \Rightarrow PIN'',\ PVV'' = PVV' | b = 0\right] = \\
&\qquad = \frac{1}{10^4} \cdot \sum_{PIN' \in \mathbb{D}^4} \Pr\left[\mathcal{A} \Rightarrow PIN'', f(PIN'') = PVV'\right] \leqslant \\
&\qquad\qquad \leqslant \frac{1}{10^4} \cdot 10^4 \cdot \left(\frac{2}{10^4} - \frac{1}{10^8}\right) \leqslant \frac{2}{10^4} - \frac{1}{10^8}. \quad (22)
\end{aligned}$$

Combining (21), (22), we get the desired estimation:

$$\mathsf{Adv}^{\mathrm{PRF}}_{F^P}(\mathcal{B}) \geqslant \mathsf{Adv}^{\mathrm{PR}}_{F^P}(\mathcal{A}) - \frac{2}{10^4} + \frac{1}{10^8}.$$

Note that additional computational resources when using $\mathcal{A}$ by $\mathcal{B}$ are needed to form and pass his queries. The volume of these additional resources can be estimated as $q + 2$. □

Parallel and double block cipher mode of operation (PD–mode) for authenticated encryption

Vladislav Nozdrunov

Abstract

An authenticated encryption (AE) scheme is a secret key cryptographic primitive wich combines encryption and authentication. In this paper, we propose synthesis and analysis for new block cipher mode of operation that is called PD (parallel and double). PD–mode in the same time ensure confidentiality and integrity of information. It is fully parallelizable, inverse free and online. We also propose it's provable characteristics according to the modern way in analysis of cryptographic primitives.

Keywords: block cipher mode, authenticated encryption, provable security.

1 Introduction

Despite the fact that Authenticated Encryption is being developed for nearly 20 years the Russian Federation has not got yet National standards in these sphere. To begin the discussion of this problem we propose a new block cipher mode of operation that is called PD–mode (parallel and double). PD–mode may be considered as development of well known block cipher mode of operation – GCM [1]. During the construction of PD–mode our task was to create paralleziable, inverse free, online and secure block cipher mode without the drawbacks inherent to GCM [2], [3], [4], [7]. To solve this problem we replace the polynomial function GHASH by multilinear function. To support full parallelization for multilinear function we add second counter. But due to two counters and in order to provide necessary security level it becomes necessary to change encryption. That's why encryption in PD–mode differs from counter mode (CTR) of Russian Federation national standard GOST R 34.13–2015.

2 Preliminaries

n	block length (in bits) for a block cipher;
k	length (in bits) of the block cipher key;
V_s	the set of all binary strings of length s, where s is a non-negative integer;
V^*	the set of all binary vector-strings of finite length (hereinafter referred to as strings), including an empty string;
N	$n-1$–bit string called nonce;
A_i	i–th n–bit associated data block (the last block may be a partial block);
P_i	i–th n–bit plaintext block (the last block may be a partial block);
E_K	encryption function of the block cipher keyed by key $K \in V_k$;
$\oplus$	bitwise addition modulo 2 of two binary strings of the same length;
$\otimes$	multiplication in $GF(2^n)$;
$\boxplus_s$	the addition operation in $\mathbb{Z}_{2^s}$;
$MSB_s(X)$	mapping associating the string $z_{m-1}\| \ldots \|z_1\|z_0$, $m \geq s$, with the string $z_{m-1}\| \ldots \|z_{m-s+1}\|z_{m-s}$, $z_i \in V_1$, $i = 0, 1, \ldots, m-1$;
$LSB_s(X)$	mapping associating the string $z_{m-1}\| \ldots \|z_1\|z_0$, $m \geq s$, with the string $z_{s-1}\| \ldots \|z_1\|z_0$, $z_i \in V_1$, $i = 0, 1, \ldots, m-1$;
$Vec_s : \mathbb{Z}_{2^s} \to V_s$	the bijective mapping which for an integer from $\mathbb{Z}_{2^s}$ puts into correspondence its binary representation, i.e. for any $z \in \mathbb{Z}_{2^s}$ represented as $z = z_0 + 2 \cdot z_1 + \ldots + 2^{s-1} \cdot z_{s-1}$, where $z_i \in \{0, 1\}$, $i = 0, 1, \ldots, s-1$, the equality $Vec_s(z) = z_{s-1}\| \ldots \|z_1\|z_0$ holds;
$Int_s : V_s \to \mathbb{Z}_{2^s}$	the mapping inverse to the mapping Vec_s, i.e. $Int_s = Vec_s^{-1}$;
a^s	binary strings of length s: $a^s = (a, a, \ldots, a)$, $a \in V_1$;

$l : \bigcup_{s=0}^{64} V_s \to V_{64}$ The function that returns a 64-bit string containing the nonnegative integer describing the number of bits in its argument.;

3 Description of AE

PD mode has four inputs:

1. nonce $N \in V_{n-1}$,
2. associated data $A \in V^*$,
3. plaintext $P \in V^*$,
4. cipher key $K \in V_k$.

and two outputs:

1. ciphertext $C \in V^*$;
2. an authentication tag $T \in V_n$.

Associated data A consists of n bit strings, where the bit length of the last string is $t \in \mathbb{N}$:

$$A = A_1 \| \ldots \| A_h^*,\ A_j \in V_n, A_h^* \in V_t, h \in \mathbb{N} \cup \{0\},$$

plaintext P consists of n bit strings, where the bit length of the last string is u:

$$P = P_1 \| \ldots \| P_q^*,\ P_i \in V_n, P_q^* \in V_u, q \in \mathbb{N} \cup \{0\}$$

$1 \le j \le q-1$, $1 \le i \le h-1$, and $1 \le u, t \le n$, $h + q > 0$.

PD–mode utilizes two sequences that we call counters. First one Y_i, $i = 1, 2, \ldots$. Y_i, $i = 1, 2, \ldots$ is initialized by value $Y_1 = E_K(0^1 \| N)$.

Encryption is defined by the following equations:

$$\begin{cases} Y_i = incr_r(Y_{i-1}), & 2 \le i \le q, \\ C_i = P_i \oplus E_K(Y_i), & 1 \le i \le q-1, \\ C_q^* = P_q^* \oplus MSB_u(E_K(Y_q)), & \end{cases}$$

where $incr_r$ is the function defined by equation:

$$incr_r(L\|R) = L\|Vec_{n/2}(Int_{n/2}(R) \boxplus_{2^{n/2}} 1),\ L, R \in V_{n/2}.$$

The last blocks $A_h^* \in V_t$ and $C_q^* \in V_u$ are padded in the following manner:

$$\begin{cases} A_h = A_h^*\|0^{n-t}, \\ C_q = C_q^*\|0^{n-u}. \end{cases}$$

Authentication tag T is generated as follows:

$$T = E_K\left(\sum_{i=1}^{h} H_i \otimes A_i \oplus \sum_{j=1}^{q} H_{h+j} \otimes C_j \oplus H_{h+q+1} \otimes (l(A)\|l(C))\right),$$

$H_i = E_K(Z_i)$ and the second counter Z_i, $i = 1, 2, \ldots$ is defined by equations:

$$\begin{cases} Z_1 = E_K(1^1\|N), \\ Z_i = incr_l(Z_{i-1}),\ 2 \le i \le h+q+1. \end{cases}$$

The function $incr_l$ is defined by equation:

$$incr_l(L\|R) = Vec_{n/2}(Int_{n/2}(L) \boxplus_{2^{n/2}} 1)\|R,\ L, R \in V_{n/2}.$$

The encryption is illustrated on Figure 3.
By ML_K we denote the following function:

$$ML_K(A\|C, N) = \sum_{i=1}^{h} H_i \otimes A_i \oplus \sum_{i=1}^{q} H_{h+i} \otimes C_i \oplus H_{h+q+1} \otimes (l(A)\|l(C)).$$

At the end of the section let's make some remarks on PD–mode:

- There must be difference between nonces for each message. But nonce need not be a random vector.
- Associated data and plaintext are not mandatory input for scheme. If there is no assosiated data input we deal with AE. If there is no plaintext input we deal with MAC.
- Number of blocks of associated data or plaint text must be less than $2^{n/2}$ for avoiding counter's overlap.

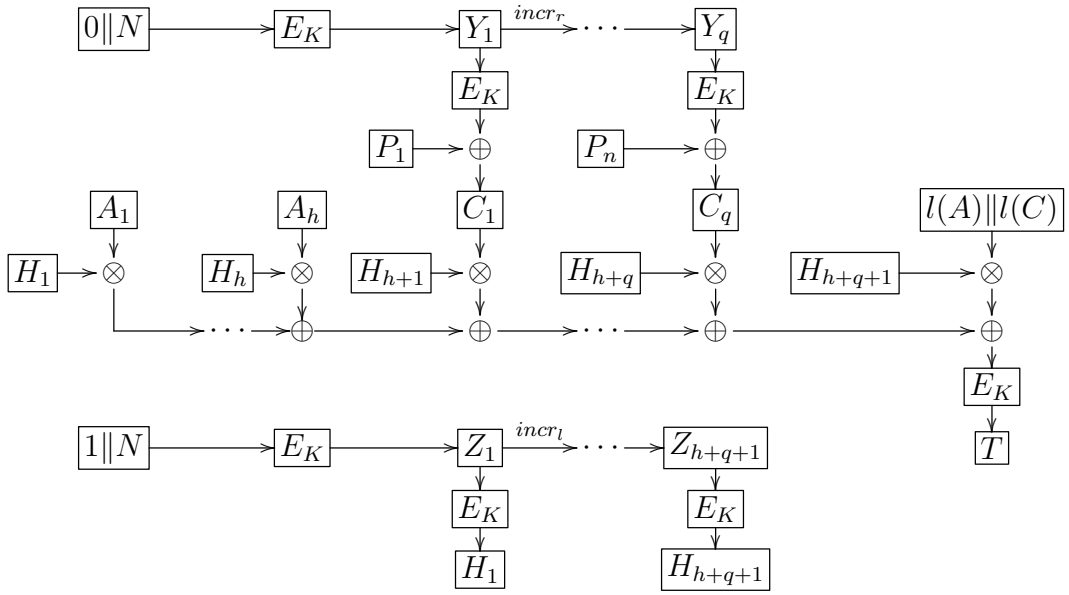

Figure 1: PD–mode encryption

4 Provable security

In this section we will prove two theorems describing th security of PD–mode. The main result on provable security of block cipher modes stated in [5, 6]. In the rest of the paper we will be used basic notations stated in mentioned articles.

4.1 Security of encryption

Let's prove the theorem about security of encryption under the condition that E_K is a set of random substitutions E. Let $CTR_r(N, P)$ be the encryption function, where N is nonce and P is plaintext. Proof of the theorem is similar to [5, Theorem 10].

Theorem 1 *Let E be a set of random permutations on $GF(2^n)$. Then, for any $t, q \in \mathbb{N}$ and $\mu < n \cdot 2^{n/2}$*

$$Adv^{lor-cpa}_{CTR_r}(\cdot, t, q, \mu) \leq \frac{(q-1)\mu}{n \cdot 2^{n-1}} - \frac{q(q-1)}{2^{n-1}}. \qquad (1)$$

□ Let $(S_1, U_1), \ldots, (S_q, U_q)$ be the oracle queries of the adversary A, each consists, by definition, of two equal length messages. Let l_i be the number of

block of length n in i'th query, $i = 1, 2, \ldots, q$. Let $N_1, \ldots, N_q \in V_{n-1}$ - be the nonces associated to queries $(S_1, U_1), \ldots, (S_q, U_q)$. Nonces are chosen at random by the oracle such that $N_i \neq N_j,\ i \neq j$.

Let $\triangle_j^{(i)} = E\left(E(0\|N_i) + j\right),\ i = 1, 2, \ldots, q,\ j = 0, 1, \ldots, l_i - 1$. Since E is random permutation, it follows that the values $E(0\|N_i)$ are different. By *collision* we assume pairs (i, j), (p, t), where $i, p \in \{1, 2, \ldots, q\},\ j \in \{0, 1, \ldots, l_i - 1\}, t \in \{0, 1, \ldots, l_p - 1\}, (i \neq p)$ such that $\triangle_j^{(i)} = \triangle_t^{(p)}$. Let D be the event when no collision occurs. Let $P_b[\cdot]$, $b = 0, 1$, be the probability of an event under condition that oracle chose value b. By definition of adversary's advantage we have

$$Adv_{CTR_r,A}^{lor-cpa}(\cdot) = P_1\left[A = 1\right] - P_0\left[A = 1\right].$$

Using the theorem of total probability, we get

$$\begin{aligned} Adv_{CTR_r,A}^{lor-cpa}(\cdot) = {} & P_1[A = 1/D] \cdot P_1[D] + P_1[A = 1/\overline{D}] \cdot P_1[\overline{D}] - \\ & - P_0[A = 1/D] \cdot P_0[D] - P_0[A = 1/\overline{D}] \cdot P_0[\overline{D}]. \end{aligned}$$

It's easy to see that $P_1[\overline{D}] = P_0[\overline{D}]$, thus we have $P_1[D] = P_0[D]$, since these probabilities depend on random values $E(N_i)$. The following probabilities $P_1[A = 1/D]$ and $P_0[A = 1/D]$ are equal. It now follows that

$$Adv_{CTR_r,A}^{lor-cpa}(\cdot) = \left(P_1[A = 1/\overline{D}] - P_0[A = 1/\overline{D}]\right) \cdot P_1[\overline{D}].$$

Finally, we obtain

$$Adv_{CTR_r,A}^{lor-cpa}(\cdot) \leq P_1[\overline{D}].$$

Let's get the upper bound for $P[\overline{D}]$. By p_i denote the event that collision occurs on i'th query $i,\ i = 1, 2, \ldots, q$. It is obvious that $p_1 = 0$. Let us evaluate the probability p_2. Collision occurs under condition $E(0\|N_1) + s = E(0\|N_2) + t$, where $s < l_1,\ t < l_2$. Let us remark that if it is true and $z = \min\{s, t\}$, then we have $E(0\|N_1) + s - j = E(0\|N_2) + t - j$, for all $j = 1, 2, \ldots, z$. Collision occurs if $E(0\|N_2)$ equals to one of $l_1 - 1$ points $(E(0\|N_1)+1, \ldots, E(0\|N_1)+l_1-1)$ ($E(0\|N_2)$ can not be equal $E(0\|N_1)$) or equals one of $l_2 - 1$ points preceding point $E(0\|N_1)$. Probability of this event equals to $\frac{l_1+l_2-2}{2^{n-1}}$. Now we could evaluate the upper bound $p_i,\ i = 3, \ldots, q$.

This probability is equal to the probability of the following two events. First one is that point $E(0\|N_i)$ equals to one of $l_j - 1$ point of sequences generated by $E(0\|N_j)$ $j = 1, 2, \ldots, i$. Second one is that point $E(0\|N_i)$ equals to one of $l_i - 1$ points preceding $E(0\|N_j)$ $j = 1, 2, \ldots, i$. That's why we have

$$p_i \leq \frac{1}{2^{n-1}} \sum_{j=1}^{i-1} (l_j - 1) + (i-1)(l_i - 1),$$

$i = 2, 3, \ldots, q$. Therefore

$$\begin{aligned} P[\overline{D}] \leq \sum_{i=1}^{q} p_i \leq \sum_{i=1}^{q} \frac{1}{2^{n-1}} \sum_{j=1}^{i-1} (l_j - 1) + (i-1)(l_i - 1) = \\ = \frac{1}{2^{n-1}} \sum_{i=1}^{q} \left(\sum_{j=1}^{i-1} (l_j - 1) + (i-1)(l_i - 1) \right) = \\ = \frac{\sum\limits_{i=1}^{q} \left(\sum\limits_{j=1}^{i-1} l_j + l_i(i-1) \right) - q(q-1)}{2^{n-1}} = \\ = \frac{\sum\limits_{i=1}^{q} (q-1)l_i - \frac{q(q-1)}{2}}{2^{n-1}} = \frac{\frac{(q-1)\mu}{n} - q(q-1)}{2^{n-1}}. \square \end{aligned}$$

Now, let us obtain the upper bound of adversary's advantage, assuming that E is a block cipher E_K. Note that block cipher E_K could be considered as class of permutations $\{E_K : V_n \to V_n \,|K \in V_k\}$.

Theorem 2 *Let E_K be a block cipher, $K \in \mathcal{K}$, then, for any t, q and $\mu = q' \cdot n$*

$$Adv_{CTR}^{lor-cpa}(\cdot, t, q, \mu) \leq 2 \cdot Adv_{E_K}^{prp}(t, q') + \frac{(q-1)\mu}{n \cdot 2^{n-1}} - \frac{q(q-1)}{2^{n-1}}. \qquad (2)$$

$\square$ The proof is similar to [5, Theorem 11] and is omitted. $\square$

4.2 Security of authentication

Recall the condition on nonce N: it must be changed for every message. Therefore it changes function $h \in ML_K$, since the function is defined by

nonce N. Security of proposed scheme is stated in the following theorem.

Theorem 3 *For adversary with forgery advantage Adv_{Π}^{auth} against PD–mode and with distinguishing advantage Adv^{prp} against pseudorandom permutation E_K, the following inequality holds*

$$Adv_{\Pi}^{auth}(t, q, \mu) \leq Adv_{E_K}^{prp}(t, q) + \frac{1}{2^{127}}.$$

□ We build an adversary B, using A, that has an advantage $Adv_{E_K}^{prp}$ in distinguishing E_K from random permutation E. By definition:

$$Adv_B^{prp-cpa} = P\{B^{E_K(\cdot)} = 1/K \leftarrow \mathcal{K}\} - P\{B^{E(\cdot)} = 1/E \leftarrow Perm(V_{2^n})\}.$$

B runs A and tries to see whether A forges the scheme. B use encryption oracle to answer on A queries. Before attack oracle makes a choice between pseudorandom permutation E_K and random permutation E to encrypt every query. If A is successfull – B chooses E_K, and E otherwise.

We have $P\{B^{E_K(\cdot)} = 1/K \leftarrow \mathcal{K}\} = Adv_{\Pi}^{auth}$. That's why we need to evaluate the following probability $P\{B^{E(\cdot)} = 1/E \leftarrow Perm(V_{2^n})\}$.

Adversary chooses $N \in V_{127}$ for each query. After that it determines $H_i = E(E(1\|N) + i - 1)$. Elements H_i, $i \geq 1$ define multilinear function f_N. Probability $P\{B^{E(\cdot)} = 1/E \leftarrow Perm(V_{2^n})\}$ is less or equal advantage in the forgery attack on our scheme, where set of block ciphers E_K replaced by set of random permutations E.

Lemma 4 *Lets replace block cipher E_K by random permutation E in our scheme. Then the forgery advantage against our scheme with number of queries to encryption oracle less then 2^{n-1} and q queries to decryption oracle is $\frac{1}{2^{n-1}}$.*

□ Let us prove that probability to determine tag T_1 for message S_1 by known pair (S_2, T_2) is equal to the probability to guess the value of nonce. Let W be the following event: $Y_q^{(N_1)} = Y_p^{(N_2)}$, where $Y^{(N_j)}$ are counters that obtained from nonce N_j, $j = 1, 2$, $q, p \in \mathbb{N}$.

$$P[f_{N_1}(S_1) = T_1 / f_{N_2}(S_2) = T_2] =$$
$$= P[f_{N_1}(S_1) = T_1 / (f_{N_2}(S_2) = T_2) \cap \overline{W}] P[\overline{W}] +$$
$$+ P[f_{N_1}(S_1) = T_1 / (f_{N_2}(S_2) = T_2) \cap W] P[W].$$

Let $\overline{W}$ occurs, then probability to determine value f_{N_1} is less or equal to $\frac{1}{2^n}$. If W occurs, then probability to determine value f_{N_1} is equal to the probability of guessing the value of collision of counters $Y^{(N_j)}$, $j = 1, 2$. Therefore:

$$P[f_{N_1}(S_1) = T_1 / f_{N_2}(S_2) = T_2] =$$
$$= \frac{1}{2^n} + P[\text{compute } \min t : \ E(N_1) + t = E(N_2)].$$

We recall, the adversary does not know the values $E(N_1), E(N_2)$. It now follows that:

$$P[f_{N_1}(S_1) = T_1 / f_{N_2}(S_2) = T_2] \leq \frac{1}{2^{n-1}}.$$

So forgery advantage against our scheme is equal to the probability of guess value N:

$$P[N = X] = \frac{1}{2^{n-1}}. \square$$

Using the lemma we get

$$Adv_{\Pi}^{auth}(t, q, \mu) \leq Adv_{E_K}^{prp}(t, q) + P\{B^{E(\cdot)} = 1 / E \leftarrow Perm(V_{2^n})\},$$

finally, we obtain

$$Adv_{\Pi}^{auth}(t, q, \mu) \leq Adv_{E_K}^{prp}(t, q) + \frac{1}{2^{n-1}}. \square$$

References

[1] David A. McGrew, John Viega, "The Security and Performance of the Galois/Counter Mode (GCM) of Operation", *IACR Cryptology ePrint Archive*, **2004** (2004), 193.

[2] John Mattsson, "Authentication Key Recovery in Galois/Counter Mode (GCM)", *IACR Cryptology ePrint Archive*, **2015** (2015), 477.

[3] Markku-Juhani O. Saarinen, "GCM, GHASH and Weak Keys", *IACR Cryptology ePrint Archive,*, **2011** (2011), 202.

[4] Markku-Juhani Olavi Saarinen, "Cycling Attacks on GCM, GHASH and Other Polynomial MACs and Hashes", *Lecture Notes in Computer Science*, **7549** (2012), 317–330..

[5] Mihir Bellare, Anand Desai,E. Jokipii, Phillip Rogaway, "A Concrete Security Treatment of Symmetric Encryption", 38th Annual Symposium on Foundations of Computer Science, FOCS '97, Miami Beach, Florida, USA, October 19-22, 1997, 394–403.

[6] Mihir Bellare, Phillip Rogaway, "Encode-Then-Encipher Encryption: How to Exploit Nonces or Redundancy in Plaintexts for Efficient Cryptography", Advances in Cryptology - ASIACRYPT 2000, 6th International Conference on the Theory and Application of Cryptology and Information Security, Kyoto, Japan, December 3-7, 2000, Proceedings, 2000, 317–330..

[7] Mohamed Ahmed Abdelraheem and Peter Beelen and Andrey Bogdanov and Elmar Tischhauser, "Twisted Polynomials and Forgery Attacks on GCM", *IACR Cryptology ePrint Archive*, **2015** (2015), 1224.

Equidistant filters based on skew ML-sequnces over fields

Mikhail Goltvanitsa

Abstract

Let p be a prime number, $R = \mathrm{GF}(q)$ be a field of $q = p^r$ elements and $S = \mathrm{GF}(q^n)$ be an extension of R. Let $\check{S}$ be the ring of all linear transformations of the space ${}_RS$. A linear recurring sequence v of order m over the module ${}_{\check{S}}S$ is said to be a *skew linear recurring sequence (skew LRS) of order m over S*. The period $T(v)$ of such sequence satisfies the inequality $T(v) \leq \tau = q^{mn} - 1$. If $T(v) = \tau$ we call v a *skew LRS of maximal period (skew MP LRS)*. Here we investigate periodic properties and linear complexity of the sequence

$$y(i) = v(i)v(i+k) \cdot \ldots \cdot v(i+k(s-1)),\ k, s \in \mathbb{N}_0,\ i \geq 0,$$

where v is a skew MP LRS. On the basis of the obtained results we propose new methods for filtering generators construction based on skew MP LRS.

Keywords: ML-sequence, linear complexity, period, equidistant filter, skew linear recurrence

1 Introduction. Preliminaries

Below $R = \mathrm{GF}(q)$ is a Galois field with identity e, $q = p^r$, $p = \mathrm{char}R$, $S = \mathrm{GF}(q^n), n \geq 2$ is an extension of R with multiplicative group S^* [1]. Let σ be a generator of the group $\mathrm{Aut}(S/R)$ of automorphisms of S over R. Then $\mathrm{ord}\sigma = n$ [1]. Let $\check{S} = S^{\sigma}\langle\sigma\rangle$ be a ring of formal sums $\psi = \sum_{i=0}^{n-1} s_i\sigma^i$, $s_0, \ldots, s_{n-1} \in S$ with standard addition and multiplication, defined by distributivity from the identity $\sigma s = \sigma(s)\sigma,\ \ s \in S$.

Each element $\psi \in \check{S}$ defines a linear transformation of the space ${}_RS$ such that $\psi(s) = \sum_{i=0}^{n-1} s_i\sigma^i(s)$ for every $s \in S$.

Let us set a structure of a left $\check{S}[x]$-module on the set $S^{\langle 1\rangle}$ of all sequences over S by defining the product of a sequence $v \in S^{\langle 1\rangle}$ by a polynomial $A(x) = \sum_{i\geq 0} a_i x^i \in \check{S}[x]$ by the equality

$$A(x)v = w \in S^{\langle 1\rangle} : \quad w(i) = \sum_{j\geq 0} a_i(v(i+j)), \quad i \geq 0.$$

So the structure of a left $\check{S}[x]$-module on $S^{\langle 1\rangle}$ is given. We say that $v \in S^{\langle 1\rangle}$ is a *skew linear recurring sequence* (*LRS*) of order m over S, if it is LRS of order m over the module ${}_{\check{S}}S$ [2], i.e. $\Psi(x)v = 0$ for some monic polynomial $\Psi(x) = x^m - \sum_{j=0}^{m-1} \psi_j x^j \in \check{S}[x]$, called *skew characteristic polynomial of LRS* v, that is $\forall i \in \mathbb{N}_0\ v(i+m) = \sum_{j=0}^{m-1} \psi_j(v(i+j))$. By $L_S(\Psi)$ we denote the set of all skew LRS over S with characteristic polynomial Ψ. If $\Psi(x) \in S[x]$, then we call LRS from $L_S(\Psi)$ *classical LRS*.

It is easily to see that a period of any skew LRS v of order m over S satisfies the inequality: $T(v) \leq \tau = q^{nm} - 1$. If $T(v) = \tau$, then we say that v is a skew LRS of *maximal period* (*MP LRS*) or *maximal length sequence (ML-sequence)*.

Skew MP LRS over Galois rings and finite fields were studied earlier in works [2]-[10]. Systematic researching of skew MP LRS over Galois rings was started in articles [3, 4, 5], where, in particular, the results allowing to construct large classes of such sequences without brute force method were developed. In articles [4], [8], [9], [10] the classes of skew MP LRS oriented toward fast implementation are constructed.

We define the *rank* $\mathrm{rank}_S\, u$ of the sequence $u \in S^{\langle 1\rangle}$ as the degree of its minimal polynomial over S [1], [19]. In paper [3] there shown that the rank of skew MP LRS of order m over S can be in n times greater then the rank of classical MP LRS of the same order. Rank and statistical properties are among the most important cryptographic characteristics of pseudorandom sequences [11]- [13]. Skew MP LRS exhibit good statistical properties, but their ranks are still not high enough. To eliminate this disadvantage in this paper we offer to use the well-known technique, based on nonlinear filtering [11]-[13]. This technique consists in application of non-linear function to the stages of LRS. There is an extensive literature devoted to the description of the upper bounds of filtered sequences ranks (see for example [16] and cited

literature). However, there is not much fundamentally different constructions for which it is possible to obtain the lower bounds of filtered sequences ranks. This paper is dedicated to the study of filtering functions constructions, which allow to obtain pseudorandom sequences with guaranteed big rank and period.

Let v be a skew MP LRS of order m over S. Proposed methods for construction of non-linear filtering functions are based on the results about the sequence of the form

$$y(i) = v(i)v(i+k) \cdot \ldots \cdot v(i+k(s-1)),\ k, s \in \mathbb{N},\ 2 \leq s \leq m. \qquad (1.1)$$

Using the terminology from [17] we say that the sequence y is obtained from LRS v by *equidistantly filtering*. In the case where v is a classic MP LRS the rank of sequence (1.1) was investigated in many papers among which we mention [11], [13], [14], [15], [17], [18]. Also there noted in [15] that sequence (1.1) was investigated by Nechaev V.I., Nechaev A.A. and Kurakin V.L. Most of this works ([13], [14], [17]) devoted to the case where $S = \mathrm{GF}(2)$ or to the case where $S = \mathrm{GF}(2^n)$ [18]. In articles [11] and [15] sequence (1.1) was investigated over the arbitrary finite field, for the case where $s = 2$ and $k = 1$, respectively. Wherein on sequence (1.1) there imposed the restriction $(k, \tau) = 1$, where (k, τ) is the greatest common divisor of k and τ, in all mentioned works, except [11].

In this paper we study the case where v is a skew MP LRS. Wherein we extend the class of sequences studied by refusing the restriction $(k, \tau) = 1$. Along with the case of field of characteristic 2 we investigate the general case. Also we investigate the periodic properties of sequence (1.1).

Below for $i, t \in \mathbb{N}_0$ we use the notations $\overline{i, i+t} = \{i, i+1, \ldots, i+t\}$ and $v[\overline{i, i+t}] = (v(i), v(i+1), \ldots, v(i+t))$.

Theorem 1. ([3]) *A sequence $v \in S^{\langle 1 \rangle}$ is a skew MP LRS of order m if and only if $\forall i \in \mathbb{N}_0 :\ v[\overline{i, i+m-1}] \neq (0, 0, \ldots, 0)$, and there exists a primitive polynomial $F(x) \in R[x]$ of degree mn such that $v \in L_S(F)$.*

Under the denotations of Theorem 1 let θ be a root of the polynomial $F(x)$ in the extension $K = \mathrm{GF}(q^{mn})$ of the field S [1]. Then $F(x)$ has a canonical decomposition over S:

$$F(x) = F_0(x) \cdot \ldots \cdot F_{n-1}(x),\ F_j(\theta_j) = 0, \theta_j = \theta^{q^j},\ j = 0, 1, \ldots, n-1,$$

where $F_0(x), \ldots, F_{n-1}(x)$ are primitive polynomials of degree m over S [1]. Therefore, every $v \in L_S(F)$ has a unique decomposition $v = w_0 + \ldots + w_{n-1}$, where $w_j \in L_S(F_j)$ [19]. Let tr_S^K be a trace-function from the field K onto the field S [1]. Then for every $j \in \overline{0, n-1}$ there exists a unique element $\varepsilon_j \in K$ such that $w_j(i) = \mathrm{tr}_S^K(\varepsilon_j \theta_j^i),\ i \geq 0$, [19]. So we obtain the following decomposition for the LRS v:

$$v(i) = \mathrm{tr}_S^K(\varepsilon_0 \theta^i) + \mathrm{tr}_S^K(\varepsilon_1 \theta^{iq}) + \ldots + \mathrm{tr}_S^K(\varepsilon_{n-1} \theta^{iq^{n-1}}). \tag{1.2}$$

The tuple $(\varepsilon_0, \ldots, \varepsilon_{n-1})$ is called *the defining tuple of factors* of LRS v. The approach to description and construction of skew MP LRS based on defining tuples of factors was proposed in [5].

Further in this paper v is a skew MP LRS with the defining tuple of factors $(\varepsilon_0, \ldots, \varepsilon_{n-1})$, i.e. condition (1.2) is fulfilled.

We put $N(v) = \{j \in \overline{0, n-1} :\ \varepsilon_j \neq 0\}, |N(v)| = n_0$. The rank $\mathrm{rank}_S\, v$ of sequence (1.2) is equal to mn_0 and in the case where $n_0 = 1$ v is a classical MP LRS from the set $L_S(F)$. The methods for construction of skew MP LRS of order m over S of the highest rank mn are described in the papers [5], [8].

For any element $\alpha \in K$ denote via $[R(\alpha) : R]$ the degree of the field extension $R(\alpha)$ over the field R [19]. For every integer $\nu \in \overline{1, m}$ such that $\nu | m$ we use the designation

$$D_s(m, n_0, \nu) = n_0^s \binom{\nu}{s} \left(\frac{m}{\nu}\right)^s + (n_0)_s \left(\binom{m}{s} - \binom{\nu}{s} \left(\frac{m}{\nu}\right)^s \right) + \binom{n_0}{s} m, \tag{1.3}$$

where $(n_0)_s = n_0(n_0 - 1) \cdot \ldots \cdot (n_0 - s + 1)$ if $n_0 \geq s$ and $(n_0)_s = 0$ if $n_0 < s$. We recall that y is a sequence of the form (1.1).

Theorem 2. *Let*

$$[R(\theta^k) : R] = \mu, n | \mu, \nu = \frac{\mu}{n}. \tag{1.4}$$

1. *If* $\mathrm{char} R \neq 2$, $s = 2$, *and* k *satisfies the condition*

$$[R(\theta^{2k}) : R] = [R(\theta^k) : R], \tag{1.5}$$

then

$$\mathrm{rank}_S\, y = \frac{mn_0(mn_0 + 1)}{2}. \tag{1.6}$$

2. In the case where $\mathrm{char}R = 2$*, we have*

$$\mathrm{rank}_S\, y \geq D_s(m, n_0, \nu), \tag{1.7}$$

wherein if $s < q$*, then*

$$\mathrm{rank}_S\, y \geq D_s(m, n_0, \nu) + mn_0. \tag{1.8}$$

3. In the case where $\mathrm{char}R = 2$ *and* $s = 2$ *the following equality is fulfilled:*

$$\mathrm{rank}_S\, y = D_2(m, n_0, \nu) + mn_0. \tag{1.9}$$

As we note above, in the previous papers devoted to the studying of classical LRS there investigated basically the case where $(k, \tau) = 1$. In this case $\mu = mn,\ \nu = m$ and from (1.3) we obtain

$$D_s(m, n_0, \nu) = n_0^s \binom{m}{s} + \binom{n_0}{s} m. \tag{1.10}$$

For classic MP LRS $n_0 = 1$, so we can see from (1.10) and Theorem 2 that the lower bound for $\mathrm{rank}_S\, y$ can be in n^s times greater the corresponding lower bound for $\mathrm{rank}_S\, y$ in the case where v is a the classical MP LRS. Thus, the best lower bounds for $\mathrm{rank}_S\, y$ we obtain when using skew MP LRS v such that $\mathrm{rank}_S\, v = mn$.

Theorem 3. *The period* $T(y)$ *of the sequence* y *satisfies the condition*

$$T(y) | \frac{\tau}{(q-1, s)}. \tag{1.11}$$

If $s = 2$*, then*

$$T(y) = \frac{\tau}{(q-1, 2)}. \tag{1.12}$$

In the case where $\mathrm{char}R = 2$ *and condition* (1.4) *is fulfilled*

1. if $m < s$*, then*

$$\frac{\tau}{(q^n - 1, s)} | T(y); \tag{1.13}$$

2. if $3 \le s < m,\ n_0 \ge 2$, *then*

$$\frac{\tau}{(q^\lambda - 1, s)} | T(y), \tag{1.14}$$

where λ *is equal to greatest common divisor of all elements from the set*

$$\{j_b - j_a :\ j_a, j_b \in N(v),\ a, b \in \overline{1, n_0},\ j_a < j_b\} \cup \{n\}.$$

Consequence 1. *Let under the preconditions of Theorem 2 the inequalities* $\mathrm{char} R = 2,\ 3 \le s < m$ *are fulfilled. Then if any of the conditions*

1. $n_0 \ge 2$ *and* n *is prime;*

2. $n_0 > \frac{n}{2}$*;*

is fulfilled, then the period of sequence y *reaches the maximal value, that is*

$$T(y) = \frac{\tau}{(q-1, s)}. \tag{1.15}$$

The validity of Corollary 1 follows from the fact that if the condition from the first or the second item is fulfilled then the value of λ in Theorem 3 is equal to 1.

From the Consequence 1 we obtain that using skew MP LRS v such that $\mathrm{rank}_S\, v = mn$ is most preferably to obtain the maximum value $T(y)$.

On the basis of the obtained results we propose one class of functions for filtering generators construction based on skew MP LRS. Recall (see for example [1]) that for every function $f : S^m \to S$ there exists a unique representation in the form $f(x_1, x_2, \ldots, x_m) = \sum_{\mathbf{l} \in \overline{0, q^n - 1}^m} a_{l_1 l_2 \ldots l_m} x_1^{l_1} x_2^{l_2} \ldots x_m^{l_m}$, where $\mathbf{l} = (l_1, l_2, \ldots, l_m),\ l_j \in \overline{0, q^n - 1},\ j \in \overline{1, m},\ a_{l_1 \ldots l_m} \in S$ and a *degree* $\deg f$ of f is defined as $\max\{l_1 + l_2 + \ldots + l_m :\ a_{l_1 l_2 \ldots l_m} \ne 0\}$. The degree of zero function is equal to zero.

Let $f : S^m \to S$ be a function such that $\deg f < s$. Consider the sequence

$$z(i) = \sum_{j=0}^{N-1} h_j y(i + j) + f(v(i), v(i+1), \ldots, v(i+m-1)), \tag{1.16}$$

where $h_0, h_1, \ldots, h_{N-1}$ are arbitrary elements from S and not all of them are equal to zero and y is sequence (1.1).

Theorem 4. *Let conditions of Theorem 2 are fulfilled. In the case where* $\mathrm{char}R = 2$

$$\mathrm{rank}_S\, z \geq D_s(m, n_0, \nu) - (N-1). \tag{1.17}$$

In the case where $\mathrm{char}R \neq 2, s = 2$ *and* k *satisfies* (1.5)

$$\mathrm{rank}_S\, z \geq D_2(m, n_0, m) - (N-1). \tag{1.18}$$

The author is grateful to Professor A.A. Nechaev and Professor A.S. Kuzmin for helpful discussions and valuable remarks.

2 Proofs

Bellow we use the designations $\mathbf{j} = (j_1, \ldots, j_s)$, $\mathbf{l} = (l_1, \ldots, l_s)$. We put $J = N(v)^s$ and decompose the sequence y to the sum of binomial sequences [19]:

$$\begin{aligned} y(i) &= v(i)v(i+k)\cdot\ldots\cdot v(i+k(s-1)) = \\ &= \left(\sum_{l=0}^{m-1}\sum_{j\in N(v)} \varepsilon_j^{q^{nl}}\theta^{iq^{nl+j}}\right)\cdot\ldots\cdot\left(\sum_{l=0}^{m-1}\sum_{j\in N(v)} \varepsilon_j^{q^{nl}}\theta^{k(s-1)q^{nl+j}}\theta^{iq^{nl+j}}\right) = \\ &= \sum_{(j_1,\ldots,j_s)\in J}\ \sum_{0\leq l_1,\ldots,l_s\leq m-1}\left(\prod_{a\in\overline{1,s}}\varepsilon_{j_a}^{q^{nl_a}}\theta^{k(a-1)q^{nl_a+j_a}}\right)\theta^{i\sum_{a\in\overline{1,s}}q^{nl_a+j_a}}. \end{aligned}$$

So, the i-th term $y(i)$ of LRS y has the folowing

$$y(i) = \sum_{\substack{0\leq l_1,\ldots,l_s\leq m-1,\ \mathbf{j}\in J\\ nl_a+j_a\leq nl_b+j_b,\ a,b\in\overline{1,s},\ a<b}} c_{\mathbf{jl}}\left(\theta^{q^{nl_1+j_1}+\ldots+q^{nl_s+j_s}}\right)^i, c_{\mathbf{jl}}\in K. \tag{2.1}$$

We define two sets

$$W = \{(l_1,\ldots,l_s),\ l_a\in\overline{0,m-1}, a\in\overline{1,s},\ l_a\neq l_b,\ a\neq b,\ a,b\in\overline{1,s}\},$$

$$W_1 = \begin{cases}\{(j_1,\ldots,j_s)\in J,\ j_a\neq j_b,\ a\neq b, a,b\in\overline{1,s}\}, & s\leq n_0,\\ \varnothing, & s>n_0.\end{cases}$$

Then we have $y(i) = y'(i) + \tilde{y}(i) + y''(i)$, where

$$y'(i) = \sum_{\mathbf{j}\in J} \sum_{\mathbf{l}\in W} \left(\prod_{a\in\overline{1,s}} \varepsilon_{j_a}^{q^{nl_a}} \right) \cdot \theta^{kq^{nl_2+j_2}} \dots \theta^{k(s-1)q^{nls+js}} \theta^{i\sum_{a\in\overline{1,s}} q^{nla+ja}}, \quad (2.2)$$

$$\tilde{y}(i) = \begin{cases} \sum\limits_{\mathbf{j}\in W_1} \sum\limits_{l=0}^{m-1} \left(\prod\limits_{a\in\overline{1,s}} \varepsilon_{j_a}^{q^{nl}} \right) \theta^{kq^{nl+j_2}} \dots \theta^{k(s-1)q^{nl+js}} \theta^{i\sum_{a\in\overline{1,s}} q^{nl+ja}}, & s \le n_0, \\ 0, & s > n_0; \end{cases} \quad (2.3)$$

$$y''(i) = y(i) - y'(i) - \tilde{y}(i).$$

Lemma 5. *Minimal polynomials over K of the sequences y', $\tilde{y}$ and y'' are coprime.*

□ Consider the case when $s \le n_0$. The sequences y', $\tilde{y}$ and y'' are binomial sequences of the first order. So to prove Lemma 5 it is sufficiently to show that the sets of roots of binomial sequences from the corresponding decompositions are disjoint.

Since $n \ge 2$, $s \le m$, we obtain that for every $\mathbf{j} \in J$, $\mathbf{l} \in W$ the following inequalities are valid

$$\sum_{t=1}^{s} q^{nl_t+j_t} \le q^{n-1} \sum_{t=0}^{m-1} q^{nt} = q^{n-1} \frac{q^{mn}-1}{q^n-1} < q^{mn} - 1 = \mathrm{ord}\theta.$$

Further, for every $\mathbf{j} \in W_1$, $l \in \overline{0, m-1}$

$$\sum_{t=1}^{s} q^{nl+j_t} \le q^{n(m-1)} \sum_{t=0}^{n-1} q^{t} = q^{n(m-1)} \frac{q^{n}-1}{q-1} < q^{mn} - 1 = \mathrm{ord}\theta.$$

So, the set of binomial sequences roots from the y' decomposition (2.2) is disjoint with the corresponding set from the $\tilde{y}$ decomposition (2.3).

Now we show that if $s = 2$ or $\mathrm{char}R = 2$, then the set of binomial sequences roots from the y' decomposition is disjoint with binomial sequences roots from the y'' decomposition. If $s = 2, q > 2$, then this follows from the

fact that all binomial sequences roots in decomposition (2.1) are pairwise distinct since

$$\sum_{a=1}^{s} q^{nl_s+j_s} \leq sq^{(m-1)n+n-1} = sq^{mn-1} < q^{mn} - 1 = \mathrm{ord}\theta. \qquad (2.4)$$

Further we consider the case where $\mathrm{char}R = 2$. Assume the contrary. Let there exist the tuples

$$(l_1, \ldots, l_s, j_1, \ldots, j_s), (\tilde{l}_1, \ldots, \tilde{l}_s, \tilde{j}_1, \ldots, \tilde{j}_s),\ l_t, \tilde{l}_t \in \overline{0, m-1},\ j_t, \tilde{j}_t \in N(v)$$

with the property $l_a \neq l_b$ under $a \neq b, a, b \in \overline{1, s}$, such that:

$$q^{nl_1+j_1} + \ldots + q^{nl_s+j_s} \equiv q^{n\tilde{l}_1+\tilde{j}_1} + \ldots + q^{n\tilde{l}_s+\tilde{j}_s} \pmod{q^{mn} - 1}. \qquad (2.5)$$

Since $\mathrm{charS} = 2$ congruence (2.5) has the form

$$2^{t_1} + \ldots + 2^{t_s} \equiv 2^{\tilde{t}_1} + \ldots + 2^{\tilde{t}_s} \pmod{2^h - 1}, \qquad (2.6)$$

where $2^{t_i} = q^{nl_i+j_i},\ 2^{\tilde{t}_i} = q^{n\tilde{l}_i+\tilde{j}_i}$ for all $i \in \overline{1, s}$, $h = mn\log_2 q$, and $t_i \neq t_j$ under $i \neq j, i, j \in \overline{1, s}$. If $\tilde{t}_i \neq \tilde{t}_j$ for all $i, j \in \overline{1, s}$, $i \neq j$, then congruence (2.6) is equivalent to the equality $2^{t_1} + \ldots + 2^{t_s} = 2^{\tilde{t}_1} + \ldots + 2^{\tilde{t}_s}$, which is impossible because of the condition $(t_1, \ldots, t_s) \neq (\tilde{t}_1, \ldots, \tilde{t}_s)$.

If there exist $i, j \in \overline{1, s},\ i \neq j$ with the property $\tilde{t}_i = \tilde{t}_j$, then grouping equal terms in the right side of congruence (2.6) and replacing, if necessary, 2^h by 1, we obtain the equality $2^{t_1} + \ldots + 2^{t_s} = 2^{\hat{t}_1} + \ldots + 2^{\hat{t}_w}$, where $w < s$ and $\hat{t}_i \neq \hat{t}_j$ under $i \neq j$. This is a contradiction. The coprimeness of minimal polynomials of the sequences $\tilde{y}$ and y'' established similarly.

In the case where $s > n_0$, $\tilde{y}$ is zero-sequence and its minimal polynomial is equal to e. So, in this case it is sufficiently to prove the coprimeness of minimal polynomials of the sequences y' and y'', what has been done above. Lemma 5 is proved. $\square$

2.1 Proof of Theorem 2

Firstly we prove item 2. We note that $y', \tilde{y}$ are the sequences over K. So, from condition $S \subset K$ we obtain:

$$\mathrm{rank}_K\, y' \leq \mathrm{rank}_S\, y', \mathrm{rank}_K\, \tilde{y} \leq \mathrm{rank}_S\, \tilde{y}. \qquad (2.7)$$

In the following two lemmas we estimate the ranks of the sequences y' and $\tilde{y}$ as LRS over K, and then use the corresponding inequality from (2.7).

Lemma 6. *The following inequality is fulfilled*

$$\operatorname{rank}_K y' \geq n_0^s \binom{\nu}{s} \left(\frac{m}{\nu}\right)^s + (n_0)_s \left(\binom{m}{s} - \binom{\nu}{s} \left(\frac{m}{\nu}\right)^s \right). \tag{2.8}$$

□ We put $L = \{(l_1, \ldots, l_s) : \ 0 \leq l_1 < \ldots < l_s \leq m-1\}$. Using (2.2) we obtain the decomposition

$$y'(i) = \sum_{\mathbf{j} \in J} \sum_{\mathbf{l} \in L} c'_{\mathbf{jl}} \theta^{(q^{nl_1+j_1}+\ldots+q^{nl_s+j_s})i}, \ c'_{\mathbf{jl}} \in K. \tag{2.9}$$

Let $\mathcal{P}(\overline{1,s})$ be the set of all permutations of the set $\overline{1,s}$ [19]. Sequence (2.9) is the sum of binomial sequences of the first order with distinct roots, wherein using (2.2) we get that the coefficient $c'_{\mathbf{jl}}$ is equal to

$$c'_{\mathbf{jl}} = \sum_{(\rho_1 \ldots, \rho_s) \in \mathcal{P}(\overline{1,s})} \varepsilon_{j_{\rho_1}}^{q^{nl\rho_1}} \cdot \ldots \cdot \varepsilon_{j_{\rho_s}}^{q^{nl\rho_s}} \theta^{kq^{nl\rho_2+j\rho_2}+\ldots+k(s-1)q^{nl\rho_s+j\rho_s}}. \tag{2.10}$$

Since $\operatorname{char} R = 2$ using (2.10) we obtain

$$c'_{\mathbf{jl}} = \begin{vmatrix} \varepsilon_{j_1}^{q^{nl_1}} & \varepsilon_{j_2}^{q^{nl_2}} & \ldots & \varepsilon_{j_s}^{q^{nl_s}} \\ \varepsilon_{j_1}^{q^{nl_1}} \theta^{kq^{nl_1+j_1}} & \varepsilon_{j_2}^{q^{nl_2}} \theta^{kq^{nl_2+j_2}} & \ldots & \varepsilon_{j_s}^{q^{nl_s}} \theta^{kq^{nl_s+j_s}} \\ \ldots & \ldots & \ldots & \ldots \\ \varepsilon_{j_1}^{q^{nl_1}} \theta^{k(s-1)q^{nl_1+j_1}} & \varepsilon_{j_2}^{q^{nl_2}} \theta^{k(s-1)q^{nl_2+j_2}} & \ldots & \varepsilon_{j_s}^{q^{nl_s}} \theta^{k(s-1)q^{nl_s+j_s}} \end{vmatrix}. \tag{2.11}$$

For any elements $\xi_1, \ldots, \xi_t \in K$ denote via $V(\xi_1, \ldots, \xi_t)$ Vandermonde's determinant, that is $V(\xi_1, \ldots, \xi_t) = \det(\xi_j^{i-1})_{i,j=1}^t$.

Recall (see for example [19]) that condition $V(\xi_1, \ldots, \xi_t) \neq 0$ is equivalent to condition $\xi_a \neq \xi_b, \ a, b \in \overline{1,t}, \ a \neq b$.

Using (2.11) we have $c'_{\mathbf{jl}} = \varepsilon_{j_1}^{q^{nl_1}} \cdot \ldots \cdot \varepsilon_{j_s}^{q^{nl_s}} V(\theta^{kq^{nl_1+j_1}}, \ldots, \theta^{kq^{nl_s+j_s}})$.

So, coefficient (2.10) is not a zero iff

$$\theta^{kq^{nl_a+j_a}} \neq \theta^{kq^{nl_b+j_b}}, \ a, b \in \overline{1,s}, \ a \neq b. \tag{2.12}$$

We consider two cases. In the case where all coordinates of vector $\mathbf{j}$ are pairwise distinct condition (2.12) if fulfilled for all $\mathbf{l} \in L$. Indeed, condition (2.12) if valid iff

$$\mu \nmid n(l_a - l_b) + (j_a - j_b), a, b \in \overline{1, s},\ a \neq b. \tag{2.13}$$

Using equality $\mu = n\nu$, we obtain that the last condition is true since $n \nmid j_a - j_b$. So, in this case there are exactly

$$(n_0)_s \binom{m}{s} \tag{2.14}$$

non-zero coefficients $c'_{\mathbf{jl}}$ of the form (2.10).

If there exist $a, b \in \overline{1, s}$ with the property $a \neq b$ such that $j_a = j_b$, then arguing similarly and taking into account (2.13) and equality $\mu = n\nu$ we obtain that for the validity of condition (2.12) it is sufficiently that the following condition $\mu \nmid n(l_a - l_b), a, b \in \overline{1, s}$ is valid, that is, l_a and l_b are not congruent modulo ν. There exist exactly $n_0^s - (n_0)_s$ tuples $\mathbf{j} \in J$ with at least two equal coordinates and there is exactly $\binom{\nu}{s}\left(\frac{m}{\nu}\right)^s$ tuples $\mathbf{l} \in L$ with coordinates pairwise non-congruent modulo ν. So, in the case where there exists at least one pair of equal coordinates of $\mathbf{j} \in J$ there are no less then

$$(n_0^s - (n_0)_s)\binom{\nu}{s}\left(\frac{m}{\nu}\right)^s \tag{2.15}$$

non-zero coefficients $c'_{\mathbf{jl}}$ of the form (2.10).

So, $\operatorname{rank}_K y'$ is no less then the sum of values (2.15) and (2.14). Lemma 6 is proved. □

Lemma 7. *The following equality is fulfilled*

$$\operatorname{rank}_K \tilde{y} = \binom{n_0}{s} m. \tag{2.16}$$

□ We note that in the case where $s > n_0$, $\tilde{y}$ — is zero sequence and the equality (2.16) is fulfilled. Consider the case where $s \leq n_0$. Using (2.3) we obtain that the sequence $\tilde{y}$ has the following form

$$\tilde{y}(i) = \sum_{\substack{(j_1,\ldots,j_s)\in J \\ j_1<\ldots<j_s}} \sum_{l=0}^{m-1} \tilde{c}_{l\mathbf{j}} \theta^{(q^{nl+j_1}+\ldots+q^{nl+j_s})i}, \tag{2.17}$$

where the $\tilde{c}_{l\mathbf{j}}$ can be calculated by the formula

$$\tilde{c}_{l\mathbf{j}} = \left(\sum_{(\rho_1 \ldots, \rho_s) \in \mathcal{P}(\overline{1,s})} \varepsilon_{j_{\rho_1}} \cdot \ldots \cdot \varepsilon_{j_{\rho_s}} \theta^{kq^{j\rho_2} + \ldots + k(s-1)q^{j\rho_s}} \right)^{q^{nl}}. \qquad (2.18)$$

Further, since $\mathrm{char} S = 2$ we obtain that coefficient (2.18) is not equal to zero iff the element $\varepsilon_{j_1} \cdot \ldots \cdot \varepsilon_{j_s} V(\theta^{kq^{j_1}}, \ldots, \theta^{kq^{j_s}})$ is not equal to zero. The last condition is equivalent to the pairwise non-congruence of the numbers $j_a, j_b \in N(v)$, where $a \neq b,\ a, b \in \overline{1,s}$ modulo μ. Since $n|\mu$ we have $\mu \geq n$ and $\mu \nmid j_a - j_b \in \{-n+1, -n+2, \ldots, n-1\}$. So, $\mathrm{rank}_K \tilde{y}$ is equal to the number of non-zero coefficients (2.18), that is to the number $\binom{n_0}{s} m$. Lemma 7 is proved. $\square$

Now the validity of (1.7) follows from Lemma 5, Lemma 6 and Lemma 7.

Let us show that under the condition $s < q$ inequality (1.8) is valid. Since $s < q$ we obtain that all the roots of binomial sequences if (2.1) are pairwise distinct (see (2.4)). Further, the coefficient $c_{\mathbf{jl}}$ in decomposition (2.1) for

$$\mathbf{l} = (l, l, \ldots, l), \mathbf{j} = (j, j, \ldots, j), \qquad (2.19)$$

for every $l \in \overline{0, m-1}$, $j \in N(v)$, is equal to $c_{\mathbf{lj}} = (\varepsilon_j)^{sq^{nl}} \theta^{\binom{s}{2} kq^{nl+j}} \neq 0$. There are exactly mn_0 pairs of vectors (2.19). Now the validity of (1.8) follows from the validity (1.7).

Now we prove item 3 of Theorem 2. We need in the following result.

Lemma 8. *Let u be LRS over S and* $\mathrm{rank}_K u = N$. *Then* $\mathrm{rank}_S u = N$.

$\square$ If u — is zero sequence, then its minimal polynomials over S and over K coincides and equal to e. Let $u \neq 0$ and $f(x) = x^N - \sum_{j=0}^{N-1} f_j x^j \in K[x]$ be a minimal polynomial of u, wherein $N \geq 1$. Then $u(i+N) = \sum_{j=0}^{N-1} f_j u(i+j)$ for every $i \in \mathbb{N}_0$. Using the equality $|S| = q^n$ from the last condition we obtain $u(i+N) = \sum\limits_{j=0}^{N-1} f_j^{q^n} u(i+j)$ [1]. If $f(x) \in S[x]$, then Lemma 8 is valid. Assume that $f(x) \notin S[x]$. Choose the maximal value $j \in \overline{0, N-1}$ such that $f_j \notin S$. Then $f_j^{q^n} \neq f_j$ and

$$\left(x^j + (f_j^{q^n} - f_j)^{-1}(f_{j-1}^{q^n} - f_{j-1})x^{j-1} + \ldots + (f_j^{q^n} - f_j)^{-1}(f_0^{q^n} - f_0)\right) \cdot u = 0,$$

in the case where $j \geq 1$, or $e \cdot u = 0$, in the case where $j = 0$. So, the sequence u can be annihilated by a monic polynomial of degree $j < N$ over K. This is a contradiction. Thus, $f(x) \in S[x]$. Lemma is proved. $\square$.

We note that under the condition $s = 2$ regardless of field characteristic the following equality if fulfilled $\mathrm{rank}_K\, y'' = mn_0$. To complete the prove we use Lemma 5, Lemma 6, Lemma 7 and Lemma 8. Item 3 of Theorem 2 is proved.

Now we prove item 1 of Theorem 2.

Lemma 9. *Under condition* (1.5) *the following equality is fulfilled*

$$\mathrm{rank}_K\, y' = n_0^2 \binom{m}{2}. \tag{2.20}$$

$\square$ Since $s = 2$ coefficient (2.10) from decomposition (2.9) is not equal to zero iff $\theta^{kq^{nl_1+j_1}} \neq -\theta^{kq^{nl_2+j_2}}$. We put

$C_1 = \{(\mathbf{j}, \mathbf{l}) \in J \times L :\ \theta^{kq^{nl_1+j_1}} \neq -\theta^{kq^{nl_2+j_2}}\}, C_2 = \{(\mathbf{j}, \mathbf{l}) \in J \times L :\ \theta^{kq^{nl_1+j_1}} \neq \theta^{kq^{nl_2+j_2}}\},$
$C_3 = \{(\mathbf{j}, \mathbf{l}) \in J \times L :\ \theta^{2kq^{nl_1+j_1}} \neq \theta^{2kq^{nl_2+j_2}}\}.$

So, $\mathrm{rank}_K\, y' = |C_1|$. We obtain $|C_1| = (|J| \cdot |L| - |C_2|) + |C_3|$, since $\mathrm{char} R \neq 2$. Using (1.5) we get $|C_2| = |C_3|$. So, $\mathrm{rank}_K\, y' = |C_1| = |J| \cdot |L| = n_0^2 \binom{m}{2}$. Lemma is proved. $\square$

Arguing analogously to the proof of Lemma 9, we get

$$\mathrm{rank}_K\, \tilde{y} = \binom{n_0}{2} m. \tag{2.21}$$

As it was noted above $\mathrm{rank}_K\, y'' = mn_0$. Now using Lemma 8 we get

$$\mathrm{rank}_S\, y = n_0^2 \binom{m}{2} + m \binom{n_0}{2} + mn_0 = \binom{mn_0 + 1}{2}.$$

2.2 Proof of Theorem 3

We show that $T(y) | \frac{q^{mn}-1}{(q-1,s)}$. Let $\Delta = \frac{q^{mn}-1}{q-1}$, then

$$v(i + \Delta) = \mathrm{tr}_S^K \left(\sum_{j=0}^{n-1} \varepsilon_j \theta^{q^j (i+\Delta)} \right) = \mathrm{tr}_S^K \left(\sum_{j=0}^{n-1} \varepsilon_j \theta^{\Delta q^j} \theta^{q^j i} \right). \tag{2.22}$$

Since $\theta^{\Delta} \in R$ from (2.22) we establish

$$v(i+\Delta) = \mathrm{tr}_S^K \left(\sum_{j=0}^{n-1} \varepsilon_j \theta^{\Delta q^j} \theta^{q^j i} \right) = \theta^{\Delta} v(i). \tag{2.23}$$

Using (1.1) and (2.23) we get $y(i+\Delta) = \theta^{s\Delta} y(i)$. So,

$$T(y)|\Delta \cdot \mathrm{ord}\theta^{s\Delta} = \frac{\tau}{q-1} \cdot \frac{q-1}{(q-1,s)} = \frac{\tau}{(q-1,s)}.$$

Now we prove (1.12). Since the minimal polynomials of the sequences y', $\tilde{y}$ и y'' are pairwise coprime, the following relation is valid $T(y'')|T(y)$ [19]. As it was noted above, in the case where $s = 2$ the sequence y'' is the sum of binomial sequences with the roots from the set

$$M(y'') = \{\theta^{2q^{nl+j}},\ l \in \overline{0, m-1},\ j \in N(v)\}.$$

So, the period $T(y'')$ of LRS y'' is equal to the lest common multiple (LCM) of orders of the elements from $M(y'')$, that is to the number $\frac{\tau}{(\tau,2)}$. Now (1.12) follows from the equality $\frac{\tau}{(\tau,2)} = \frac{\tau}{(q-1,2)}$.

Now we prove (1.13). Since the minimal polynomials of the sequences y', $\tilde{y}$ и y'' are pairwise coprime, the following relation is valid $T(y')|T(y)$ [19]. Using (2.9) we obtain that the period $T(y')$ of the sequence y' is equal to the LCM of orders of the elements from the set

$$M(y') = \{\theta^{q^{nl_1+j_1}+\ldots+q^{nl_s+j_s}}, \mathbf{j} \in J, \mathbf{l} \in L, c'_{\mathbf{jl}} \neq 0\}.$$

For $j_1 \in N(v)$ we define the numbers

$$\alpha = q^{j_1} + q^{2n+j_1} + \ldots + q^{ns+j_1}; \beta = q^{n+j_1} + q^{2n+j_1} + \ldots + q^{ns+j_1}.$$

From the proof of Lemma 6 we obtain that $\theta^{\alpha}, \theta^{\beta} \in M(y')$. It is easily to see that $[\mathrm{ord}\theta^{\alpha}, \mathrm{ord}\theta^{\beta}] = \frac{\tau}{(\tau,\alpha,\beta)}$, where by $[k_1, \ldots, k_l]$ we denote the LCM of the numbers $k_1, \ldots, k_l$. Now to prove (1.13) it is sufficiently to note that

$$(\tau, \alpha, \beta) = (\tau, \beta, \beta - \alpha) = (\tau, \beta, q^{j_1}(q^n - 1)) = (q^n - 1, s). \tag{2.24}$$

Now we establish (1.14). For every $j_a, j_b \in N(v)$,such that $j_a < j_b$ we define the numbers

$$\gamma_{ab} = q^{j_a} + q^{n+j_b} + q^{2n+j_a} + \ldots + q^{n(s-1)+j_a}; \tilde{\gamma}_{ab} = q^{j_b} + q^{n+j_b} + q^{2n+j_a} + \ldots + q^{n(s-1)+j_a}.$$

From the proof of Lemma 6 it follows that $\theta^{\gamma_{ab}}, \theta^{\tilde{\gamma}_{ab}} \in M(y')$. It is easily to see that for any fixed $j_a, j_b \in N(v),\ a \neq b$ the following equality is fulfilled

$$[\mathrm{ord}\theta^{\alpha}, \mathrm{ord}\theta^{\beta}, \mathrm{ord}\theta^{\gamma_{ab}}, \mathrm{ord}\theta^{\tilde{\gamma}_{ab}}] = \frac{q^{mn}-1}{(q^{mn}-1, \alpha, \beta, \gamma_{ab}, \tilde{\gamma}_{ab})}.$$

Further, $(q^{mn}-1, \alpha, \beta, \gamma_{ab}, \tilde{\gamma}_{ab}) = ((q^{mn}-1, \alpha, \beta), \gamma_{ab}, \tilde{\gamma}_{ab})$. Thus, using (2.24) we get that the value $(q^{mn}-1, \alpha, \beta, \gamma_{ab}, \tilde{\gamma}_{ab})$ divides

$$(q^n-1, s, \tilde{\gamma}_{ab}-\gamma_{ab}) = (q^n-1, s, q^{j_b}-q^{j_a}) = (q^{(n, j_b-j_a)}-1, s).$$

Looking over all elements $j_a, j_b \in N(v), j_a < j_b$ and arguing similarly we establish (1.14). Theorem 3 is completely proved.

2.3 Proof of Theorem 4

Let $H(x) = \sum_{j=0}^{N-1} h_j x^j$. Then $z(i) = z'(i) + \tilde{z}(i) + z''(i) + \hat{z}(i)$, where

$$\begin{array}{l} z'(i) = H(x)y'(i),\ \tilde{z}(i) = H(x)\tilde{y}(i),\quad z''(i) = H(x)y''(i), \\ \hat{z}(i) = z(i) - z'(i) - \tilde{z}(i) - z''(i). \end{array} \tag{2.25}$$

For every number $k = \sum_{t\geq 0} \nu_s(k) q^k \in \mathbb{N}_0,\ \nu_s(k) \in \overline{0, q-1}$ define its weight $\mathrm{wt}_q(k)$ as arithmetic sum $\mathrm{wt}_q(k) = \sum_{t\geq 0} \nu_t(k)$. Using (1.2) and condition $\deg f < s$ we obtain that the sequence $\hat{z}(i)$ is the sum of binomial sequences of first order with the roots θ^{α}, where $\alpha < \tau,\ \mathrm{wt}_q\alpha < s$. As shown above all roots θ^{β}, where $\beta < \tau$ of non-zero binomial sequences from the decomposition of y' satisfy the equality $\mathrm{wt}_q\beta = s$, and therefore all roots from binomial decomposition of z' satisfy the same equality. In the case where $\tilde{y}$ is non c sequence the same is correct for $\tilde{z}$. For every sequence u over K denote via $m_u(x)$ its minimal polynomial over K. So, we establish the equalities $(m_{z'}(x), m_{\hat{z}}(x)) = e$ and $(m_{\tilde{z}}(x), m_{\hat{z}}(x)) = e$ (we use the designation $(G(x), L(x))$ for the greatest common divisor of polynomials $G(x), L(x) \in K[x]$). Taking into account Lemma 5 we obtain $(m_{z'+\tilde{z}}(x), m_{z''+\hat{z}}(x)) = e$ and therefore $\mathrm{rank}_S\, z \geq \mathrm{rank}_K\, z \geq \mathrm{rank}_K(z'+\tilde{z})$. Using (2.25) we get [19] $\mathrm{rank}_K(z'+\tilde{z}) = \deg m_{z'+\tilde{z}}(x) = \deg \frac{m_{y'+\tilde{y}}(x)}{(m_{y'+\tilde{y}}(x), H(x))}$. In the case where $\mathrm{char}R = 2$ using Lemma 5, Lemma 6 and Lemma 7 we obtain $\deg \frac{m_{y'+\tilde{y}}(x)}{(m_{y'+\tilde{y}}(x), H(x))} \geq \deg m'_y + \deg m_{\tilde{y}}(x) - \deg H(x) \geq D_s(m, n_0, \nu) - (N-1)$.

In the case where $\mathrm{char}R \neq 2, s = 2$ and k satisfies (1.5) using (2.20) and (2.21) we obtain (1.18). Theorem 4 is proved.

References

[1] Lidl R. and Niederreiter H. Finite Fields — Cambridge University Press. 1983, Encyclopedia of Mathematics and its Applications v. 20.

[2] Kurakin V. L., Mikhalev A. V., Nechaev A. A., and Tsypyschev V. N. Linear and polylinear recurring sequences over abelian groups and modules. Journal of Mathematical Sciences — 2000 — V. 102 — No 6. — P. 4598-4626.

[3] *Goltvanitsa M.A., Nechaev A.A., Zaitsev S.N.* Skew linear recurring sequences of maximal period over Galois rings — Fund. Appl. Math. 2011/2012, v. 17, No 3, p. 5-23. CNIT MSU (in Russian). English transl. in Journ. Math. Sci., Nov. 2012, v. 187, No 2, p 115-128, (DOI) 10.1007/s10958-012-1054-2.

[4] *Goltvanitsa M.A., Nechaev A.A., Zaitsev S.N.* Skew LRS of maximal period over Galois rings — Mat. Vopr. Kriptogr. 2013, v 4, No 2, p 59-72.

[5] Goltvanitsa M.A. A construction of skew LRS of maximal period over finite fields based on the defining tuples of factors // Mat Vop Kriptogr — 2014 — V.5 — No 2. — P. 37-46.

[6] Goltvanitsa M.A. Digit sequences of skew linear recurrences of maximal period over Galois rings. // Mat Vop Kriptogr — 2015 — V.6 — No 2. — P. 189-197.

[7] Goltvanitsa M.A. The first digit sequence of skew linear recurrence of maximal period over Galois ring. // Mat Vop Kriptogr — 2016 — V.7 — No 3. — P. 5-18.

[8] Goltvanitsa M.A. About one class of skew linear recurrences of maximal period over Galois rings. // Sistemy visokoi dostupnosti — 2015 — V.11 — No 3. — P. 28-48. (In Russian)

[9] Tsaban B. and Vishne U., Efficient Linear Feedback Shift Registers with Maximal Period — Finite Fields and Their Applications — 2002 — V.8 — No 2 — P. 256-267.

[10] Zeng, G., He, K.C., Han,W. A trinomial type of $\sigma-$LFSR oriented toward software implementation — Science in China Series F-Information Sciences — 2007 — V.50 — No 3 — P. 359-372.

[11] Alferov A.P., Zubov A.U., Kuzmin A.S., Cheremushkin A.V. Osnovi Kriptografii —Moscow 2001. Gelios ARV. (in Russian)

[12] Menezes A.J., Vanstone S.A., Van Oorschot P.C. Handbook of Applied Cryptography. — USA 1996. Boca Raton, CRC Press.

[13] Rueppel R.A. Analysis and design of stream ciphers. — Berlin 1986. Springer-Verlag.

[14] Bernasconi C., Gunter G. Analysis of a Nonlinear Feedforward Logic for Binary Sequence Generators // LNCS — 1986 — V.219 — P. 161-166.

[15] Bogonatov R.V. Gipoteza Nechaeva o proizvedenii lineinih recurrent. // Chebyshevskii sbornik — 2005 — V.6 — No 1. — P. 48-55. (In Russian)

[16] Kurakin V.L. Polynomial transformations of linear recurrent sequences over the ring $\mathbf{Z}_{p^2}$. // Discr mat and Appl — 1999 — V.9 — No 2. — P. 185-210. (In English) Diskretnaia Matematika — 1999 — V.11 — No 2. — P. 40-65. (In Russian)

[17] Kolokotronis N., Liminiotis K., Kaloupsidis N. Lower bounds on sequence complexity via generalised Vandermonde determinants // LNCS — 2006 — V.4086 — P. 271-284.

[18] Lam C., Gong G. A lower bound for the linear span of filtering sequences // SASC —2004 — P. 220-233.

[19] Glukhov M. M., Elizarov V. P., Nechaev A. A. Algebra. — Moscow 2003. Gelios ARV. (In Russian).

An Approach to Studying Periods of Binary Digit-position Sequences over Prime Rings

Alexey Kuzmin

Abstract

We study how the subgroups of group of multipliers of linear recurring sequences of maximal period (LRS MP) over prime rings influence the period of some fixed binary digit-position sequence.

Keywords: linear recurring sequences of maximal period, binary digit-position sequences, prime rings, finite prime fields, period of sequence.

1 Introduction

A special interest in recent years can be observed in studying p-adic digit-position sequences over residue ring modulo p^n, where $p > 2$ is a prime number. This is due to fact that these sequences possesses high linear complexity, hence they can be used in random-number generators. A list of papers on this thematic can be seen in [1].

A lot of papers are dedicated to reconstruction of LRS over prime residue rings from its images, especially when LRS MP over residue ring is mapped into the highest order p-adic digit-position sequence [2].

This paper is dedicated to less studied object of r-ary digit-position sequences over prime fields and residue rings where $r \neq p$. Such digit-position sequences were studied by Kuzmin A.S. in paper [3]. He has found all binary digit position-sequences over finite prime fields, which admit the effect of reduction of period. Researches of Kuzmin A.S. were extended in paper [4] where the author proved that the period of r-ary digit-position sequences where $r \geq 3$ equals the period of forming LRS MP. Studying of

digit-position sequences over Galois-rings is more difficult due to fact that the function which returns some fixed digit-position can't be represented as a polynomial over Galois-ring. In paper [5] a number of binary digit-position sequence over Galois-rings which admits the twofold reduction of a period was found. Moreover the author showed that twofold reduction of a period doesn't exist for digit-position sequences with other numbers. In [6] a sufficient condition when there is no twofold reduction of period in high order binary digit-position sequences was found. It happens when not all the elements of $\mathbb{Z}_{p^n}$ appears on a cycle of LRS MP.

2 Definitions

Let $\mathbb{Z}_{p^n}$, be a primary ring with the generator polynomial $F(x)$, $\deg F(x) = m$, notably $u(i) = (u(i))_{i=0}^{\infty}$ is LRS MP over this ring. Period $T(u)$ of LRS MP u equals to $p^{n-1}(p^m - 1)$ [7].

Every element $u(i)$ of some LRS MP u over prime ring can be uniquely represented as follows

$$u(i) = \sum_{t=0}^{k} u_t(i)2^t,$$

where $k = [\log_2 p^n]$.

Sequence $u_t, t = \overline{1,k}$ is called t^{th} binary digit-position sequence.

From [7] it is known that the property $T(u_t)|T(u)$ holds.

Multiplier of a sequence u is an element $c \in \mathbb{Z}_{p^n}^*$, for which there exists $q \in \mathbb{N}$ with property $x^q u = cu$ [7].

Let $c \in \mathbb{Z}_{p^n}$ be a multiplier of a sequence u over $\mathbb{Z}_{p^n}$. Let $M(u)$ be a set of all of multipliers of u. $M(u)$ forms a subgroup in $\mathbb{Z}_{p^n}^*$.

Let $H = \{1, \beta, \beta^2, \ldots, \beta^{2d-1}\}$ be a subgroup of $M(u)$, here β is a forming element of group H, value $2d$ satisfies condition

$$2d|GCD(T(u), |\mathbb{Z}_{p^n}^*|) = p^{n-1}(p-1).$$

The set of $\mathbb{Z}_{p^n} \setminus \{0\}$ can be represented as a decomposition of non-intersecting classes $g_j H$ for some $g_j \in \mathbb{Z}_{p^n} \setminus \{0\}$

As β is a multiplier, then for elements of u it holds that $u(i+\frac{jT(u)}{2d}) = \beta^j u(i)$, where $j = \overline{0, 2d-1}$. So the existence of groups of multipliers allows us to study the reduction of a period of digit-position sequences of u if we study digit-positions in classes gH, where $g \in \mathbb{Z}_{p^n}$ and $H < M(u)$.

A value of $p = \sum\limits_{t=0}^{[\log_2 p]} p_t 2^t$, where $p_t \in \{0,1\}$, can be represented as follows:

$$p = \sum_{t=z+1}^{[\log_2 p]} p_t 2^t + \sum_{t=0}^{z-1} 2^t,$$

number z — is a first appearance of 0 in binary representation of p. Let $a(z) = \sum\limits_{t=z+1}^{[\log_2 p]} p_t 2^{t-z-1}$, so we obtain the following representation for value of $p = a(z)2^{z+1} + 2^z - 1$. Analogously we can consider values of s_j where $j = \overline{1,3}$, the first appearance of 0 in binary digit-positions of numbers p^{n-2}, p^{n-1}, p^n and obtain the following expressions

$$p^{n-2} = a^{(1)}(s_1)2^{s_1+1} + 2^{s_1} - 1,$$

$$p^{n-1} = a^{(2)}(s_2)2^{s_2+1} + 2^{s_2} - 1,$$

$$p^{n} = a^{(3)}(s_3)2^{s_3+1} + 2^{s_3} - 1.$$

Here $a^{(j)}(s_j)$ represents sums of digit-positions with number greater than s_j for $j = \overline{1,3}$.

Note that if the value of power of p is even, then $s_j = 1$, otherwise $s_j = z$.

We will show how the obtained approach can be used. As the example we will study the period of binary digit-position sequence with number s. It is due to fact that according to [5], the period of digit-position sequence with that number is at least 2 times shorter than for other sequences. That is why it is interesting to study if it is possible to reduce the period of sequence u_s in more than two times.

We will need the following definitions and representations $pH = \{p, \alpha_1, ..., \alpha_{2d-1}\}$, where $\alpha_j \equiv p\beta^j \pmod{p^n}, j = \overline{1, 2d-1}$ and let $r_{2^{s+1}}(x)$ be a remainder of division x by 2^{s+1}. More over, we will point out the first

digit-position which equals to 0 in numbers

$$\beta = \omega(l)2^{l+1} + 2^l - 1,$$

$$\alpha_j = \gamma(l_j)2^{l_j+1} + 2^{l_j} - 1, j = \overline{1, 2d-1},$$

here $\omega(l) \geq 0, \gamma(l_j) \geq 0, j = \overline{1, 2d-1}$.

3 Results of investigations.

The proof of the main result is divided into 5 lemmas.

Lemma 1. *Let u be an LRS MP over $\mathbb{Z}_{p^n}$, all the elements of $\mathbb{Z}_{p^n}$ occure in the cycle of u, u_1 be the 1^{st} digit-position sequence of u. Let $H = \{1, \beta, \beta^2, \beta^3\} < M(u)$, $s = 1$, $p^n = a^{(3)}(1)4 + 1, p \geq 3$. Then the following expression holds*

$$T(u_1) \not| \frac{T(u)}{4}.$$

Proof.

Let $\beta = \omega(l)2^{l+1} + 2^l - 1$. We recall that in conditions of Lemma 1 $p - 1 \equiv 0 \pmod 4$. We suppose that $(gH)_s = const$ for each $g \in \mathbb{Z}_{p^n}$. We shall study 2 variants.

1) $\beta < p^{n-1}$.

a) If $l > 1$, it is obvious that $(1)_1 = 0 \neq (\beta)_1 = 1$.

b) If $l = 1$, then there is $t = min\{j | 2^j\beta > p^n\}$, as $\beta < p^{n-1}$ then $t > 1$. Element $2^t\beta$ can be represented in the form of $2^t\beta = 2^t(\omega(1)4+1) - a^{(3)}4 - 1$. So class 2^tH will lead us to a contradiction with the assumption because $(2^t)_1 = 0 \neq (2^t\beta)_1 = 1$.

c) If $l < 1$, class $2H$ will lead us to a contradiction with the assumption because $(2)_1 = 1 \neq (2\beta)_1 = 0$.

2) $\beta \geq p^{n-1}$. We will study class $pH = \{p, \alpha_1, -p, -\alpha_1\}$.

Let $\alpha_1 = \gamma(r)2^{r+1} + 2^r - 1$.

If $(p)_1 = 0$.

a) If $r > 1$, it is obvious that $(p)_1 = 0 \neq (\beta)_1 = 1$.

b) Class $(\alpha_1 - p)H = \{\alpha_1 - p, p^n - \alpha_1 - p, p^n - \alpha_1 + p, \alpha_1 + p\}$ will lead us to a contradiction with the assumption because $(\alpha_1 - p)_1 \neq (\alpha_1 + p)_1$ for $r \leq 1$.

Let now $(p)_1 = 1$.

As $z > 1$, class $(\alpha_1 - p)H = \{\alpha_1 - p, p^n - \alpha_1 - p, p^n - \alpha_1 + p, \alpha_1 + p\}$ will lead us to a contradiction with the assumption because $(\alpha_1 - p)_1 \neq (\alpha_1 + p)_1$ for $r \leq 1$ regardless the value of r.

So we always can find class gH which contradicts the condition $(gH)_1 = const$.

Lemma 2. *Let u be an LRS MP over $\mathbb{Z}_{p^n}$, all the elements of $\mathbb{Z}_{p^n}$ occure in the cycle of u, u_1 be the 1^{st} digit-position sequence of u. Let $H < M(u)$, $|H| = 2d$, $p^n = a^{(3)}(1)4 + 1, p \geq 3$. Then the following expression holds*

$$T(u_1) \not| \frac{T(u)}{2d}.$$

Proof. Recall that $p = a(z)2^{z+1} + 2^z - 1$ and not necessarily $z = 1$.

Let $H = \{1, \beta, \beta^2, \ldots, \beta^d = p^n - 1, \ldots, \beta^{2d-1}\}$,

$pH = \{1, \alpha_1, \alpha_2, \ldots, \alpha_d = p^n - p, \ldots, \alpha_{2d-1}\}$.

Let $\alpha_j = \gamma(l_j)2^{l_j+1} + 2^{l_j} - 1, j = \overline{1, 2d-1}$.

We suppose that for every class $gH, g \in \mathbb{Z}_{p^n}$ it holds that $(gH)_1 = const$.

If $\beta < p^{n-1}$ the proof of Lemma 2 is equivalent to the proof of analogical part of Lemma 1, so let $\beta \geq p^{n-1}$.

We will study class $(\alpha_1 - p)H = \{\alpha_1 - p, \alpha_2 - p, \ldots, -p - \alpha_{d-1}, p - \alpha_1, \ldots, p + \alpha_{d-1}\}$.

Let $z = 1$.

As $(pH)_1 = 0$ according to conditions of Lemma 2 it is enough to study cases when $l_1, l_{d-1} \leq 1$ otherwise we easily obtain contradiction.

a) Case $l_1 = l_{d-1} = 1$. Then

$\alpha_1 - p = \gamma_1(1)4 + 1 - a(1)4 - 1$,

$\alpha_{d-1} + p = \gamma_{d-1}(1)4 + 1 - a(1)4 - 1$.

So $(\alpha_1 - p)_1 = 1 \neq (p + \alpha_{d-1})_1 = 0$. So class $(\alpha_1 - p)H$ doesn't satisfy condition of lemma 2 in that case.

b) Case $l_1 = l_{d-1} = 0$. As $(\gamma_1(0)2)_1 = (\gamma_{d-1}(0)2)_1 = 0$, then again $(\alpha_1 - p)_1 = 1 \neq (p + \alpha_{d-1})_1 = 0$.

c) Without loss of generality we have to find contradiction with conditions of lemma 2 in case $l_1 = 0, l_{d-1} = 1$.

Let us look through classes $2^t pH$, $t \geq 1$.

The equalities $(2^j p)_1 = (2^j \alpha_1)_1 = (2^j \alpha_{d-1})_1 = 0$, where $j > 1$ are hold only if $2^j \alpha_1 > \frac{p^n-1}{2}$ and $2^j \alpha_{d-1} < \frac{p^n-1}{2}$. Otherwise we will come to contradiction.

Note that $a_{d-1} \geq 3p$ otherwise $(pH)_1 \neq 0$.

There exists $t | 2^t p < \alpha_{d-1} 2^t < \frac{p^n-1}{2}$ and $2^{t+1} p < \frac{p^n-1}{2} < \alpha_{d-1} 2^{t+1}$. Then the following expression for elements $2^{t+2} p, \alpha_{d-1} 2^{t+2} \in 2^{t+2} pH$ holds $(2^{t+2} p)_1 = 0 \neq (\alpha_{d-1} 2^{t+2})_1 = 1$ and we obtain contradiction with conditions of lemma 2.

Let now $z > 1$.

Then $(pH)_1 = 1$ and $(\alpha_1)_1 = (\alpha_{d-1})_1 = 1$ so $l_1, l_{d-1} \neq 1$. In that case $n = 2j, j \in \mathbb{N}$.

a) Case $l_1, l_{d-1} > 1$. Then

$\alpha_1 - p = \gamma_1(l_1) 2^{l_1+1} - 2^{l_1} - 1 - a(z) 2^{z+1} - 2^z + 1$,

$\alpha_{d-1} + p = \gamma_{d-1}(l_{d-1}) 2^{l_{d-1}+1} + 2^{l_{d-1}} - 1 + a(z) 2^{z+1} + 2^z - 1$.

The following expression holds $(\alpha_1 - p)_1 = 0 \neq (\alpha_{d-1} + p)_1 = 1$, so we obtain contradiction.

b) Case $l_1, l_{d-1} = 0$. Then

$\alpha_1 - p = \gamma_1(0) 2 - a(z) 2^{z+1} - 2^z + 1$,

$\alpha_{d-1} + p = \gamma_{d-1}(0) 2 + a(z) 2^{z+1} + 2^z - 1$.

The following expression holds $(\alpha_1 - p)_1 = 1 \neq (\alpha_{d-1} + p)_1 = 0$, so we obtain contradiction.

c) Without loss of generality we have to find contradiction with conditions of Lemma 2 in case $l_1 = 0, l_{d-1} > 1$.

Let us look through classes $2^t pH$, $t \geq 1$.

The equalities $(2^j p)_1 = (2^j \alpha_1)_1 = (2^j \alpha_{d-1})_1 = 0$, where $j > 1$ are hold only if $\alpha_1 > \frac{p^n-1}{2}$ and $\alpha_{d-1} < \frac{p^n-1}{2}$. Otherwise we will come to contradiction.

Note that $a_{d-1} \geq 2p$ because $a_{d-1} \in pH$.

There exists $t | 2^t p < \alpha_{d-1} 2^t < \frac{p^n-1}{2}$ and $2^{t+1} p < \frac{p^n-1}{2} < \alpha_{d-1} 2^{t+1}$. Then the following expression for elements $2^{t+2} p, \alpha_{d-1} 2^{t+2} \in 2^{t+2} pH$ holds $(2^{t+2} p)_1 = 0 \neq (\alpha_{d-1} 2^{t+2})_1 = 1$ and we obtain contradiction with conditions of Lemma 2.

Lemma 3. *Let* $p = a(s)2^{s+1} + 2^s - 1, p^n = a^{(3)}(s)2^{s+1} + 2^s - 1, p \geq 3, s > 1, n = 2k + 1, k \in \mathbb{N}$, *if* $a^{(3)}(s)$ *is even then for* $a^{(1)}(s)$ *such that* $p^{n-2} = a^{(1)}(s)2^{s+1} + 2^s - 1$ *it holds that* $a^{(1)}(s)$ *is odd.*

Proof.

The following expressions hold

$p^{n-1} = p^{n-2}p =$

$$= (a^{(1)}(s)2^{s+1} + 2^s - 1)(a(s)2^{s+1} + 2^s - 1) =$$

$$= a(s)a^{(1)}(s)2^{2s+2} + a(s)2^{2s+1} + a^{(1)}(s)2^{2s+1} + 2^{2s} -$$

$$-2^{s+1} - a^{(1)}(s)2^{s+1} - a(s)2^{s+1} + 1 =$$

$= \delta_1(s)2^{s+2} - (a^{(1)}(s) + a(s) + 1)2^{s+1} + 1$, where $\delta_1(s) \in \mathbb{N}$.

Then $p^n = p^{n-1}p =$

$$= (\delta_1(s)2^{s+2} - (a^{(1)}(s) + a(s) + 1)2^{s+1} + 1)(a(s)2^{s+1} + 2^s - 1) =$$

$$= \delta_2(s)2^{s+2} - (a^{(1)}(s) + a(s) + 1)2^{s+1} + a(s)2^{s+1} + 2^s - 1 =$$

$= \delta_2(s)2^{s+2} - (a^{(1)}(s) + 1)2^{s+1} + 2^s - 1$, where $\delta_2(s) \in \mathbb{N}$.

The value $a^{(1)}(s) + 1$ is even according to conditions of lemma, hence $a^{(1)}(s)$ is odd.

Lemma 4. *Let* $p = a(s)2^{s+1} + 2^s - 1, p^n = a^{(3)}(s)2^{s+1} + 2^s - 1, p \geq 3, s > 1, n = 2k + 1, k \in \mathbb{N}$. *A set* $C = \{c \in p\mathbb{Z}_{p^n} \setminus \{0\} | 0 < c \leq \frac{p^n-1}{2}\}$ *can be represented as follows*

$$C = C_{00} \sqcup C_{10} \sqcup C_{01} \sqcup C_{11},$$

where $C_{ij} = \{c_{ij} \in p\mathbb{Z}_{p^n} \setminus \{0\} | 0 < c_{ij} \leq \frac{p^n-1}{2}, (c_{ij})_{s-1} = i, (c_{ij})_s = j\}, i, j \in 0, 1$ *and*

$$|C_{ij}| = \frac{p^{n-1} - 1}{8}. \qquad (3)$$

Proof. The first part of lemma 4 is obvious. Let us prove (3). Each element $g \in C$ can be represented as $g = k_0 p$ for some $k_0 \in \mathbb{Z}_{p^n}$.

Note that $(r_{2^{s+1}}(k_0 p))_s = (r_{2^{s+1}}(k_0 r_{2^{s+1}}(p)))_s = (r_{2^{s+1}}(k_0(2^s - 1)))_s$

It is obvious that the increase of k_0 by $j2^{s+1}$ makes remainder the same.

Let suppose that there exists $k_1 < k_2 < 2^{s+1}$ such that $r_{2^{s+1}}(k_1 p) = r_{2^{s+1}}(k_2 p)$, let $k_2 = k_1 + x$ then the following equations are hold $r_{2^{s+1}}(k_2 p) = r_{2^{s+1}}(k_2)(2^s - 1) = r_{2^{s+1}}(k_1 + x)(2^s - 1) => x = 0$ and $k_1 = k_2$.

So there are 4 disjoint classes.

The value of $p^{n-1} - 1$ according to proof of Lemma 3 equals $\delta_1(s)2^{s+2} - (a^{(1)}(s) + a(s) + 1)2^{s+1} = a^{(2)}(s)2^{s+1}$, so $|C_{ij}| = \frac{p^{n-1}-1}{8}$.

Lemma 5. *Let u be an LRS MP over $\mathbb{Z}_{p^n}$, all the elements of $\mathbb{Z}_{p^n}$ occure in the cycle of u, u_s be the s^{th} digit-position sequence of u, where s satisfies conditions $p^n = a^{(3)}(s)2^{s+1} + 2^s - 1, s \geq 2$. Let $H < M(u)$, $|H| = 2d$, $p = a(s)2^{s+1} + 2^s - 1, p \geq 3$. Then the following expression holds*

$$T(u_1) \not| \frac{T(u)}{2d}.$$

Proof.

We will study 2 cases.

The first case is when $a^{(3)}(s)$ is odd.

Lets consider the following sets

$$A_{00} = \{a_{00} | 0 < a_{00} \leq \frac{p^n - 1}{2}, (a_{00})_{s-1} = 0, (a_{00})_s = 0\},$$

$$A_{10} = \{a_{10} | 0 < a_{10} \leq \frac{p^n - 1}{2}, (a_{10})_{s-1} = 1, (a_{10})_s = 0\},$$

$$A_{01} = \{a_{01} | 0 < a_{01} \leq \frac{p^n - 1}{2}, (a_{01})_{s-1} = 0, (a_{01})_s = 1\},$$

$$A_{11} = \{a_{11} | 0 < a_{11} \leq \frac{p^n - 1}{2}, (a_{11})_{s-1} = 1, (a_{11})_s = 1\}.$$

It is easy to see that $A = \{a \in \mathbb{Z}_{p^n} \setminus \{0\} | 0 < a \leq \frac{p^n-1}{2}\} = A_{00} \sqcup A_{10} \sqcup A_{01} \sqcup A_{11}$.

If we prove that there exists class gH such that elements $a_{i_1 i_2}, a'_{j_1 j_2} \in gH$, for $(i_1, i_2) \neq (j_1, j_2)$ we obtain a contradiction with existence of reduction of a period in s^{th} digit-position.

There are three possible variants.

1) If $i_2 \neq j_2$ then we obviously come to contradiction because $(a_{i_1 i_2})_s = (a'_{j_1 j_2})_s + 1$.

2) If $i_2 = j_2$ and $i_1 \neq j_1$ we can consider class $2gH$ where $(2a_{i_1 i_2})_s = (2a'_{j_1 j_2})_s + 1$.

3) If $(i_1, i_2) = (j_1, j_2)$ we can not say anything.

Let us show that it is impossible to divide elements from A to obtain classes $gH, g \in \mathbb{Z}_{p^n}$ that fulfills the condition $(gH)_s = const$.

For each class gH we have $|gH| = 2d$ and exactly half of these elements are less or equal to $\frac{p^n-1}{2}$. So the following expression must hold

$$d|GCD(|A_{00}|, |A_{10}|, |A_{01}|, |A_{11}|).$$

Recall that $\frac{p^n-1}{2} = a^{(3)}(s)2^s + 2^{s-1} - 1$.

It is easy to see that

$$\begin{aligned} |A_{00}| &= a^{(3)}(s)2^{s-2} + 2^{s-2} - 1, \\ |A_{10}| &= a^{(3)}(s)2^{s-2} + 2^{s-2}, \\ |A_{01}| &= a^{(3)}(s)2^{s-2} + 2^{s-2}, \\ |A_{11}| &= (a^{(3)}(s) - 1)2^{s-2}. \end{aligned}$$

So $d|GCD(|A_{00}|, |A_{10}|, |A_{01}|, |A_{11}|) = 1$ and we obtain contradiction.

The second case is when $a^{(3)}(s)$ is even.

Lets consider the following sets

$$B_{00} = \{b_{00} | 0 < b_{00} \leq \frac{p^n - 1}{2}, (b_{00})_{s-1} = 0, (b_{00})_s = 0, b_{00} \equiv 0 \pmod{p},$$
$$b_{00} \not\equiv 0 \pmod{p^2}\},$$
$$B_{10} = \{b_{10} | 0 < b_{10} \leq \frac{p^n - 1}{2}, (b_{10})_{s-1} = 1, (b_{10})_s = 0, b_{10} \equiv 0 \pmod{p},$$
$$b_{10} \not\equiv 0 \pmod{p^2}\},$$
$$B_{01} = \{b_{01} | 0 < b_{01} \leq \frac{p^n - 1}{2}, (b_{01})_{s-1} = 0, (b_{01})_s = 1, b_{01} \equiv 0 \pmod{p},$$
$$b_{01} \not\equiv 0 \pmod{p^2}\},$$
$$B_{11} = \{b_{11} | 0 < b_{11} \leq \frac{p^n - 1}{2}, (b_{11})_{s-1} = 1, (b_{11})_s = 1, b_{11} \equiv 0 \pmod{p},$$
$$b_{11} \not\equiv 0 \pmod{p^2}\}.$$

Similarly to the first case we have to show that

$$GCD(|B_{00}|, |B_{10}|, |B_{01}|, |B_{11}|) = 1$$

It is easy to see that

$$B_{ij} = C_{ij} \setminus D_{ij},$$

sets C_{ij} are defined in Lemma 4,

$$D_{ij} = \{d_{ij} \in p^2\mathbb{Z}_{p^n} \setminus \{0\} | 0 < d_{ij}, (d_{ij})_{s-1} = i, (d_{ij})_s = j\},$$

where $i, j \in 0, 1$.

The value of $p^2 = (a(s)2^{s+1} + 2^s - 1)^2 = \delta_3(s)2^{s+1} + 1$.

Each d_{ij} can be represented as follows $d_{ij} = kp^2, k \in \{1, \ldots, \frac{p^{n-2}-1}{2}\}$.

Recall that $\frac{p^{n-2}-1}{2} = a(s)^{(1)}2^s + 2^{s-1} - 1$.

It is obvious that while multiplying k by p^2 only the value of $r_{2^{s+1}}(k)$ affects digit-positions with numbers $s-1$ and s.

So the following conditions are hold

$$p^2k \in D_{00} <=> 0 \leq r_{2^{s+1}}(k) \leq 2^{s-1} - 1,$$
$$p^2k \in D_{10} <=> 2^{s-1} \leq r_{2^{s+1}}(k) \leq 2^s - 1,$$
$$p^2k \in D_{01} <=> 2^s \leq r_{2^{s+1}}(k) \leq 2^s + 2^{s-1} - 1,$$
$$p^2k \in D_{11} <=> 2^s + 2^{s-1} \leq r_{2^{s+1}}(k) \leq 2^{s+1} - 1.$$

Hence recalling that $\frac{p^{n-2}-1}{2} = a(s)^{(1)}2^s + 2^{s-1} - 1$ and taking in consideration that minimal value of k equals 1 we obtain

$$|D_{00}| = a^{(1)}(s)2^{s-2} + 2^{s-2} - 1,$$
$$|D_{10}| = a^{(1)}(s)2^{s-2} + 2^{s-2},$$
$$|D_{01}| = a^{(1)}(s)2^{s-2} + 2^{s-2},$$
$$|D_{11}| = (a^{(1)}(s) - 1)2^{s-2}.$$

In conclusion we obtain the following equations

$$GCD(|B_{00}|, |B_{10}|, |B_{01}|, |B_{11}|) =$$

$= GCD(\frac{p^{n-1}-1}{8} - |D_{00}|, \frac{p^{n-1}-1}{8} - |D_{10}|, \frac{p^{n-1}-1}{8} - |D_{01}|, \frac{p^{n-1}-1}{8} - |D_{11}|) =$
$= GCD((a'(s)-1)2^{s-2}+1, (a'(s)-1)2^{s-2}, (a'(s)-1)2^{s-2}, ((a'(s)+1)2^{s-2}) = 1,$
where $a'(s) = a^{(2)}(s) - a^{(1)}(s)$.

Remark 1. *We considered sets B_{ij} instead D_{ij} because $(D_{ij})_s$ may be equal to 0. For example for $p^n = 343$.*

Now the main result can be represented as the following theorem.

Theorem 1. *Let u be an LRS MP over $\mathbb{Z}_{p^n}$ with generator polynomial $F(x)$, $degF(x) = m$, all the elements of $\mathbb{Z}_{p^n}$ occure in the cycle of u, u_s be the s^{th} digit-position sequence of u, where s satisfies conditions $p^n = a^{(3)}(s)2^{s+1} + 2^s - 1, s \geq 1$. Let $H < M(u)$, $|H| = 2d$, $p \geq 3$. Then the following expression holds*

$$T(u_s) \not| \frac{T(u)}{2d}.$$

Corollary 1. *If power m equals 1, then*

$$T(u_s) = \frac{T(u)}{2}.$$

4 Conclusion

We presented a method which allows to identify the existence of reduction of periods of binary digit-position sequences over prime rings. That method is based on studying digit-positions in subgroups of groups of multipliers of LRS MP. That approach was applied to show that there is no reduction of a period in one fixed digit-position sequence in $2d$ times, where $2d$ is a cardinality of subgroups of group of multipliers of the forming LRS MP u.

References

[1] Kuzmin A.S., Kurakin V.L., Nechaev A.A. Trudy po diskretnoy matematike. Pseudorandom and polylinear sequences. vol.1 Moscow:1997-VIII, pp. 139-202.

[2] Kuzmin A.S., Marshalko G.B., Nechaev A.A. Reconstruction of linear recurrent sequence over prime residue ring from its image. Mathematical Aspects of Cryptography, 2010, vol. 1, no. 2, pp. 3156.

[3] Kuzmin A.S. About periods of digit positions in r-ary notation of elements of linear recurring sequences over finite prime fields. Bezopasnost Informatsionnykh Tekhnology, no. 4 1995, pp. 71-75

[4] Kuzmin S.A. Periods of digit-position sequences received from linear recurrent sequences of maximal period over finite prime fields. Applied Discrete mathematics, 1(27) 2015, pp. 62-68.

[5] Kuzmin S.A. On binary digit-position sequences over Galois rings, admitting an effect of reduction of period. Fundamentalnaya i prikladnaya matematika. vol. 20 1 2015, pp. 223-230.

[6] Kuzmin S.A. On a sufficient condition for impossibility to reduce the period of the high order binary digit position sequences over primary rings. Applied Discrete mathematics (Supplement). 9 2016, pp. 12-14.

[7] Glukhov M.M., Elizarov V.P., Nechaev A.A. Algebra. Moscow: Gelios ARV, 2003. - vol. 2

Practical secrecy of a key under individual attack in quantum cryptography

Igor Arbekov

Abstract

In this article, we show how to ensure the practical secrecy of a key (in terms of the complexity of the truncated key search algorithms) under individual attack on quantum key distribution (QKD) protocol BB84 in quantum cryptography.

Keywords: practical secrecy of a key, truncated key search, individual attack, quantum cryptography.

1 Individual attack

We consider the oldest and best known QKD protocol BB84 [1]. The legitimate sender, Alice, randomly selects rectangular or diagonal basis and encodes logical bit, 0 or 1, into the polarization of a single photon, along the corresponding direction.The receiver, Bob, measures the polarization of the photon in one of the two bases, either rectangular or diagonal, randomly chosen by him. Only after that, Alice reveals to him the basis she used. This information is sent on a public channel that can be monitored, but not modified, by anyone else. Bob then likewise tells Alice whether he used the correct basis. If he did, Alice and Bob know one bit, that no one else ought to know [2]. Then they repeat this protocol many times.

Next, we consider the *individual attack* on quantum channel, where eavesdropper Eve uses a quantum memory and lets each signal (a single photon) interact separately with its own ancilla, and keeps the ancillas apart at later times [3]. Under this attack Eve knows the correct bases from a public channel and takes measurements after agreeing bases Alice and Bob.

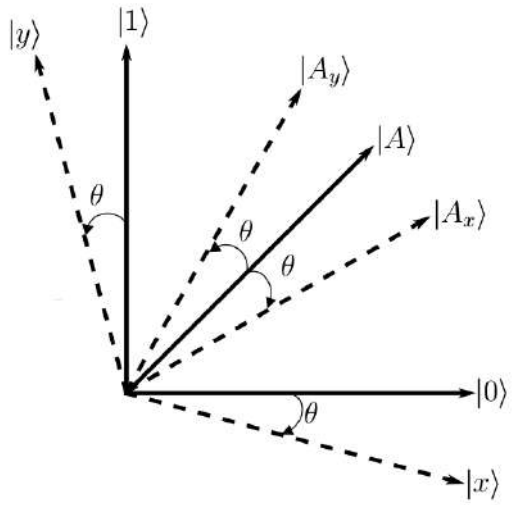

Figure 1: Mechanism of individual attack

To simplify, we present on the Fig.1 the mechanism of individual attack on a single photons in *rectangular* basis.

Here $|0\rangle, |1\rangle$, $|A\rangle$ are the quantum states in a two-dimensional Hilbert (unitary) vector space with a unit norm [4]. Vectors $|0\rangle, |1\rangle$ (horizontal or vertical photon polarizations) correspond 0 or 1 logical bits, $|A\rangle$ is the Eve's ancilla.

Eve's intervention in the quantum channel is defined by a certain angle θ, which at the same time turns quantum states $|0\rangle$ and $|A\rangle$ - clockwise to quantum states $|x\rangle$ and $|A_x\rangle$ and turns quantum states $|1\rangle$ and $|A\rangle$ - anticlockwise to quantum states $|y\rangle$ and $|A_y\rangle$. Eve knows the basis, but she does not know, $|0\rangle$ or $|1\rangle$ operates at ancilla $|A\rangle$.

Eve measures the ancilla in the form of projections on the $|0\rangle$ and $|1\rangle$ axis, and solves the problem of the statistical classification of two distributions:

$$P_{A_x} = \left(\cos^2(\pi/4 - \theta), \sin^2(\pi/4 - \theta)\right)$$

and

$$P_{A_y} = \left(\cos^2(\pi/4 + \theta), \sin^2(\pi/4 + \theta)\right).$$

As a result, Eve uses the optimal statistical classification procedure for distinguishing a priori equally probable "close" quantum states $|A_x\rangle$

and $|A_y\rangle$ with probability of error [5,6]

$$p_{AE} = \frac{1}{2} - \frac{1}{2}\sqrt{1 - (\langle A_x|A_y\rangle)^2} = \frac{1}{2} - \frac{1}{2}\sqrt{1 - \cos^2(2\theta)}. \quad (1)$$

Bob uses the same procedure for distinguishing a priori equally probable "distant" quantum states $|x\rangle$ and $|y\rangle$ with probability of error

$$p_{AB} = \frac{1}{2} - \frac{1}{2}\sqrt{1 - (\langle x|y\rangle)^2} = \frac{1}{2} - \frac{1}{2}\sqrt{1 - \sin^2(2\theta)}. \quad (2)$$

From (1,2) it is easy to obtain, that

$$p_{AE} = \frac{1}{2} - \frac{1}{2}\sqrt{p_{AB}(1 - p_{AB})}.$$

A similar result holds when the *diagonal* basis is considering .

Thus, we believe that after the transfer and measurement of quantum states, Alice and Bob have random and uniformly distributed bit strings $W^A, W^B \in \{0,1\}^L$ through a binary symmetric channel with probability of bit error p_{AB} and Eve has a bit string $W^E \in \{0,1\}^L$ through a binary symmetric channel with probability of bit error

$$p_{AE} = \frac{1}{2} - \frac{1}{2}\sqrt{p_{AB}(1 - p_{AB})}.$$

2 Privacy amplification

Alice and Bob use the some error reconciliation procedure to W^A and W^B [see, for example, 7,8] to give a common bit string $W \in \{0,1\}^S, S = L - s$, s - the number of bits to be deleted.

Then they use privacy amplification procedure by public discussion [9] to obtain the final key $\kappa = g(W) \in \{0,1\}^n$ by hash function $g : \{0,1\}^S \to \{0,1\}^n$, which is known to Eve through a public channel.

Hash function $g : \{0,1\}^S \to \{0,1\}^n$ is selected randomly from a set $\mathbf{G}$ of functions. The set $\mathbf{G} = \{g : \{0,1\}^S \to \{0,1\}^n\}$ is *universal*$_2$ (second order) [9,10]:

for all $W_1 \neq W_2 \in \{0,1\}^S$ the probability

$$\mathbf{Pr}\{g : g(W_1) = g(W_2)\} \leq 2^{-n}.$$

An example of an *universal*$_2$ class of functions is next [9, Lemma 1].

Let g be the element of $GF(2^S)$ and also interpret W as an element $GF(2^S)$. Consider the function $\{0,1\}^S \to \{0,1\}^n$ assigning to an argument W the first n bits of the element $gW \in GF(2^S)$. The class of all such functions for $g \in GF(2^S)$ is a universal$_2$ class of functions for $0 \leq n \leq S$.

What is the point of application of universal hash functions?

Suppose that after intervention in the quantum channel Eve gets the bit string $W^E \in \{0,1\}^L$. Consider the conditional (posterior) distribution

$$P\left(W|W^E\right), \quad W \in \{0,1\}^S,$$

conditional Renyi entropy [10]

$$R\left(\mathbf{W}|W^E\right) = -\log \sum_W P^2\left(W|W^E\right).$$

and *average conditional Renyi entropy*

$$\widetilde{R}\left(\mathbf{W}|\mathbf{W}^E\right) = -\log \mathbf{E}_{W^E} \sum_W P^2\left(W|W^E\right).$$

Following [11] we get

$$\frac{1}{2} \sum_{W,g,W^E} \left|P\left(g(W),g,W^E\right) - 2^{-n} P\left(g,W^E\right)\right| \leq \frac{1}{2}\sqrt{exp_2\left\{-\widetilde{R}\left(\mathbf{W}|\mathbf{W}^E\right)+n\right\}}. \tag{3}$$

We can say that the final key $\kappa = g(W)$ as a random variable is defined on the set $\{1,...,m,...,2^{-n}\}$. Then (3) can be rewritten as

$$\frac{1}{2} \sum_{m,g,W^E} \left|P\left(m,g,W^E\right) - 2^{-n} P\left(g,W^E\right)\right| \leq \frac{1}{2}\sqrt{exp_2\left\{-\widetilde{R}\left(\mathbf{W}|\mathbf{W}^E\right)+n\right\}}. \tag{4}$$

Imagine Alice's bit string as $W^A = (W, W')$ where $W' \in \{0,1\}^s$ - disclosed bits. Then

$$P\left(W|W^E\right) = \sum_{W'} P\left(W,W'|W^E\right),$$

$$\mathbf{E}_{W^E} \sum_W P^2\left(W|W^E\right) = \mathbf{E}_{W^E} \sum_W \left(\sum_{W'} P\left(W, W'|W^E\right)\right)^2 \leq$$

$$\leq \mathbf{E}_{W^E} \sum_W 2^{2s} \left(\sum_{W'} 2^{-s} P\left(W, W'|W^E\right)\right)^2 \leq$$

$$\leq 2^s \mathbf{E}_{W^E} \sum_{W^A} P^2\left(W^A|W^E\right).$$

Then it is easy to get

$$\widetilde{R}\left(\mathbf{W}|\mathbf{W}^E\right) = -\log_2 \mathbf{E}_{W^E} \sum_W P^2\left(W|W^E\right) \geq \widetilde{R}\left(\mathbf{W}^A|\mathbf{W}^E\right) - s. \quad (5)$$

From (4,5) we have

$$\frac{1}{2} \sum_{m,g,W^E} \left|P\left(m, g, W^E\right) - 2^{-n} P\left(g, W^E\right)\right| \leq \frac{1}{2}\sqrt{exp_2\{-\widetilde{R}(\mathbf{W}^A|\mathbf{W}^E) + s + n\}}. \quad (6)$$

Our assumption is that bit string W^E is connected to the bit string W^A through a binary symmetric channel with probability of bit error p_{AE}.

Let $r = W^A \oplus W^E$ be a vector of errors. Then

$$P\left(W^A|W^E\right) = \mathbf{Pr}\{r = W^A \bigoplus W^E\},$$

$$\sum_{W^A} P^2\left(W^A|W^E\right) = \left((1-p_{AE})^2 + p_{AE}^2\right)^L,$$

$$\widetilde{R}\left(\mathbf{W}^A|\mathbf{W}^E\right) = -L\log_2\left((1-p_{AE})^2 + p_{AE}^2\right).$$

Thus, using (6) we obtain the estimate

$$\frac{1}{2} \sum_{m,g,W^E} \left|P\left(m, g, W^E\right) - 2^{-n} P\left(g, W^E\right)\right| \leq$$

$$\leq \frac{1}{2}\sqrt{exp_2\{-L\log_2\left((1-p_{AE})^2 + p_{AE}^2\right) + s + n\}}.$$

The value

$$d = \frac{1}{2} \sum_{m,g,W^E} \left|P\left(m, g, W^E\right) - 2^{-n} P\left(g, W^E\right)\right| =$$

$$= \sum_{g,W^E} P\left(g, W^E\right) \left(\frac{1}{2} \sum_m \left|P\left(m|g, W^E\right) - 2^{-n}\right|\right)$$

is called the *total variational (statistical) distance* [11,12] and characterizes the proximity the conditional probability distribution of the key to the uniformly distribution on the average in the observations.

3 Practical secrecy of a key

In [13,14] we introduced the concept of the *practical secrecy* of a key as the average amount of work to determine the *encryption* key. We use *truncated* key search algorithms with a check for readability of decrypted messages. A *truncated* algorithm finds an *encryption* key with some probability of *success*.

Let an *encryption* key $\kappa \in \{1, 2, ..., N\}$ and an observation $\eta \in Z$ (Z - some space) have the joint probability distribution

$$\mathbf{Pr}(\kappa = m, \eta = z) = P(m, z).$$

. If κ is a bit string, then $N = 2^{-n}$.

When observing $\eta = z$ the posterior probability distribution of keys is calculated

$$P(1|z), ..., P(m|z), ..., P(N|z),$$

$$P(m|z) = \frac{P(m, z)}{P(z)}, \quad P(z) = \sum_{m=1}^{N} P(m, z),$$

and ordered

$$P(i_1(z)|z) \geq ... \geq P(i_m(z)|z) \geq ... \geq P(i_N(z)|z). \tag{7}$$

If for some $\alpha, \beta, ..., \gamma$ probabilities $P(i_\alpha(z)|z) = P(i_\beta(z)|z) = ... = P(i_\gamma(z)|z)$, then the posterior probabilities are ordered in accordance with the $\{1, 2, ..., N\}$.

Thus $\sigma(z) = (i_1(z), i_2(z), ..., i_N(z))$ is a some permutation of $\{1, 2, ..., N\}$ depending on the z.

Truncated algorithm U is that the M keys are being tested in the appropriate order $i_1(z), i_2(z), ..., i_M(z)$. We believe that if the M keys

have been tested, then the value of the amount of work R is equal to $M : R = M$.

Let

$$p_m^* = \mathbf{E}_z P(i_m(z)|z) = \sum_{z \in Z} P(i_m(z)|z)P(z).$$

The probability

$$\pi_U^*(M) = \sum_{m=1}^{M} p_m^*.$$

is a probability of *success* i.e., the probability of finding an *encryption* key when applying the *truncated* algorithm.

It is shown [14] that the average amount $\overline{R}_U^*(M)$ of work to determine the *encryption* key is

$$\overline{R}_U^*(M) = \frac{(1 - \pi_U^*(M))M}{\pi_U^*(M)} + \sum_{m=1}^{M} m \frac{p_m^*}{\pi_U^*(M)}$$

We define the *practical secrecy* of a key as

$$Q^* = \min_{M : \pi_U^*(M) \geq \pi_0} \overline{R}_U^*(M).$$

Let

$$d = \frac{1}{2} \sum_{m,z} |P(m, z) - \frac{1}{N} P(z)|$$

is the *total variation distance*, then the inequality [14]

$$Q^* \geq \left(1 - \frac{2d}{\pi_0}\right) \left(\frac{N(1 - 8d) + 1}{2}\right). \tag{8}$$

takes place.

When considering *truncated* algorithms it is interesting to include the point $M = 0$ in the set of keys to be tested. This is the case when keys are not tested for some observations.

Let $D \subseteq Z$ be some region of observations, $\mathbf{Pr}(\eta \in D) = P(D)$. The algorithm A_D is that we wait until an event $z \in D$ occurs. Then we arrange the keys by (7) and test before obtaining the true key, i.e. we use the *exhaustive* key search algorithm.

The average amount of work to determine the *encryption* key is

$$\overline{R}^*(D) = \sum_{z \in D} \frac{P(z)}{P(D)} \left(\sum_{m=1}^{N} mP(i_m(z)|z) \right),$$

the probability of *success* is $P(D)$, and so we define the *practical secrecy* of a key as

$$q^* = \min_{D:P(D) \geq \pi_0} \overline{R}^*(D).$$

The following chain of relations holds for $P(D) \geq \pi_0$:

$$\begin{aligned}
\overline{R}^*(D) = \sum_{z \in D} \frac{P(z)}{P(D)} \left(\sum_{m=1}^{N} m \left(P(i_m(z)|z) - \frac{1}{N} + \frac{1}{N} \right) \right) = \\
= \frac{N+1}{2} + \sum_{z \in D} \frac{P(z)}{P(D)} \left(\sum_{m=1}^{N} m \left(P(i_m(z)|z) - \frac{1}{N} \right) \right) \geq \\
\geq \frac{N+1}{2} - \frac{N}{P(D)} \sum_{z \in Z} P(z) \sum_{m=1}^{N} \left| P(i_m(z)|z) - \frac{1}{N} \right| \geq \\
\geq \frac{N+1}{2} - N\frac{2d}{\pi_0} \geq \left(1 - \frac{4d}{\pi_0} \right) \frac{N+1}{2}.
\end{aligned}$$

Hence we obtain the inequality

$$q^* = \min_{D:P(D) \geq \pi_0} \overline{R}^*(D) \geq \left(1 - \frac{4d}{\pi_0} \right) \frac{N+1}{2}.$$

4 Practical secrecy of a key under individual attack

Alice and Bob have a common bit string $W \in \{0,1\}^S, S = L - s$, s - the number of bits to be deleted, and the final key $\kappa = g(W) \in \{0,1\}^n$ by hash function $g : \{0,1\}^S \to \{0,1\}^n$.

Eve's observations are $z = \left(g, W^E \in \{0,1\}^L \right)$.

From Section 2 of this article we have

$$d = \frac{1}{2} \sum_{m,z} |P(m,z) - \frac{1}{N} P(z)| \leq \frac{1}{2} \sqrt{exp_2\{-L \log_2 ((1-p_{AE})^2 + p_{AE}^2) + s + n\}}.$$

The following error reconciliation procedure is considered in [7].

Alice and Bob:

- estimate the probability of error p_{AB},

- divide $W^A, W^B \in \{0,1\}^L$ into the same optimum size blocks $b = (p_{AB})^{-1/2}$, count the block parities, exchange the parity values over an open channel,

- remove the first bit from each block where parities are matched,

- delete all blocks where the parities did not match,

- revalue p_{AB} (to select the optimal block size), rearrange the bits and repeat the procedure.

For example, let

$n = 256$, $p_{AB} = 0.05$, $p_{AE} = \frac{1}{2} - \sqrt{p_{AB}(1-p_{AB})} = 0.282$, $L = 1500$,

then $s = L/2$, $S = 750$, $d < 10^{-15}$.

For any reasonable choice of π_0, we get that *practical secrecy* of a key

$$Q^*, q^* \approx \frac{N+1}{2}.$$

References

[1] C. H. Bennett and G. Brassard, in Proceedings of the IEEE International Conference on Computer, Systems, and Signal Processing, Bangalore, India (IEEE, New York, 1984), pp. 175-179.

[2] C. A. Fuchs, N. Gisin, R. B. Griffiths, Chi-Sheng Niu, and A. Peres, Optimal eavesdropping in quantum cryptography. I., arXiv:quant-ph/9701039v1, 1997.

[3] ETSI GS QKD 005 V1.1.1 (2010-12)

[4] M.A.Nielsen, I.L. Chung, Quantum computation and Quantum Information, Cambrige University Press, 2000, 2001.

[5] H. P. Yuen, Unconditionally secure quantum bit commitment is possible, arXiv:quant-ph/0006109v7, 2000.

[6] D.A. Kronberg, Y.I. Ozhigov, A.Y. Chernyavsky, Quantum cryptography, Lomonosov Moscow State University, CM&C, 2011.

[7] R.P. Brent, A Simple Approach to Error Reconciliation in QKD, arXiv: 1005.1206v1 [cs.DS], 2010.

[8] W. T. Buttler, S. K. Lamoreaux, J. R. Torgerson, G. H. Nickel, C. H. Donahue and C. G. Peterson, Fast, efficient error reconciliation for quantum cryptography, arXiv: quant-ph/0203096v2, 2003.

[9] C. H. Bennett, G. Brassard, C. Crepeau, U. M. Maurer, Generalized Privacy Amplification, IEEE Transaction on Information Theory, 41 (6), 1995.

[10] J. Hastad, R. Impagliazzo, L.A. Levin, M. Luby, A pseudorandom generator from any one-way function, SIAM Journal on Computing, 28(4), 1999.

[11] Y. Dodis, R. Ostrovsky, L. Reyzin, Smith A. Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data, SIAM J. on Computing, V.38(1), 2008, arXiv: cs/0602007v4 [cs.CR], 2008.

[12] C. Portmann, R. Renner, Cryptographic security of quantum key distribution, arXiv: 1409.3525v1 [quant-ph], 2014.

[13] I. M. Arbekov, Criteria of key security, Mathematical Aspects of Cryptography, V. 7, (1), 2016 (in Russian).

[14] I. M. Arbekov, Lower bound for the practical secrecy of a key, Pre-Proceeding CTCrypt, Jaroslavl, 2016.

Group Properties of Block Ciphers of the Russian Standards GOST R 34.11-2012 and GOST R 34.12-2015

Viktoriya Vlasova, Marina Pudovkina

Abstract

A group generated by the set of the round functions is often used to describe properties of a block cipher. In this paper, we use the results obtained in [11] to prove that the groups generated by the round functions of Kuznyechik and Stribog are the alternating groups. We also describe properties of the linear transformations and generalize them for the family of Stribog-like ciphers (Stribog, Anubis, etc.). We prove a theorem in which the mixing properties of linear transformations of such ciphers are considered.

Keywords: permutation groups, GOST R 34.11–2012, GOST R 34.12–2015, Kuznyechik, Stribog, alternating group, linear transformation of block cipher.

1 Introduction

In recent years, round functions of most block ciphers can be represented as a composition of functions of X-, L- and S-layer, where X-layer is a key addition layer, S-layer is an S-boxes mapping and L-layer is a linear transformation. Such block ciphers are called XSL-ciphers. This principle underlie many official standards, such as AES [1] (adopted by NIST as FIPS PUB 197), Whirlpool hash function [2] (included in ISO/IEC 10118–3:2004), the Russian standards GOST R 34.11–2012 [3] and GOST R 34.12–2015 [4].

Group properties of XSL-ciphers are widely discussed ([5], [6], [8], [9], [10], [11]). In [5], a group theoretic approach to the design and analysis of cryptographic systems has been discussed.

It was proved by Wernsdorf [7] that the permutation group G generated by the round functions of DES is also the alternating group. In his later paper [8], he used ad hoc methods to prove that the group G of AES is alternating. Some years later, Sparr and Wernsdorf [9] have given another, permutation group theoretic proof. They also have obtained a set of sufficient conditions that the group G is alternating. Another sets of such sufficient conditions have been given in [10] and [11].

The conditions given in [10] represent the slightly changed conditions of primitivity of the group G proposed earlier by the same authors [6]. Also in [10], it is shown that the AES satisfies the given conditions. It should be noted that the key schedule isn't taken into account in mentioned works.

If the group G of an XSL-cipher is alternating, several possible regularities such as the existence of non-trivial factor groups or a too small diversity of occurring permutations can be excluded for this cipher (the alternating group is a large, simple, primitive and multiple transitive permutation group).

In this paper, we use the results obtained in [11] to prove that the groups generated by round functions of Kuznyechik and Stribog are the alternating groups. We get the other, matrix representation of the linear transformations of the ciphers. We also describe properties of the linear transformations and generalize them for the family of Stribog-like ciphers (Stribog, Anubis, etc.). We prove a theorem in which the mixing properties of linear transformations of such ciphers are considered.

The paper is organized as follows. In Section 2, we provide some notions and results from the graph theory which are used in this paper. In Section 3, we give the results of [11] regarding a group generated by round functions of XSL-ciphers. In Section 4, we describe some properties of linear transformations of such ciphers regarding their mixing properties. In Section 5, we give the descriptions of Kuznyechik and Stribog and also prove that the groups generated by their round functions are the alternating groups.

2 Definitions and Notations

We use 0_m to denote the zero element of $GF(2^m)$ and 1_m to denote the multiplicative identity element of $GF(2^m)$. Let V_l be the l-dimensional vector space over the field $GF(2)$. There is a natural correspondence between elements of $GF(2)^l$ and V_l; thus, we will identify them. Let $S(V_l)$ and $A(V_l)$ denote the symmetric and the alternating group acting on V_l, respectively. Also note that elements from V_{mn} can be represented as vectors from the Cartesian product V_m^n. Therefore, we will identify this representations. Symbol $||$ means strings concatenation.

The set of all $p \times q$ matrices over $GF(u)$ is denoted by $M_{p,q}(u)$. We will write $M_p(u)$ instead of $M_{p,p}(u)$.

Elements of $GF(u)^{pq}$ are identified with matrices $\boldsymbol{\beta} \in M_{p,q}(u)$ via the mapping $\zeta\colon GF(u)^{pq} \to M_{p,q}(u)$, $\alpha \mapsto \boldsymbol{\beta}$, defined by $\beta_{ij} = \alpha_{qi+j}$ for all $i \in \{0, \ldots, p-1\}$ and $j \in \{0, \ldots, q-1\}$.

Definition. A strongly connected digraph Γ with an adjacency matrix $\mathbf{a}$ is called *primitive* if there is an integer r such that all entries of $\mathbf{a}^{\mathbf{r}}$ are non-zero.

Definition. For a linear transformation $a\colon V_m^n \to V_m^n$, we will assign a digraph $\Gamma(a)$ with a set of vertices $\{1, \ldots, n\}$ and a set of edges X, where edge $(i, j) \in X$ exists if and only if $\beta = a(\alpha)$ and β_j essentially depends on α_i for all $\alpha_1, \ldots, \alpha_n \in GF(2^m)$. We say that digraph $\Gamma(a)$ is a *graph of essential dependence* of a linear transformation a.

3 Properties of Generated Group

In papers [9], [10], [11] conditions have been provided such that the group generated by round functions of XSL-ciphers is the alternating group. In this section, we give the theorem proved in [11].

We consider a block cipher with a set of all round keys K and a round function $g_k\colon V_{mn} \to V_{mn}$, $k \in K$, which is given as

$$g_k\colon \alpha \mapsto a \circ s \circ x[k](\alpha),$$

where $a, s, x[k]\colon V_{mn} \to V_{mn}$ are functions with the following properties:

- a is an invertible linear transformation over a general linear group of a vector space V_{mn};

- s is a parallel application of n single bijective S-box mappings $s_i \colon V_m \to V_m$ defined by $s(\alpha) = \beta$ if and only if $\beta_i = s_i(\alpha_i)$ for all $i \in \{1, \ldots, n\}$;

- $x[k]$ is a XOR-addition with a round key $k \in V_{mn}$.

Let G be a group generated by the set $\{g_k \mid k \in K\}$ of all round functions. In this paper, properties of the round subkeys caused by the key schedule are neglected. The sufficient conditions to provide the following equality

$$G = \langle g_k \mid k \in K \rangle = A(V_{mn})$$

are shown below.

Let $\Gamma(a)$ be a graph of essential dependence of a linear transformation $a \colon V_m^n \to V_m^n$.

For a vector $\alpha = (\alpha_1, \ldots, \alpha_n)$, $\alpha \in V_m^n$, we will assign a set

$$I(\alpha) = \{i \in \{1, \ldots, n\} \mid \alpha_i \neq 0_m\}.$$

If I is a subset of vertices of the digraph $\Gamma(a)$ then let $J(I)$ is a set of ends of edges which starts at the set I.

For the permutations s_i, we will assign the permutations

$$s_{i,k,k'} \colon \alpha \mapsto s_i^{-1}(k' + s_i(\alpha \oplus k)),$$

where $k, k' \in V_m$ for all $i \in \{1, \ldots, n\}$. Let $H(s_i) = \langle s_{i,k,k'} \mid k, k' \in V_m^2 \rangle$ be a group generated by such permutations.

Theorem 1 [11]. *Suppose that the following conditions hold:*

1) digraph $\Gamma(a)$ is primitive;

2) for any set $L \subseteq \{1, \ldots, n\}$

$$\max_{\{\alpha \in V_{mn} \mid I(\alpha) = L\}} |I(a(\alpha))| \geq |L|,$$

with inequality strict if $|J(I)| > |I|$;

3) groups $H(s_1), \ldots, H(s_n)$ are 2-transitive, and there is a permutation $s \in S_m$ belonging to the set of the elements of these groups such that

$$|\{\alpha \in V_m \mid s(\alpha) = \alpha\}| \notin \{0, 2^0, 2^1, 2^2, \ldots, 2^m\}.$$

Then G is equal to the alternating group on V_{mn}.

4 Properties of the Linear Transformations

The first condition required by Theorem 1 regarding to the properties of a linear transformation. We describe a way to define a linear transformation of some XSL-ciphers (Stribog, Anubis, etc.). In Theorem 2, we show that the first condition is valid for such ciphers. This theorem regards to a graph of essential dependence of a linear transformation. The obtained results can be used to analyse mixing properties of a linear transformation of an XSL-cipher.

Definition. A linear transformation $a\colon GF(2^m)^{p^2} \to GF(2^m)^{p^2}$ is called $\widetilde{T}$-transformation, if a can be represented as a composition of mappings t and l ($a = l \circ t$), where:

- t transpose of a matrix from a set $M_p(2^m)$, i.e. $t(\boldsymbol{\alpha}) = \boldsymbol{\beta}$ if and only if $\beta_{ij} = \alpha_{ji}$, where $\boldsymbol{\alpha}, \boldsymbol{\beta} \in M_p(2^m)$;
- l is an invertible linear transformation over $M_p(2^m)$, that can be represented as a right multiplication by a fixed matrix $\mathbf{d} \in M_p(2^m)$, i.e. $l(\boldsymbol{\alpha}) = \boldsymbol{\alpha} \cdot \mathbf{d}$ for all $\boldsymbol{\alpha} \in M_p(2^m)$.

Theorem 2. *Let $a = l \circ t$ be a $\widetilde{T}$-transformation and the matrix $\mathbf{d}$ corresponding the transformation l does not contain zero elements. Then the digraph $\Gamma(a)$ of essential dependence of the transformation a is primitive.*

The proof is given in Appendix A.

5 Application to block ciphers of Kuznyechik and Stribog

In this section, we apply the results obtained in Section 4 to block ciphers underlying GOST R 34.11–2012 and GOST R 34.12–2015. Hereinafter the

block cipher underlying GOST R 34.11–2012, we will call the Stribog block cipher.

We begin by giving brief descriptions of Kuznyechik and Stribog.

5.1 The Description of Kuznyechik

Kuznyechik is specified by the Russian Federal standard GOST R 34.12–2015. It is an iterative block cipher with a block length of 128 bits and a key length of 256 bits. Transformations of X-, S-, and L-layer are applied for encryption throughout several iterations.

X-layer is a key addition layer defined as $x[k]\colon V_{128} \to V_{128}$,

$$x[k](\alpha) = k \oplus \alpha,$$

for all $k, \alpha \in V_{128}$.

S-layer is an S-box layer defined by the mapping $s\colon V_{128} \to V_{128}$, $s(\alpha) = s'(\alpha_{15})|| \ldots ||s'(\alpha_0)$, where $\alpha \in V_{128}$, $\alpha_i \in V_8$ for all $i \in \{0, \ldots, 15\}$; the permutations $s' \in S(V_{256})$ are defined via array $s' = (s'(0), s'(1), \ldots, s'(255))$.

The L-layer transformation $a_1\colon V_{128} \to V_{128}$ is based on a linear-feedback shift register with a given feedback function $l'\colon V_8^{16} \to V_8$. The function l' is given by polynomial of degree 16 over $GF(2^8)$ with irreducible polynomial $p_k(x) = x^8 + x^7 + x^6 + x + 1$.

Depending on the values of round keys $k_1, \ldots, k_{10}$, the encryption algorithm is a substitution $e_{k_1,\ldots,k_{10}}\colon V_{128} \to V_{128}$ defined as follows:

$$e_{k_1,\ldots,k_{10}}(\alpha) = x[k_{10}] \circ \prod_{i=1}^{9} (a_1 \circ s \circ x[k_i](\alpha)),$$

where $\alpha, k_i \in V_{128}$ for all $i \in \{1, \ldots, 10\}$.

Let $g_k\colon V_{128} \to V_{128}$ be a round function of Kuznyechik

$$g_k\colon \alpha \mapsto a_1 \circ s \circ x[k](\alpha),$$

where $\alpha, k \in V_{128}$.

Further, we will consider the group $G = \langle g_k \mid k \in K \rangle$ generated by the set of all round functions of Kuznyechik. Note that properties of the key scheduling algorithm are not taken into account.

5.2 The Description of Stribog

Stribog is specified by the Russian Federal standard GOST R 34.11–2012. The hash procedure is based on block encryption with block size of 512 bits and provides digest sizes of 256 and 512 bits. The transformations of X-, S-, P- and L-layer are used to calculate the hash code of a message from V^*.

X-layer is a round constants addition layer defined as $x[k]\colon V_{512} \to V_{512}$,

$$x[k](\alpha) = k \oplus \alpha,$$

for all $k, \alpha \in V_{512}$.

S-layer is an S-box layer defined by the mapping $s\colon V_{512} \to V_{512}$, $s(\alpha) = s'(\alpha_{63})|| \dots ||s'(\alpha_0)$, where $\alpha \in V_{512}$, $\alpha_i \in V_8$ for all $i \in \{0, \dots, 63\}$; the permutations $s' \in S(V_{256})$ are defined via array $s' = (s'(0), s'(1), \dots, s'(255))$.

The P-layer transformation $V_{512} \to V_{512}$ defined by the mapping $t\colon \alpha \mapsto \beta$, $\beta = (\beta_{63}|| \dots ||\beta_0) = (\alpha_{\tau(63)}|| \dots ||\alpha_{\tau(0)})$, where $\tau \in S(\{0, 1, \dots, 63\})$.

The L-layer transformation $V_{512} \to V_{512}$ defined by the mapping $l\colon \alpha \mapsto \beta$, $\beta = (\beta_7|| \dots ||\beta_0) = l''(\alpha_7)|| \dots ||l''(\alpha_0)$, where $l''\colon V_{64} \to V_{64}$ is a right multiplication by a matrix $\mathbf{d_s} \in M_{64}(2)$.

The hash code value of a message from V^* is calculated via iterated procedure. The compression function $h_\delta\colon V_{512} \times V_{512} \to V_{512}, \delta \in V_{512}$, acts on each iteration. The compression function defined by rule

$$h_\delta(\eta, \alpha) = f(l \circ t \circ s(\eta \oplus \delta), \alpha) \oplus \eta \oplus \alpha, \eta, \alpha \in GF(2)^{512},$$

where

$$f(k, \alpha) = x[k_{13}] \circ \prod_{i=1}^{12} (l \circ t \circ s \circ x[k_i](\alpha)),$$

for all $\alpha, k_i \in V_{512}$ for all $i \in \{1, \dots, 13\}$.

Let linear transformation $a_2\colon V_{512} \to V_{512}$ be a composition of mappings acting on L- and P-layer ($a_2 = l \circ t$). Let $g_k\colon V_{512} \to V_{512}$ be a round function of Stribog

$$g_k\colon \alpha \mapsto a_2 \circ s \circ x[k](\alpha),$$

where $\alpha, k \in V_{512}$.

Further, we will consider the group $G = \langle g_k \mid k \in K \rangle$ generated by the set of all round functions of Stribog. Note that properties of the key scheduling algorithm are not taken into account.

5.3 Group properties of the Kuznyechik block cipher and the Stribog block cipher

First, we give matrix representations of linear transformations of Kuznyechik and Stribog.

Linear transformation of Kuznyechik can be represented as right multiplying by matrix from $M_{16}(2^8)$. Such matrix $\mathbf{m_k} \in M_{16}(2^8)$ has been calculated (see Appendix B).

In GOST R 34.11–2012, the description of L-layer is based on the transformation l, which is given by the right multiplication by a fixed matrix from $M_{64}(2)$.

Binary vectors $\alpha \in V_{512}$ input to the linear transformation consider as elements of $GF(2^8)^{64}$. We will identify such vector with a matrix $\boldsymbol{\beta} \in M_8(2^8)$ via mapping $GF(2^8)^{64} \to M_8(2^8)$, $\alpha \mapsto \boldsymbol{\beta}$, defined by the rule $\beta_{ij} = \alpha_{8i+j}$ for all $i, j \in \{0, \ldots, 7\}$. As was shown in [12], the L-layer transformation of Stribog can be represented as a left multiplication by the matrix from $M_8(2^8)$. Such matrix has been found in [12]. The multiplication is performed in $GF(2^8)$ with irreducible polynomial $p'(x) = x^8 + x^6 + x^5 + x^4 + 1$. But calculations using that matrix require the additional conversions of the state bits.

The L-layer transformation of the Stribog block cipher can also be represented as a right multiplication by the matrix $\mathbf{d_s} \in M_8(2^8)$, i.e. $\alpha \mapsto \beta$, $\beta = \alpha \cdot \mathbf{d_s}$ for all $\alpha, \beta \in M_8(2^8)$. Such matrix $\mathbf{d_s} \in M_8(2^8)$ has been found using the algorithm described in [12] and given in Appendix B. The multiplication is performed in $GF(2^8)$ with irreducible polynomial

$$p_s(x) = x^8 + x^4 + x^3 + x^2 + 1.$$

The P-layer transformation of Stribog can be represented as a transposition of a matrix from $M_8(2^8)$.

Thus the linear transformation a_2 of Stribog is a $\widetilde{T}$-transformation.

Theorem 3. *For the Kuznyechik and Stribog block ciphers, the groups G generated by the sets of all round functions are equal to the alternating groups $A(V_{128})$ and $A(V_{512})$, respectively.*

Proof. First, we check the first condition of Theorem 1 for the Kuznyechik block cipher by describing properties of its linear transformation a_1.

Using the matrix $\mathbf{m_k} \in M_{16}(2^8)$ of the linear transformation a_1, it is easy to find the graph of essential dependence $\Gamma(a_1)$. Obviously, this graph is primitive if the matrix $\mathbf{m_k}$ does not contain any zero elements.

Secondly, we consider properties of the linear transformation of the Stribog block cipher.

The linear transformation a_2 of Stribog is a $\widetilde{T}$-transformation and its matrix $\mathbf{d_s}$ does not contain zero elements. Therefore, the graph $\Gamma(a_2)$ of essential dependence is primitive according to Theorem 2.

To check the second condition of Theorem 1, we have used Theorem 2 proved in [11]. According to this theorem, condition 2 is correct if inequality

$$2^{mn} < (2^{m-1})^{n-1}(2^m + 2^{m-1} - 2) \tag{1}$$

is true. Through direct calculations, inequality (1) correctness has been verified for the Kuznyechik and Stribog block ciphers. For Kuznyechik, the left-hand side of inequality (1) is $3{,}4028 \times 10^{38}$ and the right-hand side is $4{,}7881 \times 10^{38}$. For Stribog, the left-hand side of inequality (1) is $1{,}3408 \times 10^{154}$ and the right-hand side is $1{,}5635 \times 10^{154}$.

The third condition of Theorem 1 characterizes the properties of S-boxes of a block cipher. S-boxes permutations are the same for Kuznyechik and Stribog. It should be noted that the same permutations are used in each S-box. The difference distribution matrix $\boldsymbol{\lambda}$ for S-boxes has been found. Then the matrix $\boldsymbol{\mu}$ has been calculated by rule $\boldsymbol{\mu} = \boldsymbol{\lambda} \cdot \boldsymbol{\lambda}^T$, where $\boldsymbol{\lambda}^T$ denotes the transposed matrix $\boldsymbol{\lambda}$. For the obtained matrix, we will assign the graph $\Lambda(s)$ with a set of vertices being the set of all non-zero vectors from V_m, its vertices α and β are connected by an edge if and only if $\mu_{\alpha\beta} > 0$.

According to Theorem 3 proved in [11], a group $H(s)$ is 2-transitive if and only if the graph $\Lambda(s)$ is connected. Connectivity of the graph Λ for S-boxes of Kuznyechik and Stribog has been verified by direct calculations.

Consider the second part of the third condition of Theorem 1. Note that proving it is equal to existence of elements υ of difference distribution matrix $\boldsymbol{\lambda}$ such that $\upsilon \notin \{0, 2^0, 2^1, 2^2, \ldots, 2^m\}$. Such elements which are equal to 6, have been found in the calculated matrix $\boldsymbol{\lambda}$.

□

References

[1] "Advanced Encryption Standard (AES)", *FIPS Publication 197*, National Institute of Standards and Technology (US), November 2001, http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf.

[2] *ISO/IEC 10118-3: Information Technology - Security Techniques - Hash-functions - Part 3: Dedicated hash-functions*, International Organization for Standardization, 2004.

[3] *GOST R 34.11–2012. Information technology. Cryptographic data security. Hash-function*, Standardinform, Moscow, 2012, 38 c.

[4] *GOST R 34.12–2015. Information technology. Cryptographic data security. Block cipher*, Standardinform, Moscow, 2015, 25 c.

[5] Glukhov M.M. and Pogorelov B.A., "On some applications of groups in cryptography", *Mathematics and informational security. Proceedings of MSU conference October, 28—29, 2004*, MCNMO, Moscow, 2005, 19–31

[6] Caranti A., Dalla Volta F., Sala M., Villani F., "Imprimitive permutation groups generated by the round functions of key-alternating block ciphers and truncated differential cryptanalysis", 2006, arXiv: math/0606022v2.

[7] Wernsdorf R., "The One-Round Functions of the DES Generate the Alternating Group", *Advances in Cryptology - EUROCRYPT '92, Workshop on the Theory and Application of of Cryptographic Techniques,*

Balatonfüred, Hungary, May 24-28, 1992, Proceedings,, Lecture Notes in Computer Science, **658**, Springer, 1992, 99–112.

[8] Wernsdorf R., "The Round Functions of RIJNDAEL Generate the Alternating Group", *Fast Software Encryption: 9th International Workshop, FSE 2002 Leuven, Belgium, February 4–6, 2002 Revised Papers*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2002, 143–148.

[9] Sparr R., Wernsdorf R., "Group Theoretic Properties of Rijndael-like Ciphers", *Discrete Appl. Math.*, **126**, 3139–3149.

[10] Caranti A., Dalla Volta F., Sala M., "An application of the O'Nan-Scott theorem to the group generated by the round functions of an AES-like cipher", *Designs, Codes and Cryptography*, **52**:3 (2009), 293–301.

[11] Maslov A.S., "On sufficient conditions to generate the alternating group by SA-permutations", *Tr. Inst. Mat.*, **15**:2 (2007), 58–68, http://mi.mathnet.ru/timb98.

[12] Kazymyrov O., Kazymyrova V., "Algebraic Aspects of the Russian Hash Standard GOST R 34.11–2012", *Cryptology ePrint Archive, Report 2013/556*, 2013, http://eprint.iacr.org/2013/556.

Appendix

A The Proof of Theorem 2

Theorem 2. *Let $a = l \circ t$ be a $\widetilde{T}$-transformation and the matrix $\mathbf{d}$ corresponding the transformation l does not contain zero elements. Then the digraph $\Gamma(a)$ of essential dependence of the transformation a is primitive.*

Proof of Theorem 2. Let $n = p^2$.

The $\widetilde{T}$-transformation a can be represented as a right multiplication by the matrix from $M_n(2^m)$. We will find the matrix $\mathbf{m} \in M_n(2^m)$ such that $a(\alpha) = \alpha \cdot \mathbf{m}$ for all $\alpha \in GF(2^m)^n$.

A matrix transposition t can be represented as a right multiplication by a matrix from $M_n(2)$. This matrix is a square block matrix $\mathbf{t} = (t_{ij})$,

where $i, j \in \{0, \ldots, p-1\}$. We will identify the matrix $\mathbf{t} \in M_n(2)$ with a matrix $\widehat{\mathbf{t}} \in M_n(2^m)$ via mapping defined by rule

$$\widehat{t}_{ij} = \begin{cases} 1_m, & \text{if } t_{ij} = 1, \\ 0_m, & \text{if } t_{ij} = 0 \end{cases}$$

for all $i, j \in \{0, \ldots, n-1\}$. Then the mapping $t\colon M_p(2^m) \to M_p(2^m)$ will be identified with the mapping $GF(2^m)^{p^2} \to GF(2^m)^{p^2}$ defined by right multiplication by the matrix $\widehat{\mathbf{t}} \in M_n(2^m)$.

The transformation $l\colon M_p(2^m) \to M_p(2^m)$ corresponds to the mapping $GF(2^m)^{p^2} \to GF(2^m)^{p^2}$ that can be represented as the right multiplication by a matrix $\widehat{\mathbf{d}} \in M_n(2^m)$. Note that the matrix $\widehat{\mathbf{d}}$ is a quasidiagonal $(n \times n)$-matrix with $p \times p$ blocks. For each $i \in \{0, \ldots, p-1\}$ block $\widehat{\mathbf{d}}_{\mathbf{ii}} \in M_p(2^m)$ is equal to the matrix $\mathbf{d} \in M_p(2^m)$. If the matrix $\mathbf{d}$ does not have zero elements than each block $\widehat{\mathbf{d}}_{\mathbf{ii}}$ does not contain zero elements for all $i \in \{0, \ldots, p-1\}$.

A matrix $\widehat{\mathbf{m}} = \widehat{\mathbf{t}} \cdot \widehat{\mathbf{d}}$ (mapping t is carried out at first) is a matrix of the transformation a which is defined by $a(\alpha) = \alpha \cdot \widehat{\mathbf{m}}$ for all $\alpha \in GF(2^m)^n$.

Consider an arbitrary block $\widehat{\mathbf{t}}_{\alpha\beta}$ of the block matrix $\widehat{\mathbf{t}}$. Elements of this block have indexes $(\alpha n+u, \beta n+v)$, where $u, v \in \{0, \ldots, n-1\}$. Obviously, all elements with indexes $(in+j, jn+i)$ such that $i, j \in \{0, \ldots, n-1\}$ are in different blocks. Therefore, in each block $\widehat{\mathbf{t}}_{\alpha\beta}$ there is only one non-zero element which is $\widehat{t}_{\beta\alpha}$.

If we multiply the block matrix $\widehat{\mathbf{t}}$ by the quasidiagonal matrix $\widehat{\mathbf{d}}$ than j^{th} row of each block $\widehat{\mathbf{m}}_{\mathbf{ij}}$ will contain only non-zero elements (because j^{th} row of the block $\widehat{\mathbf{t}}_{\mathbf{ij}}$ contains 1_m), where $i, j \in \{0, \ldots, p-1\}$.

Let $\Gamma(a)$ be a graph of essential dependence of a. Its adjacency matrix $\widehat{\mathbf{a}}$ can be obtained from the matrix $\widehat{\mathbf{m}}$ by rule

$$\widehat{a}_{ij} = \begin{cases} 0, & \text{if } \widehat{m}_{ij} = 0_m, \\ 1, & \text{if } \widehat{m}_{ij} \neq 0_m \end{cases}$$

for all $i, j \in \{0, \ldots, n-1\}$. Note that j^{th} row of each block $\widehat{\mathbf{a}}_{\mathbf{ij}}$ contains only non-zero elements. Therefore, all elements of the matrix $\widehat{\mathbf{a}}^{\mathbf{2}}$ are non-zero, i.e. graph $\Gamma(a)$ is primitive.

□

B The Matrices of the Linear Transformations

The Matrix $\mathbf{m_k}$ of the transformation a_1 of Kuznyechik is

$$\begin{pmatrix}
CF & 6E & A2 & 76 & 72 & 6C & 48 & 7A & B8 & 5D & 27 & BD & 10 & DD & 84 & 94 \\
98 & 20 & C8 & 33 & F2 & 76 & D5 & E6 & 49 & D4 & 9F & 95 & E9 & 99 & 2D & 20 \\
74 & C6 & 87 & 10 & 6B & EC & 62 & 4E & 87 & B8 & BE & 5E & D0 & 75 & 74 & 85 \\
BF & DA & 70 & 0C & CA & 0C & 17 & 1A & 14 & 2F & 68 & 30 & D9 & CA & 96 & 10 \\
93 & 90 & 68 & 1C & 20 & C5 & 06 & BB & CB & 8D & 1A & E9 & F3 & 97 & 5D & C2 \\
8E & 48 & 43 & 11 & EB & BC & 2D & 2E & 8D & 12 & 7C & 60 & 94 & 44 & 77 & C0 \\
F2 & 89 & 1C & D6 & 02 & AF & C4 & F1 & AB & EE & AD & BF & 3D & 5A & 6F & 01 \\
F3 & 9C & 2B & 6A & A4 & 6E & E7 & BE & 49 & F6 & C9 & 10 & AF & E0 & DE & FB \\
0A & C1 & A1 & A6 & 8D & A3 & D5 & D4 & 09 & 08 & 84 & EF & 7B & 30 & 54 & 01 \\
BF & 64 & 63 & D7 & D4 & E1 & EB & AF & 6C & 54 & 2F & 39 & FF & A6 & B4 & C0 \\
F6 & B8 & 30 & F6 & C4 & 90 & 99 & 37 & 2A & 0F & EB & EC & 64 & 31 & 8D & C2 \\
A9 & 2D & 6B & 49 & 01 & 58 & 78 & B1 & 01 & F3 & FE & 91 & 91 & D3 & D1 & 10 \\
EA & 86 & 9F & 07 & 65 & 0E & 52 & D4 & 60 & 98 & C6 & 7F & 52 & DF & 44 & 85 \\
8E & 44 & 30 & 14 & DD & 02 & F5 & 2A & 8E & C8 & 48 & 48 & F8 & 48 & 3C & 20 \\
4D & D0 & E3 & E8 & 4C & C3 & 16 & 6E & 4B & 7F & A2 & 89 & 0D & 64 & A5 & 94 \\
6E & A2 & 76 & 72 & 6C & 48 & 7A & B8 & 5D & 27 & BD & 10 & DD & 84 & 94 & 01
\end{pmatrix}.$$

The multiplication is performed in $GF(2^8)$ with irreducible polynomial $p_k(x) = x^8 + x^7 + x^6 + x + 1$.

The Matrix $\mathbf{d_s}$ of the transformation l of Stribog is

$$\begin{pmatrix}
83 & 47 & 8B & 07 & B2 & 46 & 87 & 64 \\
46 & B6 & 0F & 01 & 1A & 83 & 98 & 8E \\
AC & CC & 9C & A9 & 32 & 8A & 89 & 50 \\
03 & 21 & 65 & 8C & BA & 93 & C1 & 38 \\
5B & 06 & 8C & 65 & 18 & 10 & A8 & 9E \\
F9 & 7D & 86 & D9 & 8A & 32 & 77 & 28 \\
A4 & 8B & 47 & 4F & 9E & F5 & DC & 18 \\
64 & 1C & 31 & 4B & 2B & 8E & E0 & 83
\end{pmatrix}.$$

The multiplication is performed in $GF(2^8)$ with irreducible polynomial $p_s(x) = x^8 + x^4 + x^3 + x^2 + 1$.

On Software Implementation of Kuznyechik on Intel CPUs

Andrey Rybkin

Abstract

In this paper we investigate performance issues of the Kuznyechik block cipher to get high speed in software on Intel CPUs. We consider general block ciphers implementation methods, including byte slicing technique, available speed-up possibilities on Intel architecture, and evaluate the efficiency of them when applied to Kuznyechik. Practical implementation results are given, and potential speed-ups are discussed.

Keywords: block cipher, Kuznyechik, fast software implementation, byte slicing, high speed, performance.

1 Introduction

Kuznyechik is one of the block ciphers specified in Russian national standard GOST R 34.12-2015 [1]. There are a number of papers dealing with its performance issues [2], [3], [4], [5]. In this paper, we focus on high-speed software implementation of Kuznyechik on Intel CPUs. We consider several techniques of implementing block ciphers such as look-up tables and slicing, and describe the basic ways of using data and instruction parallelism on modern Intel CPUs. Applying these techniques, we produced several implementations of Kuznyechik on CPUs of various generations. We present the performance results and discuss possible improvements and perspectives.

2 Block ciphers in software

There are several general ways of implementing block ciphers in software.

One of the most efficient methods is the usage of precomputed look-up tables (LUTs for short). LUTs can significantly simplify any operation by replacing it by easy table look-ups (see, for example, [6]). In block ciphers this method is often used to implement linear transformation together with non-linear transformation when the latter precedes the former. Actually any operation can be implemented by using LUT. However, it is necessary to take into account the resulting tables size and the restrictions of the computer memory hierarchy such as the size and the performance of each memory level. Because the LUTs size may have a strong effect on the overall performance it is important to observe the particular cipher properties when choosing which operations to implements via LUTs.

Another popular method is slicing (see, for example, [7]). It is usually based on bitwise (bit slicing) or bytewise (byte slicing) manipulation with input values for calculating the output ones. Slicing techniques do not require a lot of precomputed data as LUTs do. In some cases it is possible to provide constant working time for slicing implementation [7]. Then this can safeguard the cipher against cache-timing attacks [8]. The major precondition to apply a slicing method is a bit- or byte-oriented cipher structure. Another important prerequisite is a possibility to process several blocks simultaneously. In case of both, slicing will be very effective.

Slicing and LUTs, of course, do not exclude one another. Slicing implementations may use small look-up tables.

3 Look-up tables for Kuznyechik

3.1 Linear transformation

The linear transformation L in Kuznyechik can be represented as multiplication of a 128-bit vector by a fixed 128×128 binary matrix. There is a well-known folkloristic implementation method for such multiplication using LUTs. It generalizes the approach presented in [6], and involves a parameter s which specifies the number of matrix rows defining one table. If s is a divisor of 128, then the total size of the LUTs is $\frac{128}{s} \cdot 2^s \cdot 128$ bits.

A significant feature of Kuznyechik is that L is defined recursively:

$L = R^{16}$. Therefore for L to be implemented it suffices to implement any of the transformations R^i, $i \in \{1, 2, 4, 8, 16\}$. In turn, transformation R^i can be represented as multiplication of a 128-bit vector by a fixed $128 \times n$ binary matrix, where $n = 8 \cdot i$. The remaining $(128 - n)$ bits of the output vector are calculated from the input vector by easy shift.

Thus the size of LUTs to implement L depends on the choice of the particular transformation R, R^2, R^4, R^8, or R^{16}, and the value of parameter s. The first one defines the length of the table elements, while the second fixes the number of the tables and the number of elements in one table. Altogether, the total size of LUTs to implement L via R^i with parameter s is $\frac{128}{s} \cdot 2^s \cdot n = \frac{128}{s} \cdot 2^s \cdot 8 \cdot i$ bits.

3.2 Non-linear transformation

The non-linear transformation S in Kuznyechik is defined via the bytewise non-linear substitution $\pi : V_8 \to V_8$. Hence the size of the trivial LUT for implementing S is $2^8 \cdot 8 = 2048$ bits. In [9] an alternative representation of the substitution π is given in which only non-linear substitutions of the form $V_4 \to V_4$ were used. This representation significantly increases S flexibility with respect to implementation.

The non-linear transformation S can also be implemented jointly with the linear transformation R^i for any $i \in \{1, 2, 4, 8, 16\}$ if the latter is implemented via LUTs with $s = 8k$ for some $k = 1, 2, \ldots, 16$. In this case, the LUTs for the composition R^iS can be easily obtained from the LUTs for R^i by a simple reordering of tables' elements. It should be noted that despite the easy transition between respective tables they are different, and each of them is to be stored if we plan to use both: R^i and R^iS.

4 Intel CPUs' features

The target computational platform abilities are also very important concerning implementation. For example such characteristics as registers length, instruction latency and throughput, possibility of parallel data processing have a direct impact on performance and variability of implementation.

4.1 Long registers, special instructions and parallel processing

One of the data-level parallelism methods on modern Intel CPUs is a SIMD technology. This technology is based on the usage of long registers for Multiple Data and special instruction sets to perform computations using a Single Instruction. There are several SIMD instruction sets on Intel CPUs suitable for different length registers.

Let us consider what benefits can be gained using SIMD when the linear transformation L is implemented. For this purpose we consider two L implementations: via R^i, $i \in \{1, 2, 4, 8\}$, and via R^{2i}. We have seen that when L is implemented via R^i the size of LUTs twice as small as in the implementation of L via R^{2i} with the same parameter s. So from this point of view the implementation via R^i looks more preferable.

Let the maximal length of registers be $m = 2^t$ bits. For implementing L via LUTs the LOAD and XOR operations are only needed. Assume that both operations are defined only on values of length less or equal to m bit, and the execution time of a LOAD (XOR) operation is the same for all these values. If values longer than m-bits must be processed by the LOAD (XOR) operation, then these values are cut on parts of length less or equal to m and processed part by part.

At first assume that each register is used to store only one table value or part of this value.

If $8 \cdot 2i \leq m$, then in both implementations any table value fits in an m-bit register. Therefore the implementations of R^i and R^{2i} require the same number of LOAD (XOR) operations. Since $R^{2i} = R^i R^i$, then the implementation via R^i involves twice as many LOAD (XOR) operations as the implementation via R^{2i}. Note that the implementation via R^i uses at most half of an m-bit register.

If $8 \cdot 2i > m$, then any table value in the implementation via R^{2i} requires twice as many registers as any table value in the implementation via R^i. Therefore the implementation of R^i requires twice as less the number of LOAD (XOR) operations as the implementation of R^{2i}. Hence the implementations via R^i and via R^{2i} involve the same number of LOAD (XOR) operations.

Assume now that each register can be used to store several table values. In the case of independent blocks processing it allows us to group similar parts from different blocks to fill in long registers before XOR operation is performed. It does not change the number of XOR operations needed to implement R^i, but it allows processing several blocks in parallel. It follows that for any fixed m the total number of XOR operations per one block remains the same regardless of the value of i. However the number of LOAD operations is changing in the same way as in the previous case.

In order to reduce the number of LOAD operations, it is necessary to load data from a LUT in a parallel way. There are several Intel instructions for this purpose. The first of them is *pshufb* (or *vpshufb*). This instruction is designed to implement $V_4 \to V_8$ LUTs. And it allows performing 16, 32, or 64 simultaneous LOAD operations from tables depending on register length. The other instruction is *vpgather*. It is designed for LUTs implementation up to $V_{32} \to V_{32}$, $V_{32} \to V_{64}$, $V_{64} \to V_{32}$, $V_{64} \to V_{64}$. In this case the number of simultaneous LOAD operation is equal to 2, 4, 8 or 16 depending on register length and particular *vpgather* instruction form.

Based on the this instructions structures we may conclude that *pshufb* is useful to implement R with parameter $s = 4$, and *vpgather* is useful to implement R^4 or R^8 with parameter s up to 64 (it should be taken into account that large s can lead to extremely large LUTs).

4.2 Execution units and ports

The execution core of most modern Intel CPU consists of various execution units. To provide instruction-level parallelism within a CPU core one can use several execution units simultaneously. In general, precise instruction distribution among execution ports is a very hard job. However, there are a few simple ways to do this rather effectively. For example if several instructions operate on independent data parts, then one can try to parallelize its execution by its reordering and renaming variables. If parallel processing of blocks is possible, then several blocks may be considered as an independent data parts. Smart interleaving of instructions processing different blocks can lead to a significant speed-up.

5 Implementations

5.1 Description

We practically applied the above considerations and obtained a variety of Kuznyechik implementations. Let us consider some of them.

Table 1 contains a description of the implementations. Each one uses LUTs one way or the other. Transformations implemented via LUTs and the values of s are given in the 3rd and 4th columns. The parallelism degree is the number of blocks processed simultaneously. The data-level parallelism column gives the number of blocks which parts are simultaneously loaded in one long register (see section 4.1). The instruction-level parallelism column represents the number of groups of blocks where each group is processed independently for a more efficient usage of execution units (see section 4.2). Thus, the total number of blocks processed simultaneously by a certain implementation is the product of the values from Data and Instruction columns. For simplicity we append this number to the end of the implementation name.

Table 1: Description of the implementations

Name	SIMD	Transformations by LUTs	s	LUTs size (KByte)	Parallelism degree	
					Data	Instr
GPR-LS-1	–	LS	8	64	1	1
GPR-R8S-S-1	–	R^8S, S^{-1}	8	$32 + 0.25$	1	1
GPR-R8S-S-2	–	R^8S, S^{-1}	8	$32 + 0.25$	1	2
GPR-R4-R4S-1	–	R^4, R^4S	8	$16 + 16$	1	1
SSE-LS-1	SSE2	LS	8	64	1	1
SSE-LS-4	SSE2	LS	8	64	1	4
SSE-RS8-S-1	SSE2	R^8S, S^{-1}	8	$32 + 0.25$	1	1
AVX-LS-2	AVX2	LS	8	64	2	1
AVX-LS-8	AVX2	LS	8	64	2	4
AVX-R4-R4S-8	AVX2	R^4, R^4S	8	$16 + 16$	8	1
AVX-R-S-32	AVX2	R, S	4	$0.5 + 0.25$	32	1

AVX-R4-R4S-8 and AVX-R-S-32 use *vpgather* and *vpshufb* instructions respectively for loading data from LUTs. AVX-R-S-32 involves only byte-

wise operations, so we regard it as a byte slicing implementation. It performs all look-up operations in constant time independently of input values. Therefore AVX-R-S-32 could be cache-timing attack resistant.

5.2 Speed results

For computations we used Intel CPUs of various generations: Core i7-2600 (3.40 GHz), Core i7-4770 (3.40 GHz), Core i7-6700 (3.40 GHz), Xeon E5-1650 v4 (3.60 GHz), Xeon E5-2650 v4 (2.20 GHz). All these CPUs have the same L1 and L2 cache size per core: L1i cache size is 32 KB, L1d cache size is 32 KB, and L2 cache is 256 KB. All computations were made in one thread on one core.

The source code was written in C language, and compiled using the following compilers: Visual C++ 2015 (Windows), Intel C++ 17.0 (Windows), gcc 4.7.2 (Debian GNU/Linux).

For testing purposes we fixed the cipher key, but generated it randomly for every start of the code. For measurements we did not take into account the key schedule of Kuznyechik. So the round keys were computed before measurements. The number of input blocks for an implementation was as specified in the end of its name. For measurements all input blocks were re-encrypted in ECB encryption mode [10] up to 10^8 times.

Besides key change and key schedule issues there are three main restrictive features of such measurement method. The first one is a parallel processing of several blocks. This feature is inherent to ECB, CTR (both encryption/decryption), CFB, CBC (both decryption) modes of operations and to OFB, CFB, CBC (all encryption/decryption) extended modes of operations [10]. The second one is a usage of only encryption algorithm. This feature is inherent to CTR, OFB, CFB and OMAC1 [10]. The third one is a re-encryption of short data parts. This feature can lead to some decreasing of speed in the real systems because of the additional time required to transfer data from and to memory.

It should be noted that in the case of non-parallelizable modes parallel processing of several blocks can be done by parallel processing of several messages. Restrictions related to usage of only encryption algorithm can be

easily lifted by addition of analogous implementations based on decryption algorithm. Such implementations (except AVX-R-S-32) will presumably be on 10-20% slower because of the decryption algorithm larger complexity.

The speed we obtained in MBytes per second is given in Table 2.

Table 2: Performance (MBytes/s)

CPU→	Core i7-2600			Core i7-4770			Core i7-6700	Xeon E5-1650	Xeon E5-2650
Name ↓	VC	Intel	gcc	VC	Intel	gcc	VC	gcc	gcc
GPR-LS-1	110	125	100	135	145	125	145	120	75
GPR-R8S-S-1	130	125	120	160	160	140	170	140	85
GPR-R8S-S-2	160	155	125	210	210	155	235	150	95
GPR-R4-R4S-1	65	70	55	95	95	65	105	65	40
SSE-LS-1	145	150	140	160	165	150	170	150	90
SSE-LS-4	230	225	210	300	290	270	360	260	160
SSE-RS8-S-1	95	100	105	120	125	115	125	110	70
AVX-LS-2	–	–	–	200	170	195	210	185	110
AVX-LS-8	–	–	–	255	270	275	290	265	160
AVX-R4-R4S-8	–	–	–	65	65	45	115	65	40
AVX-R-S-32	–	–	–	235	250	220	255	205	130

The analysis of Table 2 gives us the following observations.

• The transition from LS tables to R^8S and S^{-1} tables having only general purpose registers ("GPR-...") turns out to be very effective. The number of operations remains about the same but the total size of LUTs is reduced twice, and it seems that almost all LUTs fit in L1d cache.

• The transition from R^8S and S^{-1} tables to R^4 and R^4S tables with only general purpose registers dramatically decreases the speed because the number of load operations increases too much.

• The transition from LS tables to R^8S and S^{-1} tables in the case of SSE registers ("SSE-...") is not effective either by the same reason.

• The *vpgather* instruction (AVX-R4-R4S-8) is not that good for high-speed implementations on up to the 6th generation Intel CPUs due to its large latency and reciprocal throughput.

• A significant speed-up is achieved if processing of independent blocks is possible. The best our result is 360 MB/s or 10 cycles/byte in one thread

on one CPU core (SSE-LS-4). In case of using non-parallelizable modes of operation the attained speed in one thread on one core is 170 MB/s, or about 21 cycles/byte (GPR-R8S-S-1).

• Byte slicing technique and *vpshufb* instructions (AVX-R-S-32) with 256-bit registers gave the speed of about 255 MB/s. It is not the highest among the others. However, it is mainly defined by the S transformation. We tried a simplified Kuznyechik version without S, and attained the speed of more than 500 MB/s which is higher than in any other implementation of this simplified cipher. This completely downgrades the complains of some practitioners that the L transformation of Kuznyechik is too complicated for implementation. So a potential speed-up of the S transformation can make byte slicing implementations to be the fastest. Moreover, using longer 512-bit registers will presumably double the speed of byte slicing implementation. It is very unlikely that any other software implementation on Intel CPUs will give such an increase in speed.

When running on a single core Intel CPUs typically enable Turbo Boost technology, increasing CPU speed by a 10-15%. To take this effect into account, we also performed the measurements on multiple cores simultaneously. The resulting speed of the SSE-LS-4 implementation on four physical cores of Intel Core i7-6700 was 1340 MB/s.

In Table 3 we compare performance of our implementations with previous ones. All cycles per byte values are presented in the form of intervals where the lower (upper) bound of interval is calculated from processor base (maximal) frequency.

Table 3: Performance comparison

Paper	Processor	MBytes/s	Cycles/Byte	Notes
[2]	Core i7-2600 @ 3.4 GHz	138	23.5 – 26.3	non-parallelizable mode
[3]	Core i5-2500K @ 3.3 GHz	135	23.3 – 26.1	ECB mode
		129	24.4 – 27.4	CTR mode
[5]	Core i5-6500 @ 3.2 GHz	335	9.1 – 10.2	CTR mode
This paper	Core i7-6700 @ 3.4 GHz	360	9 – 10.6	parallelizable mode
		170	19.1 – 22.4	non-parallelizable mode
		255	12.7 – 15	byte slicing implementation

6 Conclusion

In this paper, performance issues of the Kuznyechik block cipher when implemented in software were considered. For implementations using look-up tables we described trade-offs for the size of LUTs and the number of operations needed. We gave particular insights and recommendations on how to use the resources of Intel CPUs to speed up Kuznyechik. Finally we presented the speed values for practical implementations on CPUs of various generations, using different compilers.

The best our speed results in one thread on one CPU core were 360 MB/s and 170 MB/s for parallelizable and non-parallelizable cipher modes of operations respectively. On 4 physical cores we achieved 1340 MB/s in parallelizable mode. We also presented the fairly effective byte slicing implementation of the Kuznyechik and attained the speed of 255 MB/s.

References

[1] "GOST R 34.12-2015 – National standard of the Russian Federation – Information technology – Cryptographic data security – Block ciphers", 2015.

[2] Borodin M. A., Rybkin A. S., "High-speed software implementation of the Kuznyechik block cipher", *Information Security Problems. Computer Systems*, **3** (2014), 67–73.

[3] Alekseev E. K., Popov V. O., Prokhorov A. S., Smyshlyaev S. V., Sonina L. A., "On the performance of one perspective LSX-based block cipher", *Math. Asp. Crypt.*, **6**:2 (2015), 7–17.

[4] Fomin D. B., "Implementation of an XSL block cipher with MDS-matrix linear transformation on NVIDIA CUDA", *Math. Asp. Crypt.*, **6**:2 (2015), 99–108.

[5] Ahmetzyanova L., Alekseev E., Oshkin I., Smyshlyaev S., Sonina L., "On the properties of the CTR encryption mode of the Magma and Kuznyechik block ciphers with re-keying method based on CryptoPro

Key Meshing", *Pre-proceedings of 5th Workshop on Current Trends in Cryptology*, CTCrypt 2016 (June 6-8, 2016, Yaroslavl, Russia), 42–54.

[6] Daemen J., Rijmen V., *The Design of Rijndael*, Springer-Verlag Berlin Heidelberg, 2002.

[7] Kasper E., Schwabe P., "Faster and timing-attack resistant AES-GCM", *Lecture Notes in Computer Science*, CHES 2009, **5747**, Springer, Berlin, Heidelberg, 2009, 1–17.

[8] Bernstein D. J., "Cache-timing attacks on AES", 2005.

[9] Biryukov A., Perrin L., Udovenko A., "Reverse-Engineering the S-Box of Streebog, Kuznyechik and STRIBOBr1", *Lecture Notes in Computer Science*, EUROCRYPT 2016, **9665**, Springer, Berlin, Heidelberg, 2016, 372–402.

[10] "GOST R 34.13-2015 – National standard of the Russian Federation – Information technology – Cryptographic data security – Modes of operation for block ciphers", 2015.

On the security properties of Russian standardized elliptic curves

Evgeny Alekseev, Vasily Nikolaev, Stanislav Smyshlyaev

Abstract

In the last two decades elliptic curves have become a sufficient part of numerous cryptographic primitives and protocols. Hence it is extremely important to use the elliptic curves, that do not break security of such protocols. This paper is about the elliptic curves used with GOST R 34.10-2001, GOST R 34.10-2012 and the accompanying algorithms, their security properties and generation process.

Keywords: elliptic curve, GOST R 34.11-2012

1 Introduction

Russian national cryptographic standards employ elliptic curve cryptography since 2001, when the national digital signature standard GOST R 34.10-2001 ([1]) was adopted. Since this standard provided no set of elliptic curves except for the test usage, there was a need to generate standardized curves which provided reasonable implementation speed and safety as well as compatibility of different implementations. Thus, three 256-bit curves were generated and defined in [2].

In 2012 new national digital signature standard GOST R 34.10-2012 was adopted ([3]). This standard introduces 256- and 512-bit signature schemes. Since GOST R 34.10-2001 introduced only a 256-bit version of the scheme, it was necessary to create a set of 512-bit elliptic curves. The other problem was generation of elliptic curves that provided higher performance of high-level protocols. The perspectives of using of different type curves (including twisted Edwards curves and Montgomery curves) with the national standard GOST R 34.10-2012 and the algorithms accompanying the usage of the GOST R

34.11-2012 and GOST R 34.10-2012 national standards (defined in [4], [5]) were studied in several papers including [6].

Four new elliptic curves (including two twisted Edwards curves) were adopted by the Technical committee for standardization «Cryptography and security mechanisms» as a standardization recommendation ([7]).

In this paper we are going to describe the process of generation of Russian standardized curves and provide some security considerations.

2 Notations

In this paper we use the following notations.

- p – characteristic of a finite field on which an elliptic curve is defined;
- a, b – short Weierstrass equation coefficients ($y^2 = x^3 + ax + b$);
- ε, δ – twisted Edwards curve equation coefficients ($\varepsilon u^2 + v^2 = 1 + \delta u^2 v^2$);
- $E(a, b)$ – the group of the points of the elliptic curve defined by short Weierstrass equation with coefficients a and b;
- $E(\varepsilon, \delta)$ – the group of the points of the elliptic curve defined by a twisted Edwards curve equation with coefficients ε and δ;
- m – the order of the elliptic curve points group;
- q – a prime number, the order of the prime subgroup of the elliptic curve points group;
- P – a base point of the prime subgroup of the elliptic curve points group;
- O – neutral element of the elliptic curve points group;
- m' – the order of the non-trivial quadratic twist points group;

- q' – a prime number, the order of the prime subgroup of the non-trivial quadratic twist points group.

  By $\oplus$ we denote elliptic curve point addition. By $[a]P$ we denote scalar point multiplication (point P is here multiplied by integer a).

3 Curve generation

All standardized elliptic curves observed in this paper were generated with accordance to the verifiable pseudo-randomness principle. This curve generation principle allows to show that there are no properties of the elliptic curve known only by it's developers. This property is met by selecting curve parameters as an output of a «one-way» function applied to a random seed. The knowledge of this random argument proves the fact that none of the curve parameters could be directly manipulated.

The curves, which can be written in twisted Edwards form were selected by generating δ value. The GOST R 34.11-2012 hash function (Streebog, [8]) was used as a «one-way» function. As only 512-bit variant of the function was used, last 32 bytes of the output were taken when generating 256-bit curves. The resulted values were then taken modulo p. The random seeds and the resulting values of δ of the standardized curves are presented in Appendix C.

The Weierstrass curves were generated by selecting a verifiable pseudo-random $k = a^3/b^2$ value and assuming $a = -3 \bmod p$ (such a value of a allows to create efficient implementations of elliptic curve arithmetic [9]). The method of selecting value k is similar.

All base points of the standardized curves were selected by iterating abscissa (x for short Weierstrass form and u for twisted Edwards form) beginning with 0 till the point was in a subgroup of order q.

The use of such a generation procedure guarantees that it is impossible for developers to add vulnerabilities into resulting elliptic curves. We would like to mention that a so called «BADA55 curves» attack ([10]) does not violate the last statement since it requires the existence of a large publicly unknown class of weak curves. We suspect it is very unlikely that such a class could

stay undiscovered for decades.

4 Cryptographic properties of the Russian standardized elliptic curves

4.1 Common computational problems on elliptic curves

In this paper we consider the following computational problems on the elliptic curve points groups.

- Discrete logarithm problem (DLP)
- Computational Diffie-Hellman problem (CDH)
- Decisional Diffie-Hellman problem (DDH)
- Discrete semilogarithm problem (DSLP)

It was shown in [11] that intractability of DSLP is a sufficient criterion of GMR-security (existential unforgeabilty under adaptive chosen message attack) of the digital schemes, defined in [1] and [3], in the tamper-proof device model. The fact, that intractability of DDH is a sufficient condition of the VKO key agreement scheme (defined in [4], [5]), was shown in [12], while intractability of the CDH problem is here the necessary condition.

It is easy to see that the following reductions are correct:

$$\mathsf{DDH} \rightarrow \mathsf{CDH} \rightarrow \mathsf{DLP} \leftarrow \mathsf{DSLP}$$

This statement implies that intractability of DLP is a necessary condition of intractability of all the computational problems observed and therefore of safety of all the cryptographic primitives and protocols discussed.

4.2 Generic algorithms

Here we show that the common algorithms, which do not rely on inner structure of elliptic points groups but on some generic group assumptions only, do not violate security of cryptographic primitives and protocols using

Russian standardized curves. The conditions in which such algorithms work are described by the generic group model ([13]). This paper also provides the results on the lower bounds of complexity of solving DDH, CDH and DLP. The same result for a similar group model was shown in [14]. These results imply that the best generic algorithm that solves the mentioned tasks is Pollard's ρ-method ([15]) which has square-root complexity.

DSLP can be easily solved when one can efficiently solve DLP. Currently there are no known results that showed this problem could be solved in the other way. This implies that Pollard's ρ-method remains the best method that solves DSLP.

According to the requirements of [3] the order q of the elliptic curve points subgroup used in the standardized digital signature schemes should be at least 2^{254} for the 256-bit fields and at least 2^{508} for the 512-bit fields. Since these requirements are met for all Russian standardized curves, it is computationally infeasible to mount a successful attack based on Pollard's ρ-method.

4.3 Specific algorithms

The complexity of the computational problems discussed above is estimated on the base of known results. The report [16] contains a summary based on observation of a major amount of publications on the complexity of these problems. It concludes the following statements.

- The fastest known algorithm that solves DLP is Pollard's ρ-method.
- The fastest known algorithm that solves CDH is finding discrete logarithm of one of public keys.
- The fastest known algorithm that solves DDH is finding discrete logarithm of one of public keys.

It is also mentioned that these three problems can be solved more effectively than using Pollard's ρ-method in the groups in which there exist some effectively computable isomorphisms (so called «pairing groups»).

There are no known results on the methods of solving DSLP which had lower complexity than computing corresponding discrete logarithm. Thus, Pollard's ρ-method remains the most effective method for solving DSLP.

All the studied elliptic curves were generated according to the requirements of the national standard [3].

- $2^{254} < q < 2^{256}$ for the 256-bit curves and $2^{508} < q < 2^{512}$ for the 512-bit curves. All the generated curves satisfy this condition.
- P belongs to the curve, $P \neq O$ and $[q]P = O$. This condition is also satisfied for all curves.
- $4a^3/b^2 + 27 = 0 \bmod p$. When this inequality does not hold, the discriminant of the curve is equal to 0. This means the curve is not smooth and therefore elliptic. All the studied curves satisfy this condition.
- $b \neq 0 \bmod p$. If this condition is violated then j-invariant of the curve equals 1728 what violates the requirements of [3]. All the studied curves satisfy this condition.
- $a^3/b^2 \neq 0 \bmod p$. When this condition is violated then $a = 0$. This implies j-invariant is equal to 0 what violates the requirements of [3]. All the studied curves satisfy this condition.

Further we study the properties of the elliptic curves which could decrease the complexity of the main computational problems on the elliptic curve points groups and thus lead to vulnerabilities in cryptographic primitives and protocols using such curves. We are going to show that Russian standardized curves do not have these properties.

There are several properties of elliptic curve points groups which can lead to weaknesses when violated. Here we base on the document [17] and the recommendations [18].

Condition 4.1. $(m, p) = 1$.

According to [19] for the elliptic curve points groups that do not satisfy this criterion it is possible to build effective isomorphism (for which evaluation of

a value takes $O(ln(p))$ time) to the additive group of F_p, therefore one can easily solve DLP in such groups in polynomial time.

It's easy to see that all the studied curves satisfy this condition.

Condition 4.2. $\operatorname{ord} p$ *in* $\mathbb{Z}_q^*$ *should not be small.*

There exist algorithms ([20]) which solve DLP in prime subgroups of the elliptic curve points groups based on embedding of these groups into finite fields multiplicative subgroups. In order to increase complexity of calculating of these embeddings (what implies high complexity of the algorithms solving DLP) one should check that $\operatorname{ord} p$ in $\mathbb{Z}_q^*$ is large enough. The document [17] recommends to select curves in such a way that inequality $\operatorname{ord} p > (q-1)/100$ holds.

In order to calculate the value of $\operatorname{ord} p$ effectively one has to factor $q-1$. The factorings of $q-1$ for the curves observed and the values of $\operatorname{ord} p$ in $\mathbb{Z}_q^*$ can be found in Appendix D. It easy to see that these values satisfy the condition. This implies it is not possible to calculate embeddings effectively so there existed a DLP solving algorithm that had lower complexity than Pollard's ρ-method.

Condition 4.3. *Complex multiplication discriminant should not be «small».*

According to SafeCurves ([18]) Pollard's ρ-method can be made more efficient for the curves having small complex multiplication discriminant D. SafeCurves propose to use elliptic curves for which inequality $|D| > 2^{100}$ holds.

The values of discriminants are given in Appendix E. One can see that this condition is satisfied for all the examined curves except for the curve id-GostR3410-2001-CryptoPro-B-ParamSet. This does not make this curve practically exploitable, however.

Condition 4.4. *The largest prime subgroup of the non-trivial quadratic twist points group should be secure.*

SafeCurves ([18]) proposes that the curve should be twist-secure in order to prevent attacks on scalar point multiplication using one coordinate formulae like Montgomery ladder. The orders of the twisted elliptic curve

points subgroups were estimated and checked to be secure against common attacks (these parameters can be found in Appendix F). These checks include security against attacks bases on Pollard's ρ-method and small embedding degrees.

These requirements are not held for the curves GostR3410-2001-CryptoPro-A-ParamSet, GostR3410-2001-CryptoPro-C-ParamSet and id-tc26-gost-3410-12-512-paramSetB. These curves, however, cannot be transformed into Montgomery form so the attacks based on the use of the Montgomery ladder are not applicable. Nevertheless, we highly recommend to add checks that a point belongs to a curve in every implementation employing one coordinate formulae.

Condition 4.5. *The F_p^* group should not have many small subgroups.*

In 2016 Petit, Kosters and Messeng proposed an algorithm ([21]) that solves DLP in the groups of elliptic curve point defined over large characteristic finite fields. This algorithm is based on the factor base algorithm. Though the authors did not provide exact complexity estimations, they supposed that curves defined over fields with characteristic p, where $p-1$ could be split into small primes, could be vulnerable to such an attack.

Here we try to obtain criteria which guarantees that implementing such an attack is more complex then implementing Pollard's ρ-method when it is met. Suppose $p-1 = p_1 \cdot \ldots \cdot p_s$, where all p_i are prime and not necessarily distinct. For some fixed $B \in \mathbb{N}$ let us set $k = \log_2(\prod_{j=1}^{n'} p_{i_j}) + 1$, where $p_{i_j} \in \{p_1, \ldots, p_s\}$, $p_{i_j} < B$ and $n = \lfloor \log_2(p) \rfloor$, $m = \lfloor n/k \rfloor$. We will denote as d_{max} the value $\max\{4, p_{i_1}, \ldots, p_{i_{n'}}\}$.

Heuristic estimation 4.6. *If the following inequality holds for every $B < n/4$:*

$$k + m(\log_2 m - 1) + 2d_{max} \log_2 mn' > n/2,$$

then the proposed method is computationally harder then Pollard's ρ-method.

The proof of this estimation and the proof of the fact that this condition is satisfied for all the curves can be found in Appendix A.

5 Conclusion

In the papers observed it was shown there are no generic methods solving DLP more efficiently than Pollard's ρ-method. These papers also state that the best way to solve DDH, CDH or DSLP is to compute a value of corresponding discrete logarithm. So there are no known efficient generic algorithms that solve these problems.

We have also shown that none of known specific conditions (including the one proposed by Petit, Kosters and Messeng) that may lead to speed-ups in solving DLP holds for the examined curves.

The curve generation procedure was also presented here. This method employs a hash function in order to create values b and δ for the short Weierstrass and twisted Edwards form respectively. One can see that it hardens intentional embedding of the properties that could lead to vulnerabilities.

Thus we can conclude the use of Russian standardized curves does not violate security of the national standards and other high-level primitives.

References

[1] Information technology. Cryptographic data security. Signature and verification processes of [electronic] digital signature, GOST R 34.10-2001, Gosudarstvennyi Standard of Russian Federation, Government Committee of Russia for Standards, 2001. (In Russian).

[2] Popov V., Kurepkin I., Leontiev S. «Additional Cryptographic Algorithms for Use with GOST 28147-89, GOST R 34.10-94, GOST R 34.10-2001, and GOST R 34.11-94 Algorithms», CRYPTO-PRO, January 2006, https://tools.ietf.org/html/rfc4357.

[3] Information technology. Cryptographic data security. Signature and verification processes of [electronic] digital signature, GOST R 34.10-2012, Federal Agency on Technical Regulating and Metrology, 2012.

[4] The use of cryptographic algorithms accompanying the usage of standards GOST R 34.10-2012 and GOST R 34.11-2012. Standardization

recommendations. Technical Committee 26. Federal Agency on Technical Regulating and Metrology, 2014.

[5] Smyshlyaev S. (Ed.), Alekseev E., Oshkin I., Popov V., Leontiev S., Podobaev V., Belyavsky D., «Guidelines on the Cryptographic Algorithms to Accompany the Usage of Standards GOST R 34.10-2012 and GOST R 34.11-2012», RFC 7836, 2016, https://tools.ietf.org/html/rfc7836.html.

[6] Alekseev E.K., Oshkin I.B., Popov V.O., Smyshlyaev S.V., Sonina L.A. «On the perspectives of the usage of twisted Edwards curves with the GOST R 34.11-2012 digital signature and the corresponding key agreement algorithm». Information Security Problems. Computer Systems. Volume 3, 2014, pp. 60-66.

[7] Information technology. Cryptographic data security. Parameters of elliptic curves for cryptographic algorithms and protocols, Federal Agency on Technical Regulating and Metrology, 2016. (In Russian).

[8] Information technology. Cryptographic Data Security. Hashing function, GOST R 34.11-2012, Federal Agency on Technical Regulating and Metrology, 2012.

[9] SEC 2. Standards for Efficient Cryptography Group: Recommended Elliptic Curve Domain Parameters. Version 1.0, 2000. www.secg.org/SEC2-Ver-1.0.pdf.

[10] Bernstein D. J., Chou T., Chuengsatiansup C., Huelsing A., Lambooij E., Lange T., Niederhagen R., van Vredendaal C. How to manipulate curve standards: a white paper for the black hat. http://bada55.cr.yp.to/pubs.html.

[11] Varnovskii N. P. Provable security of digital signatures in the tamper-proof device model, Discrete Mathematics and Applications. Volume 18, Issue 4, Pages 427-437.

[12] Alekseev E. K., Oshkin I. B., Popov V. O., Smyshlyaev S. V. , «On the cryptographic properties of algorithms accompanying the applications of

standards GOST R 34.11-2012 and GOST R 34.10-2012», Mathematical Aspects of Cryptography, 2016, v. 7, в„– 1, pp. 5-38 (In Russian).

[13] Shoup V. «Lower bounds for discrete logarithms and related problems», In Proc. EUROCRYPT 97, volume 1233 of Lecture Notes in Comput. Sci., pages 256-266. Springer-Verlag, 1997.

[14] Nechaev V. I., «Complexity of a determinate algorithm for the discrete logarithm», Mat. Zametki, 55:2 (1994), 91-101. (In Russian).

[15] Pollard J.M., Monte Carlo methods for index computation (mod p), Mathematics of Computation, 32 (1978), pp. 918-924.

[16] Vercauteren F. (editor). Final Report on Main Computational Assumptions in Cryptography, European Network of Excellence in Cryptology II. Start date of project: 1 August 2008. Duration: 4 years. ICT-2007-216676. http://www.ecrypt.eu.org/ecrypt2/documents/D.MAYA.6.pdf.

[17] The method of generating elliptic curve parameters satisfying GOST R 34.10-2012 (Project, in Russian).

[18] SafeCurves: choosing safe curves for elliptic-curve cryptography. https://safecurves.cr.yp.to/index.html.

[19] Semaev. I. A. «Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curve in characteristic p.» Mathematics of Computation 67 (1998), 353–356.

[20] Menezes A., Vanstone S. «Reducing elliptic curve logarithms to logarithms in a finite field», IEEE Trans. Inform. Theory, IT-39:5 (1993), 1639-1646.

[21] Petit C., Kosters M., Messeng A. <Algebraic Approaches for the Elliptic Curve Discrete Logarithm Problem over Prime Fields>, PKC 2016, Part II, LNCS 9615, pp. 3-18, 2016.

[22] Koblitz N. A course in number theory and cryptography. Springer-Verlag, 1987.

[23] Semaev I. A. Summation polynomials and the discrete logarithm problem on elliptic curves, Cryptology ePrint Archive: Report 2004/031.

[24] Semaev I. A. New algorithm for the discrete logarithm problem on elliptic curves, Cryptology ePrint Archive: Report 2015/310.

[25] Faugere J.-C. A new efficient algorithm for computing Groebner bases (F_4). Journal of Pure and Applied Algebra 139, 1-3 (June 1999), 61-88.

[26] Crandall R., Pomerance C. Prime numbers: a computational perspective. Springer-Verlag, 2001.

[27] Collins G. The calculation of multivariate polynomial resultants. Journal of the Association for Computing Machinery 18:515522, 1971.

[28] Diem C. On the discrete logarithm problem in elliptic curves II. 2011. Algebra & Number Theory, Vol. 7 (2013), No. 6, 1281-1323.

[29] Huang Y.-J., Petit C., Shinohara N., Takagi T. On Generalized First Fall Degree Assumptions. Cryptology ePrint Archive: Report 2015/358.

Appendices

A Resistance to Petit-Kosters-Messeng attack

Here we give a brief description of the algorithm proposed by Petit, Kosters and Messeng in [21] and prove Estimation 4.6. Here we denote by $\overline{K}$ algebraic closure of a field K and by expression «$a \in_R B$» we mean «a is taken uniformly random from the set B». We also denote by $Res_X(f(X), g(X))$ a resultant of polynomials $f(X)$ and $g(X)$ by variable X.

A.1 Algorithm description

The algorithm proposed by Petit, Kosters and Messeng in [21] uses so called factor bases ([22]). Suppose $\Omega = \{P_1, \ldots, P_N\}$ is some factor base which consists of elliptic curve points. For some fixed $m \in \mathbb{N}$ (we will provide comments on this value later) we need to build $N + 1$ equations:

$$[\alpha_i]P \oplus [\beta_i]Q = \bigoplus_{j=1}^{m} W_j,\ W_j \in \Omega,\ \alpha_i, \beta_i \in_R \{1, \ldots, q-1\}.$$

We obtain a $(N+1) \times N$ matrix of coefficients. Then using Gaussian elimination we can obtain equation $[\alpha]P \oplus [\beta]Q = O$ and deduce the value of the discrete logarithm as $-\alpha\beta^{-1} (\mathrm{mod} q)$.

These equations can be build with the use of Semaev polynomials defined in [23].

Definition A.1. *The i-th Semaev summation polynomial $S_i(x_1, \ldots, x_i) \in \mathbb{F}_p[x_1, \ldots, x_i]$, $i \geq 2$, is a polynomial which turns to 0 on the input $x_1, \ldots, x_i \in \overline{\mathbb{F}}_p$ if and only if $\exists y_1, \ldots, y_i \in \overline{\mathbb{F}}_p$: $(x_1, y_1) \in E(\overline{\mathbb{F}}_p), \ldots, (x_i, y_i) \in E(\overline{\mathbb{F}}_p)$ such that $\bigoplus_{j=1}^{i}(x_j, y_j) = O$.*

Theorem A.2 (I. A. Semaev, [23])**.**

1. *The following equalities hold:*

$$S_2(x_1, x_2) = x_1 - x_2,$$

$$S_3(x_1, x_2, x_3) =$$

$= (x_1 - x_2)^2 x_3^2 - 2((x_1 + x_2)(x_1 x_2 + a) + 2b)x_3 + ((x_1 x_2 - a)^2 - 4b(x_1 + x_2))$,

$S_n(x_1, \ldots, x_n) = Res_X(S_{n-k}(x_1, \ldots, x_{n-k-1}, X), S_{k+2}(x_{n-k}, \ldots, x_n, X))$,

for any $n \geq 4, 1 \leq k \leq n-3$.

2. *All polynomials are symmetric and have degree* 2^{n-2} *in each variable when* $n \geq 3$.

It is easy to see that the total degree of i-th Semaev polynomial does not exceed $i \cdot 2^{i-2}$ when $i > 2$. Indeed, the monomial $x_1^{2^{i-2}} \cdot \ldots \cdot x_i^{2^{i-2}}$ has the highest total degree in the i-th Semaev polynomial.

Suppose an elliptic curve point R has coordinates (R_x, R_y). The problem of decomposition of R into a sum of m points of $E(\overline{\mathbb{F}}_p)$ can be expressed as:

$$S_{m+1}(x_{1,1}, \ldots, x_{m,1}, R_x) = 0.$$

Since we have to decompose R not just into m random points but into the points from the factor base Ω we have to add constraints on the variables $x_{1,1}, \ldots, x_{m,1}$. The authors of [21] suppose to write down these constraints as a system of additional polynomial equations which is satisfied if and only if a point specified by its coordinates belongs to Ω.

The authors of [21] propose to define factor base Ω as a set of points which abscissa belongs to some subgroup of a multiplicative group of a finite field the curve is defined on. Suppose the following equality holds:

$$p - 1 = r \prod_{i=1}^{n'} p_i,$$

where p_i are primes (not necessarily pairwise distinct) such that $p_i < B$ for some natural B for all $i \in \{1, \ldots, n'\}$. Let us select a subgroup with order equals to $\prod_{i=1}^{n'} p_i$ as an abscissa subgroup. Then the polynomial system of constraints can be written as follows:

$$\begin{cases} x_{i,j+1} - x_{i,j}^{p_j} = 0, i = 1, \ldots, m, \ j = 1, \ldots n' - 1 \\ x_{i,n'}^{p_{n'}} - 1 = 0, i = 1, \ldots, m \end{cases} \tag{A.1}$$

Here $x_{i,j}$, $j \geq 2$, are additional variables for decreasing of degrees of the constraint equations added to the system.

Finally we obtain an equation system over $\mathbb{F}_p$ which consists of $mn'+1$ equations with mn' variables:

$$\begin{cases} S_{m+1}(x_{1,1}, \ldots, x_{m,1}, R_x) = 0 \\ x_{i,j+1} - x_{i,j}^{p_j} = 0, i = 1, \ldots, m,\ j = 1, \ldots n'-1 \\ x_{i,n'}^{p_{n'}} - 1 = 0, i = 1, \ldots, m \end{cases} \tag{A.2}$$

The maximal degree of an equation in the system equals $\max(B, (m+1) \cdot 2^{m-1})$. Here we can also try to build a system with a lower maximal degree using the technique proposed by Semaev in [24]. We change one $S_{m+1}(x_{1,1}, \ldots, x_{m,1}, R_x) = 0$ equation into a set of equations employing S_3 polynomial. The system (A.2) can be overwritten as:

$$\begin{cases} S_3(x_{1,1}, x_{2,1}, u_1) = 0 \\ S_3(u_i, x_{i+2,1}, u_{i+1}) = 0,\ i \in 1, \ldots, m-3 \\ S_3(u_{m-2}, x_{m,1}, R_X) = 0 \\ x_{i,j+1} - x_{i,j}^{p_j} = 0,\ i = 1, \ldots, m,\ j = 1, \ldots n'-1 \\ x_{i,n'}^{p_{n'}} - 1 = 0,\ i = 1, \ldots, m \end{cases} \tag{A.3}$$

The new system (A.3) consists of $mn'+m-1$ equations over $\mathbb{F}_p$ with $mn'+m-2$ variables. The equations have now maximal degree equal to $\max(B, 4)$.

The (A.2) and (A.3) could be solved with the means of Grobner bases. In order to build such a basis one can use common algorithms like F_4 ([25]).

A.2 Complexity estimations

Complexity of the method described can be estimated for each of three parts the algorithm consists of. First part is the initialization step. Second part is the decomposition generation step. The third part is the linear algebra step. Now we are going to estimate complexity of each of the steps.

The initialization step is performed once for each curve. One has to factor n-bit length value $p-1$ what can be done in $T_{factor} = O(2^{c_1 n^{1/3} \log_2^{2/3} n})$ using

number field sieve method ([26]). The other task is to generate $(m+1)$-th Semaev polynomial. It can be done in $T_{Sem} = O(2^{m^2})$ (according to the complexity of building resultant of two polynomials obtained in [27]). If one uses (A.3), then only third Semaev polynomial is used. This polynomial is implicitly defined in [23] therefore here $T_{Sem} = 0$.

Now we are going to estimate the complexity of the third step. After collecting all the decompositions we obtain a sparse matrix with size $(N+1) \times N$, which has only m non-null n-bit elements in each row. Complexity of finding a nontrivial equation between P and Q can be estimated as $T_{GE} = O(mnN^{\omega'})$, where $\omega' \leq 2$.

The main problem is to estimate complexity of the second step. One has to build $N+1$ decompositions of random elliptic curve points into m point of the factor base Ω. We will denote by T_{PDP} (PDP here means point decomposition problem) the complexity of building one such decomposition, thus the complexity of the second step equals $(N+1) \cdot T_{PDP}$. One obtains a decomposition by solving (A.2) or (A.3). We will denote the probability that one instance of the system has solutions as P_{PDP}. We will also denote the complexity of solving one instance of the system (A.2) or (A.3) as T_{trial}. Therefore $T_{PDP} = P_{PDP}^{-1} T_{trial}$. Following [24] the complexity of solving one instance of a system can be estimated as $T_{trial} = O(V^{\omega D_{reg}})$, where V is a number of variables, ω is a linear algebra constant and D_{reg} is the «regularity degree».

Following [28] we can estimate P_{PDP} as $\frac{N^m}{m!|E(\mathbb{F}_p)|}$. Suppose $p - 1 = r\prod_{i=1}^{n'} p_i,\ p_i < B = 2^l$. We will put $\prod_{i=1}^{n'} p_i \approx 2^k$ and $p \approx 2^n$. The size N of the factor base Ω can be estimated as $O(2^k)$. This can be justified as follows. For a random $x \in G \subset \mathbb{F}_p^*$, where G is a subgroup of $\mathbb{F}_p^*$ of order $\prod_{i=1}^{n'} p_i$, the probability that $x^3 + ax + b$ is quadratic residue modulo p is about $1/2$, while every solution gives us two points. Thus there exist about $O(2^{mk})$ sums of m points of the factor base Ω. Since the order of the summands does not change the sum we have to divide the number of sums by $m!$ in order to estimate the number of different sums. We can quite inaccurate estimate the number of the points using Hasse theorem as $O(2^n)$. Then we get the value $P_{PDP} = O(2^{mk-n}/m!)$.

In order to estimate the complexity of solving one instance of (A.2) and (A.3) one has to estimate the «regularity degree» of the corresponding polynomial ideal.

For some polynomial ring ideal $I \subseteq \mathbb{F}_p[x_1, \ldots, x_t]$ with basis $f_1(x_1, \ldots, x_t)$, $\ldots, f_s(x_1, \ldots, x_t)$ the following inequality holds ([29]):

$$D_{reg} \geq \max_{1 \leq i \leq s} \deg(f_i(x_1, \ldots, x_t)).$$

Thus the (A.2) system can be solved in $T_{trial} = O(2^{\log_2(mn')\omega D_{reg}})$, where $D_{reg} \geq \max(B, (m+1) \cdot 2^{m-1})$, while the (A.3) system for $T_{trial} = O(2^{\log_2(mn'+m-2)\omega D_{reg}})$, where $D_{reg} \geq \max(B, 4)$.

The total complexity of retrieving discrete logarithm T_{ECDLP} is:

$$T_{ECDLP} = T_{factor} + T_{Sem} + N \cdot P_{PDP}^{-1} T_{trial} + T_{GE},$$

what gives for the (A.2) system:

$$T_{ECDLP} = O(2^{c_1 n^{1/3} \log_2^{2/3} n}) + O(2^{m^2}) + O(2^{k + m\log_2 m - m + n - mk + \log_2(mn')\omega D_{reg}}) + \\ + O(2^{w'k + \log_2 m + \log_2 n}),$$

and for the (A.3) system:

$$T_{ECDLP} = O(2^{c_1 n^{1/3} \log_2^{2/3} n}) + O(2^{k + m\log_2 m - m + n - mk + \log_2(mn' + m - 2)\omega D_{reg}}) + \\ + O(2^{w'k + \log_2 m + \log_2 n}).$$

In order to avoid parameter m in the exponent, the authors of [21] suppose to assume $m = \lceil n/k \rceil$. Then for the both systems we can rewrite the complexity estimations.

$$T_{ECDLP} = O(2^{c_1 n^{1/3} \log_2^{2/3} n}) + O(2^{m^2}) + O(2^{k + m(\log_2 m - 1) + \log_2(mn')\omega D_{reg}}) + \\ + O(2^{w'k + \log_2 m + \log_2 n}),$$

$$T_{ECDLP} = O(2^{c_1 n^{1/3} \log_2^{2/3} n}) + O(2^{k + m(\log_2 m - 1) + \log_2(mn' + m - 2)\omega D_{reg}}) + \\ + O(2^{\omega'k + \log_2 m + \log_2 n}).$$

Now we are going to analyse the results in order to find out the values of m, k, n', B that make the complexity of the described method worse then

the Pollard's ρ-method, which has the complexity $T_{Pollard} = O(2^{n/2})$. We will denote by d_{max} the maximal degree of the polynomials emerging in the system. The following relations should hold:

$$\begin{cases} mk = n & \text{(A.4a)} \\ m^2 < n/2 & \text{(A.4b)} \\ 2B < n/2 & \text{(A.4c)} \\ k + m(\log_2 m - 1) + 2d_{max} \log_2 mn' < n/2 & \text{(A.4d)} \end{cases}$$

The last three inequalities are the sufficient conditions that building the required amount of decompositions will take less time then running Pollard's ρ-method. This relations system corresponds to the system (A.2). The corresponding relations system for (A.3) differs only in the absence of the (A.4b) inequality.

Thus, we have proved Estimation 4.6.

A.3 256-bit curves

1. $p = 2^{256} - 617$
   $p - 1 = 2 \cdot 7 \cdot 43 \cdot 9109 \cdot 87640387787 \cdot 16876409960174552741 \cdot$
   $\cdot 1427668375260843321126570913003304324345 3$

2. $p = 2^{255} + 3225$
   $p - 1 = 2^3 \cdot 11 \cdot 33797 \cdot 633062117 \cdot 43400749232432159 \cdot$
   $\cdot 39607009966486015397 \cdot 17888439653017795004024467$

3. $p = 7039008535208330519954771801901843784107951663004518047128434684370563350261 9$
   $p - 1 = 2 \cdot 17 \cdot 37 \cdot 113 \cdot 244997 \cdot 704476598345732707758923296 1 \cdot$
   $\cdot 2868963808335516896510938527146690437 01$

A.4 512-bit curves

1. $p = 2^{512} - 569$
   $p - 1 = 2 \cdot 23 \cdot 41 \cdot 353 \cdot 105095387 \cdot 45130584520747958722981 \cdot$
   $\cdot 582271299047893027187874292913927407 \cdot$

$\cdot 24405632944325884523100638769822040110 61\cdot$
$\cdot 29879361660612697647338220179191288608395313$

2. $p = 2^{511} + 111$
   $p - 1 = 2 \cdot 7 \cdot 17 \cdot 9433\cdot$
   $\cdot 29860769339941482698353859190482380663180854047591713557276$
   $48743353559431103560755510084456357598398009587265648506906928241789019762235105800049577$

A.5 The effectiveness of using Petit-Kosters-Messeng method for breaking Russian standardized elliptic curves

Keeping the fact that $2 < \omega \leq 3$ we have that for 256-bit curves B should be less than 64. No we are going to estimate m and k for each curve using (A.4a) and (A.4c).

Curve	Divisors	k	m	n'	d_{max} ((A.3) case)
1	2	1	256	1	4
1	$2 \cdot 7$	4	53	2	7
1	$2 \cdot 43$	6	39	2	43
1	$2 \cdot 7 \cdot 43$	10	25	3	43
1	7	2	91	1	7
1	$7 \cdot 43$	9	27	2	43
1	43	5	47	1	43
2	2	1	256	1	4
2	$2 \cdot 2$	2	128	2	4
2	$2 \cdot 2 \cdot 2$	3	85	3	4
2	$2 \cdot 2 \cdot 2 \cdot 11$	6	39	4	11
2	$2 \cdot 2 \cdot 11$	5	46	3	11
2	$2 \cdot 11$	4	57	2	11
2	11	3	74	1	11
3	2	1	256	1	4
3	$2 \cdot 17$	5	50	2	17
3	$2 \cdot 37$	6	41	2	37
3	$2 \cdot 17 \cdot 37$	11	22	3	37
3	17	4	62	1	17
3	$17 \cdot 37$	9	27	2	37
3	37	5	49	1	37

It is easy to see that (A.4d) does not hold for any of the parameters specified, so the method that uses (A.2) or (A.3) is slower then the Pollard's ρ-method.

Now we are going to study 512-bit curves. Here $B < 128$. Now we have the following limitations.

Curve	Divisors	k	m	n'	d_{max} ((A.3) case)
1	2	1	512	1	4
1	$2 \cdot 23$	5	92	2	23
1	$2 \cdot 41$	6	80	2	41
1	$2 \cdot 23 \cdot 41$	10	47	3	41
1	23	4	113	1	23
1	$23 \cdot 41$	9	51	2	41
1	41	5	95	1	41
2	2	1	512	1	4
2	$2 \cdot 7$	4	106	2	7
2	$2 \cdot 17$	5	100	2	17
2	$2 \cdot 7 \cdot 17$	7	64	3	17
2	7	2	182	1	7
2	$7 \cdot 17$	6	74	2	17
2	17	4	125	1	17

Here we see that (A.4d) does not hold for any of the parameter sets again, hence Petit-Kosters-Messeng method does not provide any improvement of DLP solving in comparison with Pollard's ρ-method.

A.6 On some probably vulnerable curve

The authors of [21] mentioned NIST P-224 elliptic curve as probably vulnerable. This curve is defined over $\mathbb{F}_p$, where $p = 2^{224} - 2^{96} + 1$. Value $p - 1$ can be factored as $2^{96} \cdot 3 \cdot 5 \cdot 17 \cdot 257 \cdot 641 \cdot 65537 \cdot 274177 \cdot 6700417 \cdot 67280421310721$. One can easily check having fixed $B = 56$ that inequality (A.4d) holds for some parameter sets. This fact, however, does not imply that method provided in [21] is computationally easier than Pollard's ρ-method for this curve.

B Parameters of the examined curves

Here we denote by $x(P)$ and $y(P)$ coordinates of the elliptic curve point P in short Weierstrass form. By $u(P)$ and $v(P)$ we denote coordinates of the elliptic curve point P in twisted Edwards form.

- id-GostR3410-2001-CryptoPro-A-ParamSet (256-bit Weierstrass curve)
  $p = 2^{256} - 617$
  $a = p - 3 = -3 \bmod p$
  $b = 166$
  $m = 115792089237316195423570985008687907853073762908499243225378155805079068850323$
  $q = m$
  $x(P) = 1$
  $y(P) = 64033881142927202683649881450433473985931760268884941288852745803908878638612$

- id-GostR3410-2001-CryptoPro-B-ParamSet (256-bit Weierstrass curve)
  $p = 2^{255} + 3225$
  $a = p - 3 = -3 \bmod p$
  $b = 28091019353058090096996979000309560759124368558014865957655842872397301267595$
  $m = 57896044618658097711785492504343953927102133160255826820068844496087732066703$
  $q = m$
  $x(P) = 1$
  $y(P) = 28792665814854611296992347458380284135028636778229113005756334730996303888124$

- id-GostR3410-2001-CryptoPro-C-ParamSet (256-bit Weierstrass curve)
  $p = 70390085352083305199547718019018437841079516630045180471284346843705633502619$
  $a = p - 3 = -3 \bmod p$
  $b = 32858$
  $m = 70390085352083305199547718019018437840920882647164081035322601458352298396601$
  $q = m$
  $x(P) = 0$
  $y(P) = 29818893917731240733471273240314769927240550812383695689146495261604565990247$

- id-tc26-gost-3410-12-512-paramSetA (512-bit Weierstrass curve)
  $p = 2^{512} - 569$
  $a = p - 3 = -3 \bmod p$
  $b = 121905800242662301561894247583400940755148440647362312522087723378253974644785404234189810743227188994270390889972216099473545205904486839481353008244181 44$
  $m = 134078079299425970995740249982058461274793658205923933777235614437217640300734492323182905858176364980496286125565968995006252799064166539938754747422931 09$
  $q = m$
  $x(P) = 3$
  $y(P) = 612856713215936837555067665053415337182670880790635313229604954686646454547260711913452922170333692151640510736902860619109774773836757192446669423679555 6$

- id-tc26-gost-3410-12-512-paramSetB (512-bit Weierstrass curve)
  $p = 2^{511} + 111$
  $a = p - 3 = -3 \bmod p$
  $b = 54725171305140472547604330712816572741710343895537697797479416031257965496939070366962372739527026378575800712932542409450794964843738542649984528870279 90$
  $m = 67039039649712985497870124991029230637396829102961966888617807218608820150369225854198537481903836150629109477434055675101483988207171002828568777761192 29$
  $q = m$
  $x(P) = 2$
  $y(P) = 13910877977955572587117358747504633286667292976475538607943404349820727624912779633246684899931850893657030334942041805681819055489680110759103577874927 97$

- id-tc26-gost-3410-2012-256-paramSetA (256-bit twisted Edwards curve)
  $p = 2^{256} - 617$
  $a = 87789765485885808793369751294406841171614589925193456909855962166505018127157$
  $b = 18713751737015403763890503457318596560459867796169830279

16251146174490100 2515
$\varepsilon = 1$
δ = 272441411047460593183426850116475764599872687847307680943260422341435167538 7
m = 1157920892373161954235709850086879078533542411923690137700486136351421214355 48
$q = m/4$ = 2894802230932904885589274625217197696333856029809225344251215340878553035 8887
$x(P)$ = 6598735018258456079030864061958683471210554512626975936540676896245329832605 6
$y(P)$ = 2285518920298496287042140250411039929315223538290810574174998740572132043529 2
$u(P) = 13$
$v(P)$ = 4377914498939898784342877916609043640693419582191518357445422440318617695050 3

- id-tc26-gost-3410-2012-512-paramSetC (512-bit twisted Edwards curve)
$p = 2^{512} - 569$
a = 1155220774172662408138485443175427045341999095815853654745363047275328427985602901303342173019597777291248497056097705489756374945796698516542818228427873 9
b = 9467654314974239364849779893497935997616546680893642377235981868741051215651032446828994750528267630604306101610711521055955290148577159125187794668181473
$\varepsilon = 1$
δ = 8291368582540391759956325599449696250171737838651241289585997940557058703682611983276933821315724273262456979490783602284025399599007870613061010570769744
m = 1340780792994259709957402499820584612747936582059239337772356144372176403007344846347320033739688509767539282340336658205886846512763738374217385971709125 2
$q = m/4$ = 3351951982485649274893506249551461531869841455148098344430890360930441007518362115868300084349221274418848205850841645514717116281909345935543464929272813

$x(P) = 11883046340949417535959253611031637438486121989357748$
$24796358501545516705356508594216113087093762259674783145997$
$95902458495903303153933228851862132220890 32$
$y(P) = 12873887912291418762163219174899249027788909354964279$
$56104470458407989428328693568863958710113734676526423783093$
$3785897290140286858111689735138773336704015$
$u(P) = 18$
$v(P) = 36979017503500364661955013706809651308929254455287941$
$06515700685530527913038331015106382234398842797314774264061$
$70232846972623627636989852682885080390713 3$

C Curve generation

Here we present the seeds used to generate twisted Edwards curves. By H_{512} we denote 512-bit variant of the Streebog hash function ([8]).

- id-tc26-gost-3410-2012-256-paramSetA:
  $seed$:
  97 B9 57 56 E0 59 D5 75 E6 05 99 7A FE A2 46 D7
  95 C6 84 FB 87 74 86 9A 9B 18 53 8F 13 F3 8A C2
  14 42 8D 87 1D 12 CA 10 9C 9A 4F E6 C4 DD F3 AF
  26 F6 38 6D 2D 51 4C CC B6 D1 AB 82 83 3C 30 D0
  79 8E 9B 30 37 FC 86 A3 B6 15 60 2E A1 56 E1 2D
  30 CD 63 53 E6 7D F6 42 82 D4 52 A2 03 3C BB 03

  $H_{512}(seed)$:
  AF 83 BE 4E C7 BA C6 DD ED 83 3C 44 F1 D0 CF 6A
  18 E7 B1 9D EA 5F 4E 69 BA E3 C2 CC B2 F9 9B 07
  06 05 F6 B7 C1 83 FA 81 57 8B C3 9C FA D5 18 13
  2B 9D F6 28 97 00 9A F7 E5 22 C3 2D 6D C7 BF FB

  Coefficient δ:
  $\delta = 0x605f6b7c183fa81578bc39cfad518132b9df62897009af7e522c32d6dc7bffb$

- id-tc26-gost-3410-2012-512-paramSetC:
  $seed$:
  1F BB 79 69 B9 1B 3E A0 81 17 FB 10 74 BF BF 55
  49 DD 66 07 63 F6 A5 AF 09 57 77 5B 66 4C B1 13
  CF CB 91 C4 A7 7D 27 98 06 BC F2 4A 56 77 F2 5E
  AF FE C6 67 76 70 2E E2 C7 AA 84 16 07 50 DA 1D
  D1 50 AE D2 8C 30 26 AC 7E D6 D1 9B 97 AC 2C B5
  82 7C 00 03 18 47 13 53 5B FA 65 24 B3 E4 60 83

  $H_{512}(seed)$:
  9E 4F 5D 8C 01 7D 8D 9F 13 A5 CF 3C DF 5B FE 4D
  AB 40 2D 54 19 8E 31 EB DE 28 A0 62 10 50 43 9C
  A6 B3 9E 0A 51 5C 06 B3 04 E2 CE 43 E7 9E 36 9E
  91 A0 CF C2 BC 2A 22 B4 CA 30 2D BB 33 EE 75 50

  Coefficient δ:
  $\delta = 0x9e4f5d8c017d8d9f13a5cf3cdf5bfe4dab402d54198e31ebde28a0$
  $621050439ca6b39e0a515c06b304e2ce43e79e369e91a0cfc2bc2a22b4ca3$
  $02dbb33ee7550$

D Embedding degrees

- id-GostR3410-2001-CryptoPro-A-ParamSet (256-bit Weierstrass curve)
  $q - 1 = 2 \cdot 3 \cdot 7 \cdot 17 \cdot 37 \cdot 127 \cdot 121493 \cdot 5592900119 \cdot$
  $\cdot 5079101754045001507145635028404503716993685576540874858 1$
  $\operatorname{ord} p = (q-1)/2$

- id-GostR3410-2001-CryptoPro-B-ParamSet (256-bit Weierstrass curve)
  $q - 1 = 2 \cdot 47336631894758162101 \cdot$
  $\cdot 61153531948973788036176540086765671793320126452664074625 1$
  $\operatorname{ord} p = (q-1)/2$

- id-GostR3410-2001-CryptoPro-C-ParamSet (256-bit Weierstrass curve)
  $q - 1 = 2 \cdot 2 \cdot 2 \cdot 3 \cdot 3 \cdot 5 \cdot 5 \cdot 47 \cdot 207130852417 \cdot 15398703602419036183 \cdot$

·26086281509712031326282791416212949263931 1
ord $p = q - 1$

- id-tc26-gost-3410-12-512-paramSetA (512-bit Weierstrass curve)
  $q - 1 = 2 \cdot 2 \cdot 19 \cdot 41 \cdot 257 \cdot 619681 \cdot 56230387 \cdot 3067250436697090551527 \cdot$
  $\cdot 15665300049585351442709496647505686649347704944533808966395$
  46221087773012987381099702230506073453913058415009288 11
  ord $p = q - 1$

- id-tc26-gost-3410-12-512-paramSetB (512-bit Weierstrass curve)
  $q - 1 = 2 \cdot 2 \cdot 3 \cdot 2389 \cdot 233846238487906325861134801838388553918643$
  88855272846619527619897398103116728380977194032980392906242173347249467 81797012383662616229653621748489178792 1
  ord $p = (q - 1)/6$

- id-tc26-gost-3410-2012-256-paramSetA (256-bit twisted Edwards curve)
  $q - 1 = 2 \cdot 1597 \cdot$
  $\cdot 90632505664774730293966018322391912847021165617070298818134$
  48155537110319
  ord $p = q - 1$

- id-tc26-gost-3410-2012-512-paramSetC (512-bit twisted Edwards curve)
  $q-1 = 2 \cdot 2 \cdot 2819 \cdot 7673 \cdot 1683089 \cdot 8490713317 \cdot 11424320126732366161793 \cdot$
  $\cdot 13932905458310476305278334845959 \cdot$
  $\cdot 17031551406759063641230097822523168865377062804005237213018$
  459123037212618699
  ord $p = q - 1$

E Complex-multiplication discriminants

- id-GostR3410-2001-CryptoPro-A-ParamSet (256-bit Weierstrass curve)
  $|D| = 16986615158945491831825635540471029757001709141841479467548269980054174491650 68$
  $|D| > 2^{100}$

- id-GostR3410-2001-CryptoPro-B-ParamSet (256-bit Weierstrass curve)
  $|D| = 2476$
  $|D| < 2^{100}$

- id-GostR3410-2001-CryptoPro-C-ParamSet (256-bit Weierstrass curve)
  $|D| = 225582403534449087570893262906079377782519721590934732343094982397458255928460$
  $|D| > 2^{100}$

- id-tc26-gost-3410-12-512-paramSetA (512-bit Weierstrass curve)
  $|D| = 176308990594895101831812239635692069490438040091159251133715331100958895534786995306510452439046262349503427868505276858381717314005007176133917595472554188$
  $|D| > 2^{100}$

- id-tc26-gost-3410-12-512-paramSetB (512-bit Weierstrass curve)
  $|D| = 733711169610957163035988136099483742961244593884181083755723481156762781921366218027107268488731306020886898973802079016197327401725226330890103743766060$
  $|D| > 2^{100}$

- id-tc26-gost-3410-2012-256-paramSetA (256-bit twisted Edwards curve)
  $|D| = 456069194652922817833172763729828275844502106275118018404003399379926643529292$
  $|D| > 2^{100}$

- id-tc26-gost-3410-2012-512-paramSetC (512-bit twisted Edwards curve)
  $|D| = 439263557933465671521763589803629925633969797171182125737961818880352371042717993365956402222756235707426276397732405345545163942770808406543853243667393 2$
  $|D| > 2^{100}$

F Twist security

Here we present auxiliary data needed to analyse twist security of Russian standardized curves. For each curve we provide the number of point of the

non-trivial quadratic twist (m'), the order of largest prime subgroup (q') and order of p in $Z^*_{q'}$. We do not estimate complex multiplication discriminants of the twists as they are equal to the discriminants of the original curves.

- id-GostR3410-2001-CryptoPro-A-ParamSet (256-bit Weierstrass curve)
  $m' = 115792089237316195423570985008687907853469620642278188485353701221074719042831 7$
  $q' = m'/(67 \cdot 1197515797 \cdot 112960388171533961) = 1277605168546469310381585958728239820766318491700 3$
  $q' < 2^{200}$
  $q' \neq p$
  $q'-1 = 2{\cdot}3{\cdot}157{\cdot}179{\cdot}1553{\cdot}4878892741094933323156995445533251695331 3$
  $\operatorname{ord} p = (q' - 1)/2$

- id-GostR3410-2001-CryptoPro-B-ParamSet (256-bit Weierstrass curve)
  $m' = 57896044618658097711785492504343953926167851505384737219388739511825397579685$
  $q' = m'/(5 \cdot 113093) = 1023866103448632500893697974310416275563790004781635242135034697316817 09$
  $q' > 2^{200}$
  $q' \neq p$
  $q' - 1 = 2^2 \cdot 3^3 \cdot 6113 \cdot 155083292959241764802045727428251915402480143225674979572228386577$
  $\operatorname{ord} p = (q' - 1)/3$

- id-GostR3410-2001-CryptoPro-C-ParamSet (256-bit Weierstrass curve)
  $m' = 70390085352083305199547718019018437841238150612926279907246092229058968608639$
  $q' = m/(11 \cdot 433 \cdot 1951561 \cdot 9245309 \cdot 104138341) = 7865325388019627074611138979082913271426129070825717$
  $q' < 2^{200}$
  $q' \neq p$
  $q - 1 = 2^2 \cdot 3469 \cdot 259421 \cdot 531572647637 \cdot 426526067872013 \cdot \cdot 9636938087179541$
  $\operatorname{ord} p = (q' - 1)/2$

- id-tc26-gost-3410-12-512-paramSetA (512-bit Weierstrass curve)
  $m' = 13407807929942597099574024998205846127479365820592393377723561443721764030073644721285458010516170357330435103816375202206882485717476485898991823269873947$
  $q' = m'/(23{\cdot}61{\cdot}4447{\cdot}142799{\cdot}205285055011140558581260164490369) = 7330784956201792373395615021787309692168767081465985083259778903069744578953826754768405708308045303549039425 7$
  $q' > 2^{200}$
  $q' \neq p$
  $q' - 1 = 2^4 \cdot 3 \cdot 328302760325728843 \cdot 46519464666093586019730140327775155096140798164807710740756084752369475788815243431277307 29$
  $\operatorname{ord} p = (q' - 1)/16$

- id-tc26-gost-3410-12-512-paramSetB (512-bit Weierstrass curve)
  $m' = 67039039649712985497870124991029230637396829102961966888617807218608820150366243913820205499765198126271209104430804833436054839912294696635767712299650 91$
  $q' = m'/(107 \cdot 457 \cdot 11279 \cdot 288867653 \cdot 8115660434350138997 \cdot 7939453682920433906546174291390263043 \cdot 4818517832205402043366003520 2680240077) = 135528408814000099356299357826039308052791 21$
  $q' < 2^{200}$
  $q' \neq p$
  $q' - 1 = 2^4 \cdot 5 \cdot 347 \cdot 1021 \cdot 22063110703 \cdot 216729680875995769427397 49$
  $\operatorname{ord} p = (q' - 1)/10$

- id-tc26-gost-3410-2012-256-paramSetA (256-bit twisted Edwards curve)
  $m' = 11579208923731619542357098500868790785318572813891211430886655438068413784309 2$
  $q' = m'/4$
  $q' > 2^{200}$
  $q' \neq p$
  $q' - 1 = 2^2 \cdot 3^2 \cdot 97223288653 \cdot 827077279482546575190449379565233863604927641138408191826988590 9$
  $\operatorname{ord} p = (q' - 1)/12$

- id-tc26-gost-3410-2012-512-paramSetC (512-bit twisted Edwards curve)
  $m' = 134078079299425970995740249982058461274793658205923933777235614437217640300736454901305482589369217577046708929696055196486393004962557561506934382950758 04$
  $q' = m'/4$
  $q' > 2^{200}$
  $q' \neq p$

Some Security Comparisons of GOST R 34.10-2012 and ECDSA Signature Schemes

Trieu Quang Phong, Nguyen Quoc Toan

Abstract

The purpose of this article is to provide two security comparisons between GOST R 34.10-2012 and ECDSA. First, we compare GOST R 34.10-2012 with ECDSA via two flaws of ECDSA analyzed by J. Stern, D. Pointcheval, J. Malone-Lee and N.P. Smart. In particular, we obtain that GOST R 34.10-2012 is able to resist these two flaws of ECDSA. Second, we consider the security of their variants in the random oracle model. In more detail, J. Malone-Lee and N.P. Smart proposed two variants of ECDSA and proved that those variants are secure in the random oracle model. In a similar way, we also describe two variants of GOST R 34.10-2012 and then provide the security proofs of these variants in the random oracle model.

Keywords: GOST R 34.10-2012, ECDSA, random oracle model, no-message attack, ECTEGTSS, improved forking lemma.

1 Introduction

ECDSA [7] and GOST R 34.10-2012 [5] are considered as the secure and popular signature schemes. These schemes are the elliptic curve versions of DSA and GOST R 34.10-94, respectively. However, there are not much research comparing the efficiency and security of these schemes.

The common point between GOST R 34.10-2012 and ECDSA is that in these two schemes the value of the hash function only depends on the signed message. This implies that there is no security proof for these two schemes in the random oracle model.

Related Works

In [3], E. Brickell, D. Pointcheval, S. Vaudenay, M. Yung introduced design

validation for discrete logarithm based signature schemes, and then provide the security proof for these schemes in the random oracle model using *The Improved Forking Lemma*. Moreover, these authors presented two variants DSA-I and DSA-II of as their example.

In [1], J. Malone-Lee and N.P. Smart described two variants ECDSA-II and ECDSA-III of ECDSA which are secure against *the no-message attack* in the random oracle model using The Improved Forking Lemma.

In [2], J. Stern, D. Pointcheval, J. Malone-Lee and N.P. Smart provided two flaws of ECDSA, namely *duplicate signature* and *malleability*.

Our contributions

In this article, we will provide a comparison between GOST R 34.10-2012 and ECDSA by applying the method of J. Malone-Lee and N.P. Smart in [1] for GOST R 34.10-2012. As a consequence, we obtain two variants GOST-I and GOST-II of GOST R 34.10-2012 that are secure against the no-message attack in the random oracle model. This result is similar to the result for ECDSA in [1]. Besides, this article also provides another comparison between GOST R 34.10-2012 and ECDSA via two flaws of ECDSA described in [2], although these two flaws do not actually affect too much to the security of ECDSA.

Organization

This article includes five Sections: Section 1 presents the introduction. Section 2 provides the description of ECDSA and GOST R 34.10-2012. In Section 3, we will present our comparison result between GOST R 34.10-2012 and ECDSA via two flaws of ECDSA described in [2]. Section 4 represents the security proof results of J. Malone-Lee and N.P. Smart for two variants ECDSA-II and ECDSA-III of ECDSA, and then we apply their method to obtain two variants GOST-I and GOST-II of GOST R 34.10-2012 that are secure against the no-message attack in the random oracle model. Finally, our conclusion and future research are presented in Section 5.

2 GOST R34.10-2012 and ECDSA Schemes

2.1 Notations

The signature schemes in this paper are the elliptic curve based signature schemes. We now present the notations used in this paper before describing these schemes in details.

p	Prime number, $p > 3$.
$\mathbb{F}_p$	Finite prime field represented by a set of integers $\{0, 1, ..., p-1\}$.
$E(\mathbb{F}_p)$	Elliptic curve defined on $\mathbb{F}_p$.
$\lvert E(\mathbb{F}_p)\rvert$	The number of $\mathbb{F}_p$-rational points on $E(\mathbb{F}_p)$.
$\mathcal{O}$	Zero point of the elliptic curve $E(\mathbb{F}_p)$.
n	A prime divisor of $\lvert E(\mathbb{F}_p)\rvert$.
c	Cofactor, $c = \frac{\lvert E(\mathbb{F}_p)\rvert}{n}$.
P	Elliptic curve point of order n.
H, H_{GOST}	Hash function.
A	Signer.
$\mathcal{A}$	Attacker.
$\in_R$	Generate a random integer.
d	Integer number, the signature (private) key of signer A.
Q	Elliptic curve point, the verification (public) key of signer A.
k	Ephemeral secret value.
M	Signer's message.
(r, s)	digital signature for the message M.
x_R, y_R	Coordinates of elliptic point R.
$\log(x)$	Binary logarithm of x.

2.2 Description of GOST R 34.10-2012

GOST R 34.10-2012 is described in [5] as follow:

- Key Generation Algorithm (of signer A):
  1. Select $d \in_R [1, n-1]$.
  2. Compute $Q = dP$.

3. The private key of signer A is d.
4. The public key of signer A is Q.

- Signing Algorithm (A signs on message M):
  1. Calculate the message hash code M: $h = H_{GOST}(M)$.
  2. Calculate an integer α, the binary representation of which is the vector h, and determine $e = \alpha(mod\ n)$. If $e = 0$, then assign $e = 1$.
  3. Generate a random (pseudorandom) integer k, satisfying the inequality: $0 < k < n$.
  4. Calculate $R = kP = (x_R, y_R)$, and $r = f(R) = x_R\ mod\ n$, if $r = 0$ return to Step 3.
  5. Calculate $s = rd + ke\ mod\ n$; if $s = 0$, return to Step 3.
  6. The signature of A on M is (r, s).
- Verification Algorithm (the verified signature (r, s) on M of signer A):
  1. Verifying whether r, s belong to $[1, n-1]$ or not.
  2. Compute $h = H_{GOST}(M)$.
  3. Calculate an integer Oç, the binary representation of which is the vector h, and determine $e = \alpha(mod\ n)$. If $e = 0$, then assign $e = 1$.
  4. Compute $w = e^{-1} mod\ n$.
  5. Compute $u_1 = sw\ mod\ n$ and $u_2 = -rw\ mod\ n$.
  6. Compute $R = u_1P + u_2Q = (x_R, y_R)$ and $v = x_R\ mod\ n$.
  7. The signature is verified only if $v = r$.

2.3 Description of ECDSA

The signing and verification algorithms of ECDSA are described in [1] as follow:

- Signing Algorithm (A signs on message M):
  1. Select $k \in_R [1, n-1]$.

2. Compute $R = kP = (x_R, y_R)$, and $r = f(R) = x_R \ mod \ n$, if $r = 0$ return to Step 1.
3. Compute $h = H(M)$.
4. Compute $s = k^{-1}(h + dr) \ mod \ n$; if $s = 0$, return to Step 1.
5. The signature of A on M is (r, s).

- Verification Algorithm (the verified signature (r, s) on M of signer A):

1. Verifying whether r, s belong to $[1, n-1]$ or not.
2. Compute $w = s^{-1} \ mod \ n$.
3. Compute $h = H(M)$.
4. Compute $u_1 = hw \ mod \ n$ and $u_2 = rw \ mod \ n$.
5. Compute $R = u_1P + u_2Q = (x_R, y_R)$ and $v = x_R \ mod \ n$.
6. The signature is verified only if $v = r$.

3 Comparison of GOST R 34.10-2012 and ECDSA scheme via two flaws of ECDSA

In this section, we compare GOST R 34.10-2012 and ECDSA via two flaws of ECDSA in [2]. Our result is that GOST R 34.10-2012 scheme is able to resist the two flaws of ECDSA.

3.1 Two flaws of ECDSA

We now consider two flaws of ECDSA specified in [2].The cause of these flaws is that the function f in Step 2 of ECDSA signing algorithm has the property: $f(R) = f(-R), \forall R \in E(\mathbb{F}_p)$. In particular, the first flaw is *duplicate signature* – for any two distinct messages m_1 and m_2, we always can generate a signature which is valid for both messages, if we have a possible control on the key generation. (It is worth noting that this flaw is out of scope of the *duplicate signature* key selection attack (DSKS) defined in [9], since in the DSKS attack the "duplicate signature" only needs a message.) The second flaw is that from a signature (r, s) of a message m, one can derive a valid

second signature of m, namely $(r, -s)$.

The first flaw. For any two distinct messages m_1 and m_2, compute $h_1 = H(m_1)$ and $h_2 = H(m_2)$. We next generate $k \in_R \{1, .., n-1\}$, compute $r = f(kP)$, and then set the private key to be

$$d = -((h_1 + h_2))/2r \ mod \ n,$$

with the public key being given by $Q = dP$. Finally, we compute $s = k^{-1}(h_1 + dr) \ mod \ n$. We can see that (r, s) is a valid signature for both messages m_1 and m_2, with the public/ private key pair (Q, d). Indeed, it is obvious that (r, s) is a valid signature for m_1. On the other hand, from the ECDSA verification algorithm, compute

$$\begin{aligned} R' &= \frac{h_2}{s}P + \frac{r}{s}Q = \frac{(h_2 + rd)}{k^{-1}(h_1 + dr)}P \\ &= k\frac{h_2 - \frac{(h_1+h_2)}{2}}{h_1 - \frac{(h_1+h_2)}{2}}P = -kP. \end{aligned}$$

The second flaw. From a valid signature (r, s) of a message m, one can derive a valid second signature of m, namely $(r, -s)$. Indeed, since (r, s) is a valid signature on m, we have

$$\begin{aligned} r &= f(\frac{H(m)}{s}P + \frac{r}{s}Q) = f(-(\frac{H(m)}{s}P + \frac{r}{s}Q)) \\ &= f(\frac{H(m)}{-s}P + \frac{r}{-s}Q). \end{aligned}$$

Therefore, $(r, -s)$ is also a valid signature for m.

3.2 GOST R 34.10-2012 is resistant to two flaws of ECDSA

Before we show that GOST R 34.10-2012 can resist two flaws of ECDSA, let us analyze the cause of these flaws from the property of the function f in ECDSA.

The first flaw. In order to generate an ECDSA duplicate signature for two distinct messages, one can choose $k \in_R \{1, .., n-1\}$, and compute the first component of duplicate signature $r = f(kP)$. After that the private key d

and the second component of the duplicate signature are computed by solving the following equation system:

$$\begin{cases} s & = k^{-1}(H(m_1) + dr) \ mod \ n \\ s & = -k^{-1}(H(m_2) + dr) \ mod \ n. \end{cases}$$

We note that this equation always has solution $d = -\frac{H(m_1)+H(m_2)}{2r} \ mod \ n$, and $s = k^{-1}(H(m_1) + H(m_2)) \ mod \ n$. Therefore, we can always find a duplicate signature for two distinct messages by using this simple method.

The second flaw. In the ECDSA for verification algorithm for signature (r, s) and message m, we need to compute an elliptic point $R = s^{-1}H(m)P + s^{(} - 1)rQ$ and check the equality $f(R) = r \ mod \ n$. However, if the equation $f(R) = r \ mod \ n$ holds, then we also obtain the equation $f(-R) = r \ mod \ n$ holds. On the other hand, it is easy to compute the elliptic point $-R$ from $(r, -s)$ and m (by $-R = (-s)^{-1}H(m)P + (-s)^{-1}rQ$). It implies that $(r, -s)$ is a valid signature for m when $(r, -s)$ is a valid signature for m.

As above reason, we observe that, for any hash function, ECDSA still has these two flaws. Here, we give an informal argument to show that GOST R 34.10-2012 is able to resist these flaws if the values of the hash function are uniformly distributed.

GOST R 34.10-2012 is able to resist the first flaw. In order to make a duplicate signature for two distinct messages m_1 and m_2, one will compute d and s by solving an equation system. However, if we apply this method to GOST R 34.10-2012, we will obtain the equation system:

$$\begin{cases} s & = kH_{GOST}(m_1) + dr \ mod \ n \\ s & = -kH_{GOST}(m_2) + dr \ mod \ n. \end{cases}$$

where k is preselected and r is derived from k. It is easy to see that this equation system has a solution (d, s) if $H_{GOST}(m_1) = e_1 = -e_2 = -H_{GOST}(m_2) \ mod \ n$. However, the number of pairs (e_1, e_2) that $e_1 = -e_2 \ mod \ n$ is $n - 1$ (i.e, $(e_1, e_2) \in \{(i, n-i) | i = (1, .., n-1)\}$), and the number of pairs (e_1, e_2) (where $e_1, e_2 \in \{1, .., (n-1)\}$) is $(n-1)^2$. Therefore, using the assumption that the values of the hash function are uniformly distributed, we have the probability of two distinct message satisfying $H_{GOST}(m_1) = -H_{GOST}(m_2)$ is only

$(n-1)^{-1}$, this value is negligible because $2^{254} < n < 2^{256}$ or $2^{508} < n < 2^{512}$ in GOST R 34.10-2012. Therefore, the first flaw on GOST R 34.10-2012 is negligible.

GOST R 34.10-2012 is able to resist the second flaw. We will show that in the GOST R 34.10-2012 verification algorithm, if an elliptic point R is derived from a signature (r, s) and a message m, then its opposite point can not be derived from $(r, -s)$ and m. Therefore, the property $f(R) = f(-R)$ $(\forall R \in E(\mathbb{F}_p))$ is no longer to be exploited to make the second flaw in GOST R 34.10-2012. It implies that GOST R 34.10-2012 is able to resist the second flaw. Indeed, if we have (r, s) and m such that

$$\begin{cases} R & = wsP - wrQ \\ -R & = -wsP - wrQ. \end{cases}$$

where $w = H_{GOST}(m)^{-1}\ mod\ n$, then we obtain the equation

$$wsP - wrQ = -(-wsP - wrQ).$$

It is equivalent to

$$wrQ = \mathcal{O}.$$

This equation does not hold (because $gcd(wr, n) = 1$ and Q is the elliptic point of order n).

Moreover, even if we can find (r', s') such that $r' \neq r\ mod\ n$ and $-R$ is derived from (r', s') and m, then the verification is not valid. For example, one can compute the elliptic point from $(-r, -s)$ and m, and then the verification equation will be whether $f(-R) = -r\ mod\ n$ or not. This verification is not valid, because

$$f(-R) = f(R) = r \neq -r\ mod\ n.$$

Hence, we obtain that GOST R 34.10-2012 is able to resist the second flaw. □

4 Constructing two variant of GOST R 34.10-2012 in the way of ECDSA-II and ECDSA-III construction

4.1 Two variants of ECDSA

In this section, we will describe two variants of ECDSA in [1], and recall the security results for these two variants.

4.1.1 ECDSA-II

The first variant of ECDSA is called ECDSA-II, which replace the hash function evaluation $h = H(m)$ with $h = H(m||r)$. This variant is similar to the Pointcheval-Vaudenay scheme defined in [8]. In [1], ECDSA-II is described as follows.

- Signing Algorithm (A signs on message M):
  1. Select $k \in_R [1, n-1]$.
  2. Compute $R = kP = (x_R, y_R)$, and $r = f(R) = x_R \ mod \ n$, if $r = 0$ return to Step 1.
  3. Compute $h = H(M||r)$.
  4. Compute $s = k^{-1}(h + dr) \ mod \ n$; if $s = 0$, return to Step 1.
  5. The signature of A on M is (r, s).
- Verification Algorithm (the verified signature (r, s) on M of signer A):
  1. Verifying whether r, s belong to $[1, n-1]$ or not.
  2. Compute $w = s^{-1} \ mod \ n$.
  3. Compute $h = H(M||r)$.
  4. Compute $u_1 = hw \ mod \ n$ and $u_2 = rw \ mod \ n$.
  5. Compute $R = u_1P + u_2Q = (x_R, y_R)$ and $v = x_R \ mod \ n$.
  6. The signature is verified only if $v = r$.

Flaws in ECDSA-II. The only difference between ECDSA-II and ECDSA is the hash function evaluation. Therefore, ECDSA-II can not resist the two

flaws of ECDSA, because this difference does not effect to these two flaws.
The security proof for ECDSA-II. In [1], ECDSA-II is proved secure against the no message attack in the random oracle model by using The Improved Forking Lemma [3] and the property of the *Elliptic Curve Trusted El Gamal Type Signature Scheme* (*ECTEGTSS*) [1] – we will present these notions in section 4.2.

Theorem 1 *([1]). Suppose an adversary $\mathcal{A}$ against ECDSA-II exists which succeeds with probability $\varepsilon > 4/p$ after q queries to the random oracle H, then one can solve the discrete logarithm problem in $E(\mathbb{F}_p)$ using fewer than*

$$\frac{150q \log 12}{\varepsilon}$$

replays of $\mathcal{A}$ with probability greater than 1/100.

Note that, two flaws of ECDSA-II does not conflict with the security against the no-message attack of this scheme. The reason is that an attacker in the no-message attack scenario does not have other ability than knowledge of the public data.

4.1.2 ECDSA-III

ECDSA-III is identical to ECDSA-II, except that replace $f(R) = x_R \ mod \ n$ with $f(R) = x_R + y_R$. [1] shows that this alternation of ECDSA-III can resist two flaws of ECDSA in [2].

- Signing Algorithm (A signs on message M):
  1. Select $k \in_R [1, n-1]$.
  2. Compute $R = kP = (x_R, y_R)$, and $r = f(R) = x_R + y_R$, if $r = 0$ return to Step 1.
  3. Compute $h = H(M||r)$.
  4. Compute $s = k^{-1}(h + dr) \ mod \ n$; if $s = 0$, return to Step 1.
  5. The signature of A on M is (r, s).
- Verification Algorithm (the verified signature (r, s) on M of signer A):

1. Verifying whether r, s belong to $[1, n-1]$ or not.
2. Compute $w = s^{-1}\ mod\ n$.
3. Compute $h = H(M||r)$.
4. Compute $u_1 = hw\ mod\ n$ and $u_2 = rw\ mod\ n$.
5. Compute $R = u_1P + u_2Q = (x_R, y_R)$ and $v = x_R + y_R$.
6. The signature is verified only if $v = r$.

The security proof for ECDSA-III. Similar to ECDSA-II, ECDSA-III is also proved secure against the no-message attack in the random oracle model by using the Improved Forking Lemma [3] and the property of the ECTEGTSS [1].

Theorem 2 *([1]). Suppose an adversary $\mathcal{A}$ against ECDSA-III exists which succeeds with probability $\varepsilon > 4/p$ after q queries to the random oracle H, then one can solve the discrete logarithm problem in $E(\mathbb{F}_p)$ using fewer than*

$$\frac{100q \log 8}{\varepsilon} = \frac{300q}{\varepsilon}$$

replays of A with probability greater than 1/100.

4.2 Two variants of GOST R34.10-2012

In this section, we will present two variants of GOST R 34.10-2012, called GOST-I and GOST-II. We also assume that the parameters p and n for these variants satisfy:

- If $2^{254} < n < 2^{256}$ then $p < 2^{256}$.
- If $2^{508} < n < 2^{512}$ then $P < 2^{512}$.

4.2.1 GOST-I

In a similar way to gain ECDSA-II, we obtain GOST-I by replacing the hash function evaluation $h = H_{GOST}(M)$ in GOST R 34.10-2012 by $h = H_{GOST}(M||r)$. This variant is described as follows.

- Signing Algorithm (A signs on message M):

  1. Generate a random (pseudorandom) integer k, satisfying the inequality: $0 < k < n$.
  2. Calculate $R = kP = (x_R, y_R)$, and $r = f(R) = x_R \ mod \ n$, if $r = 0$ return to Step 1.
  3. Calculate the message hash code M: $h = H_{GOST}(M||r)$.
  4. Calculate an integer α, the binary representation of which is the vector h, and determine $e = \alpha(mod \ n)$. If $e = 0$, then assign $e = 1$.
  5. Calculate $s = rd + ke \ mod \ n$; if $s = 0$, return to Step 1.
  6. The signature of A on M is (r, s).

- Verification Algorithm (the verified signature (r, s) on M of signer A):

  1. Verifying whether r, s belong to $[1, n-1]$ or not.
  2. Compute $h = H_{GOST}(M||r)$.
  3. Calculate an integer Oç, the binary representation of which is the vector h, and determine $e = \alpha(mod \ n)$. If $e = 0$, then assign $e = 1$.
  4. Compute $w = e^{-1} mod \ n$.
  5. Compute $u_1 = sw \ mod \ n$ and $u_2 = -rw \ mod \ n$.
  6. Compute $R = u_1P + u_2Q = (x_R, y_R)$ and $v = x_R \ mod \ n$.
  7. The signature is verified only if $v = r$.

The security proof for GOST-I. Here, we consider the security of GOST-I against the no-message attack in the random oracle model by applying the method provided in [1]. In order to consider the security of GOST-I, we will recall the definition of ECTEGTSS in [1] and the Improved Forking Lemma in [3].

According to [1], a signature scheme is an ECTEGTSS if it has the following properties:

i. The underlying group is from an elliptic curve E over a finite field $\mathbb{F}_p$ whose order is equal to a prime n times a small cofactor c, i.e. $|E(\mathbb{F}_p)| = c \cdot n$. A base point $P \in E(\mathbb{F}_p)$ of order n is given.

ii. It uses two function G and H, with ranges $\mathcal{G}$ and $\mathcal{H}$ respectively. For security analysis, the functions H is modelled as a random oracle and G requires some practical property such as (multi)-collision-resistance or (multi)-collision-freeness.

iii. There are three functions:

$$\begin{aligned} F_1 : &\ \mathbb{Z}_n \times \mathbb{Z}_n \times \mathcal{G} \times \mathcal{H} \to \mathbb{Z}_n, \\ F_2 : &\ \mathbb{Z}_n \times \mathcal{G} \times \mathcal{H} \to \mathbb{Z}_n, \\ F_3 : &\ \mathbb{Z}_n \times \mathcal{G} \times \mathcal{H} \to \mathbb{Z}_n \end{aligned}$$

satistfying for all $(k, d, r, h) \in (\mathbb{Z}_n, \mathbb{Z}_n, \mathcal{G}, \mathcal{H})$,

$$F_2(F_1(k, d, r, h), r, h) + dF_3(F_1(k, d, r, h), r, h) = k \ mod \ n.$$

iv. Each user has private and public keys d, Q such that $Q = dP$.

v. To sign a message m, the signer Alice picks $k \in \mathbb{Z}_n^*$, computes $R = kP$ and $r = G(R)$. She then gets $h = H(m||r)$ and computes $s = F_1(k, x, r, h)$. The signature on m is (s, r, h), although (s, r) is enough in practice since h may be recovered from m and r.

vi. To verify the signature (s, r, h) on a message m the verifier Bob computes $e_P = F_2(s, r, h)$, $e_Q = F_3(s, r, h)$ and finally $W = e_P P + e_Q Q$. He then checks that $r = G(W)$ and $h = H(m||r)$.

vii. The functions F_2 and F_3 must satisfy the following one-to-one condition: for given r, e_P and e_Q, there exists a unique pair (h, s) such that

$$e_P = F_2(s, r, h) \text{ and } e_Q = F_3(s, r, h).$$

Furthermore, this pair is easy to find.

Note that, if a signature scheme is an ECTEGTSS is also, it is also a TEGTSS-II [3]. Therefore, we can apply the Improved Forking Lemma in [3] for this scheme.

Lemma 3 ***(The Improved Forking Lemma, [3])*** *Let us consider a probabilistic polynomial time Turing machine $\mathcal{A}$, called the attacker, and a probabilistic polynomial time simulator $\mathcal{B}$. If $\mathcal{A}$ can find with probability $\varepsilon > 4/p$ a verifiable tuple (M, R, S, T, U) with less than q queries to the hash function, for a new message M and for a U directly defined by H, then with a constant probability 1/96, with $(1 + 24q\ell \log(2\ell))/\varepsilon$ replays of $\mathcal{A}$ and $\mathcal{B}$ with different random oracles, $\mathcal{A}$ will output $\ell+1$ verifiable tuples $(M_i, R_i, S_i, T_i, U_i)_{i=1,..,\ell+1}$ such that the U_i are pairwise distinct, and all the R_i equal for TEGTSS-I schemes but all the (M_i, T_i) equal for TEGTSS-II schemes.*

The following result shows that GOST-I is an ECTEGTSS.

Lemma 4 *The GOST-I signature scheme is an ECTEGTSS.*

Proof. We will prove that GOST-I satisfies all properties of an ECTEGTSS.

i. The underlying group of GOST-I is from the elliptic curve $E(\mathbb{F}_p)$ whose order is equal to a prime n times a cofactor c, and the base point is $P \in E(\mathbb{F}_p)$ of order n. As our assumption, the parameters p and n for GOST satisfy: the first case, $p < 2^{256}$ and $2^{254} < n < 2^{256}$; the second case, $p < 2^{512}$ and $2^{508} < n < 2^{512}$. According to the Hasse Theorem, the number of points on $E(\mathbb{F}_p)$ satisfies $p-2\sqrt{p}+1 < |E(\mathbb{F}_p)| < p+2\sqrt{p}+1$. Therefore, if $|E(\mathbb{F}_p)| = c \cdot n$, then $c \leqslant 16$.

ii. GOST-I uses the hash function H_{GOST} and the function $f(R) = x_R$ (where $R \in E(\mathbb{F}_p)$). In the random oracle model, H_{GOST} is modelled as a random function. Our task is to prove that the function f is ℓ-collision-resistance or ℓ-collision- freeness. According to [1], if there exists $c + 1$ integer numbers $k_1, .., k_{c+1}$ satisfying $f(k_1P) = f(k_2P) = .. = f(k_{c+1}P)$, then there must exist $i, j \in \{1, .., c+1\}$ with $i \neq j$ such that $k_i = k_j$ or $k_i = -k_j$. It implies that f is $2c + 1$-collision-freeness. By using (i), we have $c \leqslant 16$, therefore f is 33-collision-freeness.

iii. GOST-I satisfies property (iii). Indeed, we consider

$$\begin{aligned} F_1(k, d, r, h) &= hk + dr, \\ F_2(s, r, h) &= sh^{-1}, \\ F_3(s, r, h) &= -rh^{-1}, \end{aligned}$$

compute

$$\begin{aligned}\mathbb{S} &= F_2(F_1(k,d,r,h),r,h) + d \cdot F_3(F_1(k,d,r,h),r,h) \\ &= F_1(k,d,r,h)h^{-1} - drh^{-1} \\ &= (hk+dr)h^{-1} - drh^{-1} \\ &= k.\end{aligned}$$

It is easy to see that GOST-I satisfies properties (iv) to (vii) by using its description and the definition of the functions F_1, F_2, F_3.

Therefore, we obtain that GOST-I is an ECTEGTSS.

Proposition 5 *Suppose an adversary $\mathcal{A}$ against GOST-I exists which succeeds with probability $\varepsilon > 4/p$ after q queries to the random oracle H, then one can solve the discrete logarithm problem in $E(\mathbb{F}_p)$ using*

$$\frac{1 + 768q\log 64}{\varepsilon} = \frac{1 + 4608q}{\varepsilon}$$

replays of $\mathcal{A}$ with probability greater than 1/100.

Proof. According to Lemma 4, GOST-I is an ECTEGTSS, therefore it is also a TEGTSS-II. Applying Lemma 3 for GOST-I with the parameters M (the message), $S = s$, $U = H_{GOST}(M, r) = h$ and $l = 32$, we obtain that after

$$\frac{1 + 768q\log 64}{\varepsilon} = \frac{1 + 4608q}{\varepsilon}$$

replays of $\mathcal{A}$ with a constant probability $1/96$ (greater than $1/100$), $\mathcal{A}$ will output 33 verifiable tuples $(M_i, R_i, S_i, T_i, U_i)_{i=1,..,33}$ such that the U_i are pairwise distinct, and $(M_i, T_i) = (M, r)$. These 33 tuples correspond to 33 elliptic points, $R_1 = k_1 P, .., R_{33} = k_{33} P$, and then we have $f(R_1) = .. = f(R_{33}) = r$. Because f is 33-collision-freeness (i.e. there at most 32 distinct points such that the values of the function f at these points are equal), there must exist two points $R_i, R_j \in \{R_1, .., R_{33}\}$ with $i \neq j$ such that $R_i = R_j$. Therefore, we have

$$k_1 = k_2 \ mod \ n,$$

or

$$h_2(s_1 - rd) = h_1(s_2 - rd) \ mod \ n,$$

therefore

$$d = r^{-1}(h_2 - s_1)^{-1}(h_2 s_1 - h_1 s_2)\ mod\ n.$$

Hence, we recovered the discrete logarithm of the public key Q. □

Remark 1. It is similar to the proof of Proposition 1, we obtain that GOST-I is able to resist two flaws of ECDSA in [2].

Remark 2. In the security proof of GOST-I, we can see that the reduction needs $\frac{1+4608q}{\varepsilon}$ replays of $\mathcal{A}$. However, in the security proof of ECDSA-II, the reduction only needs $\frac{150q\log 12}{\varepsilon}$ replays of $\mathcal{A}$. This does not imply that GOST-I is less secure than ECDSA-II. The reason is that the security of ECDSA is only reduced to the hardness of the discrete logarithm problem in $E(\mathbb{F}_p)$ with the size of p being 256 bit, but the security of GOST-I can be reduced to the hardness of the discrete logarithm problem in $E(\mathbb{F}_p)$ with the size of p being 512 bit.

4.2.2 GOST-II

GOST-II is identical to GOST-I, except that replace $f(R) = x_R \bmod n$ with $f(R) = x_R + y_R$. It is similar to GOST-I and GOST R 34.10-2012, this scheme is able to resist two flaws of ECDSA in [2].

- Signing Algorithm (A signs on message M):
  1. Generate a random (pseudorandom) integer k, satisfying the inequality: $0 < k < n$.
  2. Calculate $R = kP = (x_R, y_R)$, and $r = f(R) = x_R + y_R$, if $r = 0$ return to Step 1.
  3. Calculate the message hash code M: $h = H_{GOST}(M||r)$.
  4. Calculate an integer α, the binary representation of which is the vector h, and determine $e = \alpha(mod\ n)$. If $e = 0$, then assign $e = 1$.
  5. Calculate $s = rd + ke\ mod\ n$; if $s = 0$, return to Step 1.
  6. The signature of A on M is (r, s).
- Verification Algorithm (the verified signature (r, s) on M of signer A):
  1. Verifying whether r, s belong to $[1, n-1]$ or not.

2. Compute $h = H_{GOST}(M||r)$.
3. Calculate an integer Oç, the binary representation of which is the vector h, and determine $e = \alpha(mod\ n)$. If $e = 0$, then assign $e = 1$.
4. Compute $w = e^{-1} mod\ n$.
5. Compute $u_1 = sw\ mod\ n$ and $u_2 = -rw\ mod\ n$.
6. Compute $R = u_1P + u_2Q = (x_R, y_R)$ and $v = x_R + y_R\ mod\ n$.
7. The signature is verified only if $v = r$.

The security proof for GOST-II. Here, we consider the security of GOST-II against the no-message attack in the random oracle model.

Lemma 6 *The GOST-II signature scheme is an ECTEGTSS.*

Proof. The proof of this lemma is the same that of Lemma 4, except that the proof for property (ii) will be changed. Note that, the equation $x + y = t$ will intersect the curve $E(\mathbb{F}_p)$ in at most three points. Therefore, the function $f(R) = x_R + y_R$ is 4-collision-freeness. Hence, GOST-II satisfies property (ii), and then it is an ECTEGTSS. □

Proposition 7 *Suppose an adversary $\mathcal{A}$ against GOST-II exists which succeeds with probability $\varepsilon > 4/p$ after q queries to the random oracle H, then one can solve the discrete logarithm problem in $E(\mathbb{F}_p)$ using*

$$\frac{1 + 72q \log 6}{\varepsilon}$$

replays of $\mathcal{A}$ with probability greater than 1/100.

Proof. It is the same proof of Proposition 5, we apply Lemma 3, Lemma 6 with the parameters M (message), $S = s$, $T = r$, $U = H_{GOST}(m, r) = h$, and $l = 3$. □

Remark 3. We note that the above security results of GOST-I and GOST-II still holds if we use the assumption that c is a small number (such as $c \leq 16$) instead of the assumption in the start of Section 4.2.

5 Our Future Research

In this paper, we provided two security comparisons between GOST R 34.10-2012 and ECDSA. The first, we compared GOST R 34.10-2012 with ECDSA via two flaws of ECDSA analyzed in [2], although these two flaws do not actually affect too much the security of ECDSA. In particular, we obtain that GOST R 34.10-2012 is able to resist these two flaws of ECDSA. Second, we presented another comparison between GOST R 34.10-2012 and ECDSA by applying the method of J. Malone-Lee and N.P. Smart in [1] for GOST R 34.10-2012. As a consequence, we obtain two variants GOST-I and GOST-II of GOST R 34.10-2012 that are secure against the no-message attack in the random oracle model. This result is similar to the result for the variants ECDSA-II and ECDSA-III of ECDSA in [1]. And, our comparison results may be summarized in the following table.

Feature	ECDSA-II	ECDSA-III	GOST-I	GOST-II	ECDSA	GOST R 34.10-2012
Resistance to two flaws of ECDSA	No	Yes	Yes	Yes	No	Yes
The security proof in the random oracle	Secure against the no-message attack	Secure against the no-message attack	Secure against the no-message attack	Secure against the no-message attack	No proof	No proof

In our future works, we want to consider the security of GOST-I, GOST-II, ECDSA-II, ECDSA-III against the adaptively chosen message attack in the random oracle model. Besides, another our concern is whether GOST R 34.10-2012 is secure against the adaptively chosen message attack in the generic group model [6].

Moreover, N. P. Varnovskii [4] presented a modification for GOST signature schemes (GOST R 34.10-94, GOST R 34.10-2001, and GOST R 34.10-2012) which may be proved secure in *the tamper-proof device model*. Hence, on the topic of "Comparisons of GOST R34.10-2012 and ECDSA Schemes", we will study this modification on ECDSA.

References

[1] J. Malone-Lee and N. P. Smart, "Modifications of ECDSA", *International Workshop on Selected Areas in Cryptography Selected Areas in Cryptography*, 2002, pages 1–12.

[2] J. Stern, D. Pointcheval, J. Malone-Lee, Nigel P. Smart, "Flaws in Applying Proof Methodologies to Signature Schemes", *Annual International Cryptology Conference, CRYPTO 2002: Advances in Cryptology – CRYPTO 2002*, pages 93–110.

[3] E. Brickell, D. Pointcheval, S. Vaudenay, M. Yung, "Design Validations for Discrete Logarithm Based Signature Schemes", *PKC 2000: Public Key Cryptography*, pages 276–292.

[4] N. P. Varnovskii, "Provable security of digital signatures in the tamper-proof device model", *Diskr. Mat., 2008, Volume 20, Issue 3*, pages 147-159.

[5] V. Dolmatov, A. Degtyarev, "GOST R 34.10-2012: Digital Signature Algorithm", *Independent Submission, Request for Comments: 7091, Updates: 5832, Category: Informational*, ISSN: 2070-1721.

[6] D.R.L. Brown, "Generic Groups, Collision Resistance, and ECDSA", *Designs, Codes and Cryptography, April 2005, Volume 35, Issue 1*, pages 119-152.

[7] Kerry, F. Cameron, "Digital Signature Standard (DSS)", *Federal Information Processing Standards Publication*, 2013, FIPS PUB 186-4.

[8] ISO/IEC 14888-3:2016 Information technology – Security techniques – Digital signatures with appendix – Part 3: Discrete logarithm based mechanisms.

[9] S.Blacke-Wilson, A. Menezes, "Unknown key-share attacks on the station-to- station (STS) protocol", *Public Key Cryptography – PKC 1999, LNCS 1560*, 1999, pp. 156–170.

Approximate Common Divisor Problem and Lattice Sieving

Kirill Zhukov

Abstract

In this paper we describe a heuristic algorithm for computing a common divisor of two integers, one of these integers being known approximately. We reduce this computational problem to solving a system of integer linear inequalities. We solve this system with two unknowns using a method suggested by J. Franke and T. Kleinjung for lattice sieving. There are cases in which our algorithm is applicable and the best algorithm based on Coppersmith's method is not applicable.

Keywords: approximate common divisor problem, lattice sieving, system of integer linear inequalities, integer linear programming, Gaussian Volume Heuristics

1 Introduction

Let us describe a partially approximate common divisor problem (PACDP). Consider two integers N_1 and N_2. Assume that for some unknown integer Δ integers N_1 and $N_2 - \Delta$ have common divisor $A > |\Delta|$. The goal is to find A.

The PACDP was introduced in 2001 by N. Howgrave-Graham [2] who used the continued fraction techniques and Coppersmith's method.

An algorithm of S. Sarkar and S. Maitra [3] for solving the PACDP is known to be the best. This algorithm is based on Coppersmith's method and finds a common divisor in time which is polynomial of $n = \max\{[\ln N_1], [\ln N_2]\}$ provided that $l_A > l_{N_1/A} + l_\Delta - \frac{l^2_{N_1/A}}{l_{N_1}}$, where l_A, $l_{N_1/A}$, l_{N_1} and l_Δ are the binary lengths of A, $\frac{N_1}{A}$, N_1 and Δ respectively. In paper [4] a method for computing a common divisor with continued fractions is proposed. The

time complexity of that method is polynomial of n and c provided that $l_A \geq l_{N_1/A} + l_\Delta + 2 - \log_2 c$. Here we introduce a new algorithm, which has the same restrictions but is faster in $\ln c$ times.

2 PACDP and a system of integer linear inequalities

Suppose that positive integers N_1 and $N_2 - \Delta$ have a nontrivial common divisor A. Then the integers N_1 and N_2 have representations $N_1 = AB_1$ and $N_2 = AB_2 + \Delta$.

Suppose that we want to find a divisor $A > D$, where D is some known bound, in the PACDP with the inputs N_1 and N_2. It is easy to see that if

$$A \geq D > \sqrt{\frac{N_1 \left|\Delta\right|}{c}} \tag{1}$$

for some known positive real c, then:

$$\begin{gathered} 0 < B_1 < \frac{N_1}{D} \\ -cD < N_2 B_1 - N_1 B_2 < cD \end{gathered} \tag{2}$$

Therefore, the vector (B_2, B_1) is a solution of a system of integer linear inequalities

$$\mathbf{v_1} \leq \mathbf{xA} \leq \mathbf{v_2}, \tag{3}$$

where $\mathbf{v_1} = (-cD, 0)$, $\mathbf{v_2} = (cD, \frac{N_1}{D})$, $\mathbf{A} = \left(\begin{smallmatrix} -N_1 & 0 \\ N_2 & 1 \end{smallmatrix}\right)$ and $\mathbf{x} = (x, y)$ is a vector of unknown variables. The solution of system (3) in terms of geometry of numbers is equivalent to finding all points of a lattice

$$\mathcal{L} = \{(-N_1, 0)x + (N_2, 1)y \mid x, y \in \mathbb{Z}\},$$

inside a rectangle

$$\mathcal{R} = \left\{(a, b) \in \mathbb{R}^2 \mid -cD \leq a \leq cD, 0 \leq b \leq \frac{N_1}{D}\right\}.$$

Each point $(a, b) \in \mathcal{L} \cap \mathcal{R}$ corresponds to a solution of (3) with $x = \frac{N_2 b - a}{N_1}$, $y = b$.

In the next section we describe an algorithm of J. Franke and T. Kleinjung that finds all points of two dimensional lattice inside a rectangle. Using this algorithm one could easily formulate a practical procedure for solving PACDP provided the number of solutions of system (3) is small.

Using the Gaussian volume heuristic we could estimate the number of solutions of system (3) in $\frac{\mathrm{Vol}(\mathcal{R})}{\det(\mathcal{L})}$, where $\mathrm{Vol}(\mathcal{R}) = 2cN_1$ is the volume of rectangle $\mathcal{R}$ and $\det(\mathcal{L}) = N_1$ is the determinant of the lattice $\mathcal{L}$. Further we will use the following assumption.

Assumption 1. *The rectangle $\mathcal{R}$ contains no more than $2c$ points of the lattice $\mathcal{L}$.*

Notice that for some special lattices and rectangles we can not use the Gaussian volume heuristic to estimate the cardinality of their intersection. But in our experiments all the random PACDP fulfill heuristic assumption 1.

3 Algorithm of J. Franke and T. Kleinjung

For $k \in \mathbb{N}$ and vector $\mathbf{v} \in \mathfrak{R}^k$, where $\mathfrak{R}$ — a ring, we denote the i-th coordinate as $v^{(i)}$, $i \in \overline{1,k}$ (therefore $\mathbf{v} = (v^{(1)}, v^{(2)}, \ldots, v^{(k)})$). For matrix $\mathbf{A} = \begin{pmatrix} \mathbf{a_1} \\ \mathbf{a_2} \end{pmatrix} \in \mathbb{Z}_{2,2}$, vectors $\mathbf{v_1}, \mathbf{v_2} \in \mathbb{Z}^2$ and integers $I_1, I_2 \in \mathbb{Z}$ we define the following subsets of $\mathbb{R}^2$.

$$\mathcal{L}(\mathbf{A}) = \{\mathbf{a_1}x + \mathbf{a_2}y \mid x, y \in \mathbb{Z}\},$$

$$\mathcal{R}(\mathbf{v_1}, \mathbf{v_2}) = \left\{\mathbf{v} \in \mathbb{R}^2 : \mathbf{v_1}^{(i)} < \mathbf{v}^{(i)} < \mathbf{v_2}^{(i)} \mid i = 1, 2\right\},$$

$$\mathcal{S}(I_1, I_2) = \left\{\mathbf{v} \in \mathbb{R}^2 \mid I_1 < \mathbf{v}^{(1)} < I_2\right\},$$

$$\mathcal{LS}(\mathbf{A}, I_1, I_2) = \mathcal{L}(\mathbf{A}) \cap \mathcal{S}(I_1, I_2).$$

Definition 1. *We say that matrix $\mathbf{B} = \begin{pmatrix} \mathbf{b_1} \\ \mathbf{b_2} \end{pmatrix} \in \mathbb{Z}_{2,2}$ is FK-reduced with parameter $I \in \mathbb{Z}_{>0}$ if:*

1. $-I < \mathbf{b_1}^{(1)} \leq 0 \quad$ *and* $\quad 0 \leq \mathbf{b_2}^{(1)} < I$,

2. $\mathbf{b_1}^{(2)} > 0 \quad$ *and* $\quad \mathbf{b_2}^{(2)} > 0$,

3. $\mathbf{b_2}^{(1)} - \mathbf{b_1}^{(1)} > I$

We denote the set of FK-reduced with parameter I matrices as $\mathcal{FK}_I(\mathbb{Z}_{2,2})$.

Proposition 1. *If matrix $\mathbf{A} = \left(\begin{smallmatrix}\mathbf{a_1}\\ \mathbf{a_2}\end{smallmatrix}\right) \in \mathbb{Z}_{2,2}$ satisfies the restrictions*

1. $\mathbf{a_1}^{(1)} < 0 \qquad \mathbf{a_2}^{(1)} > 0$,

2. $\mathbf{a_1}^{(2)} \geq 0 \qquad \mathbf{a_2}^{(2)} > 0$,

3. $\mathbf{a_1}^{(1)} + \mathbf{a_1}^{(1)} < 0$.

then there exists the unique decomposition $\mathbf{A} = \mathbf{UB}$ such that $\mathbf{U} \in \mathbb{Z}_{2,2}$ is invertible and $\mathbf{B} \in \mathcal{FK}_I(\mathbb{Z}_{2,2})$.

Proposition 2. *If $\mathbf{B} = \left(\begin{smallmatrix}\mathbf{b_1}\\ \mathbf{b_2}\end{smallmatrix}\right) \in \mathcal{FK}_I(\mathbb{Z}_{2,2})$, then for any $\mathbf{b} \in \mathcal{L}(\mathbf{B})$, such that $-I < \mathbf{b}^{(1)} < I$ and $\mathbf{b}^{(2)} > 0$, representation $\mathbf{b} = x\mathbf{b_1} + y\mathbf{b_2}$ has non-negative coefficients $x \geq 0$, $y \geq 0$.*

The proofs of propositions 1 and 2 are in [1]. The next algorithm is reformulation of a reduction procedure from [1].

Algorithm 1 (FK reduction)

Require: $I \in \mathbb{N}$, $\mathbf{A} = \left(\begin{smallmatrix}\mathbf{a_1}\\ \mathbf{a_2}\end{smallmatrix}\right)$: $\mathbf{a_1}^{(1)} < 0$, $\mathbf{a_2}^{(1)} > 0$, $\mathbf{a_1}^{(2)} \geq 0$, $\mathbf{a_2}^{(2)} > 0$, $\mathbf{a_1}^{(1)} + \mathbf{a_1}^{(1)} < 0$.
Ensure: $\mathbf{B} \in \mathcal{FK}_I(\mathbb{Z}_{2,2}) : \mathcal{L}(\mathbf{A}) = \mathcal{L}(\mathbf{B})$

```
procedure REDUCE(A, I)
    b_1 ← a_1, b_2 ← a_2
    while 1 do
        b_1 ← b_1 + ⌊|b_1^(1) / b_2^(1)|⌋ · b_2
        if b_1^(1) > −I then
            b_2 ← b_2 + ⌈(I − b_2^(1)) / b_1^(1)⌉ · b_1
            return (b_1 / b_2)
        b_2 ← b_2 + ⌊|b_2^(1) / b_1^(1)|⌋ · b_1
        if b_2^(1) < I then
            b_1 ← b_1 + ⌈(b_1^(1) − I) / b_2^(1)⌉ · b_2
            return (b_1 / b_2)
```

If a lattice basis is FK-reduced with parameter I, then we can successively enumerate all the lattice points inside a strip of a width I. The proof of correctness of the next algorithm is given in [1].

Algorithm 2 (Next lattice point in a strip)

Require: $I_1, I_2 \in \mathbb{Z} : I_1 < I_2$,
$\mathbf{B} \in \mathcal{FK}_{I_2 - I_1}(\mathbb{Z}_{2,2})$,
$\mathbf{a} \in \mathcal{LS}(\mathbf{B}, I_1, I_2)$
Ensure: $\mathbf{b} \in \mathcal{LS}(\mathbf{B}, I_1, I_2) : b^{(2)} = \min\{c^{(2)} \mid \mathbf{c} \in \mathcal{LS}(\mathbf{B}, I_1, I_2), c^{(2)} > a^{(2)}\}$

1: **procedure** StripNextPoint($\mathbf{B}, \mathbf{a}, I_1, I_2$)
2: **if** $a^{(1)} \geq I_1 - {b_1}^{(1)}$ **then**
3: $\mathbf{b} \leftarrow \mathbf{a} + \mathbf{b_1}$
4: **else if** $a^{(1)} < I_2 - {b_2}^{(1)}$ **then**
5: $\mathbf{b} \leftarrow \mathbf{a} + \mathbf{b_2}$
6: **else**
7: $\mathbf{b} \leftarrow \mathbf{a} + \mathbf{b_1} + \mathbf{b_2}$
8: **return b**

Algorithm 3 (Next lattice point in a strip with step D)

Require: $I_1, I_2 \in \mathbb{Z} : I_1 < I_2$,
$\mathbf{B} \in \mathcal{FK}_{I_2-I_1}(\mathbb{Z}_{2,2})$,
$\mathbf{a} \in \mathcal{LS}(\mathbf{B}, I_1, I_2)$,
$D \in \mathbb{Z}$

Ensure: $\mathbf{b} \in \mathcal{LS}(\mathbf{B}, I_1, I_2) : b^{(2)} = \min\{c^{(2)} \mid \mathbf{c} \in \mathcal{LS}(\mathbf{B}, I_1, I_2), c^{(2)} > a^{(2)} + D\}$

1: **procedure** StripJumpPoint($\mathbf{B}, \mathbf{a}, I_1, I_2, D$)
2: $J \leftarrow a^{(2)} + D$
3: $s \leftarrow \lfloor -{b_2}^{(1)} D / \det(\mathbf{B}) \rfloor$, $t \leftarrow \lfloor {b_1}^{(1)} D / \det(\mathbf{B}) \rfloor$, $\mathbf{b} \leftarrow \mathbf{a} + s\mathbf{b_1} + t\mathbf{b_2}$
4: **if** $b^{(1)} \geq I_1 - {b_1}^{(1)}$ **then**
5: $\mathbf{b} \leftarrow \mathbf{b} + \mathbf{b_1}$
6: **if** $b^{(2)} > J$ **then**
7: $l \leftarrow \min\left\{\lfloor (b^{(1)} - I_1)/{b_2}^{(1)} \rfloor, \lfloor (b^{(2)} - J)/{b_2}^{(2)} \rfloor\right\}$, $\mathbf{b} \leftarrow \mathbf{b} - l\mathbf{b_2}$
8: **else**
9: $l \leftarrow \min\left\{\lfloor (I_1 - b^{(1)})/{b_1}^{(1)} \rfloor, \lfloor (J - b^{(2)})/{b_1}^{(2)} \rfloor\right\}$, $\mathbf{b} \leftarrow \mathbf{b} + l\mathbf{b_1}$
10: $\mathbf{b} \leftarrow$ StripNextPoint($\mathbf{B}$, $\mathbf{b}$, I_1, I_2)
11: **else if** $b^{(1)} < I_2 - {b_2}^{(1)}$ **then**
12: $\mathbf{b} \leftarrow \mathbf{b} + \mathbf{b_2}$
13: **if** $b^{(2)} > J$ **then**
14: $l \leftarrow \min\left\{\lfloor (b^{(1)} - I_2)/{b_1}^{(1)} \rfloor, \lfloor (b^{(2)} - J)/{b_1}^{(2)} \rfloor\right\}$, $\mathbf{b} \leftarrow \mathbf{b} - l\mathbf{b_1}$
15: **else**
16: $l \leftarrow \min\left\{\lfloor (I_2 - b^{(1)})/{b_2}^{(1)} \rfloor, \lfloor (J - b^{(2)})/{b_2}^{(2)} \rfloor\right\}$, $\mathbf{b} \leftarrow \mathbf{b} + l\mathbf{b_2}$
17: $\mathbf{b} \leftarrow$ StripNextPoint($\mathbf{B}$, $\mathbf{b}$, I_1, I_2)
18: **else**
19: $\mathbf{b} \leftarrow \mathbf{b} + \mathbf{b_1} + \mathbf{b_2}$
20: **return b**

Remark 1. *In the original description [1] of procedure* StripJumpPoint *steps 7 and 14 are omitted. In that case the procedure may not work. We prove the correctness of our version of procedure in the appendix.*

Algorithm 4 (Lattice points inside a rectangle)

Require: $\mathbf{A} = \left(\begin{smallmatrix}\mathbf{a_1}\\ \mathbf{a_2}\end{smallmatrix}\right)$: $\mathbf{a_1}^{(1)} < 0$, $\mathbf{a_2}^{(1)} > 0$, $\mathbf{a_1}^{(2)} \geq 0$, $\mathbf{a_2}^{(2)} > 0$, $\mathbf{a_1}^{(1)} + \mathbf{a_1}^{(1)} < 0$
$\mathbf{v_1}, \mathbf{v_2} \in \mathbb{Z}^2 : \mathcal{R}(\mathbf{v_1}, \mathbf{v_2}) \neq \varnothing$
Ensure: $\mathcal{L}(\mathbf{A}) \cap \mathcal{R}(\mathbf{v_1}, \mathbf{v_2})$

1: **procedure** RectAllPoints($\mathbf{A}$, $\mathbf{v_1}$, $\mathbf{v_2}$)
2: $\quad\mathcal{T} \leftarrow \varnothing$
3: $\quad\mathbf{B} \leftarrow$ Reduce($\mathbf{A}$, $v_2{}^{(1)} - v_1{}^{(1)}$)
4: $\quad\mathbf{b} \leftarrow \lfloor v_2^{(1)}/b_2^{(1)} \rfloor \cdot \mathbf{b_2}$
5: $\quad\mathbf{b} \leftarrow$ StripJumpPoint($\mathbf{B}$, $\mathbf{b}$, $v_1^{(1)}$, $v_2^{(1)}$, $v_1^{(2)}$)
6: $\quad$**while** $b^{(2)} < v_2^{(2)}$ **do**
7: $\quad\quad\mathcal{T} \leftarrow \mathcal{T} \cup \{\mathbf{b}\}$
8: $\quad\quad\mathbf{b} \leftarrow$ StripNextPoint($\mathbf{B}$, $\mathbf{b}$, $v_1^{(1)}$, $v_2^{(1)}$)
9: $\quad$**return** $\mathcal{T}$

It easy to see that the complexity of Algorithm 4 is the sum of complexities of Reduce procedure and StripJumpPoint procedure, and also the complexity of StripNextPoint procedure multiplied by the number of iterations of loop 6–8.

4 Algorithm for PACDP

Suppose that the naturals N_1 and N_2 are the inputs of PACDP and $N_1 > N_2$. As mentioned in [2] the restriction $N_1 > N_2$ is not a limitation at all. If $N_1 < N_2$ we solve PACDP with the inputs N_1 and $N_2 - N_1 \cdot \lfloor N_2/N_1 \rfloor$.

If $N_1 > N_2$, then the matrix $\left(\begin{smallmatrix} N_2 & 1 \\ -N_1 & 0 \end{smallmatrix}\right)$ satisfies the restrictions from Proposition 1 hence we can apply algorithm 4 to find all solutions of system of integer inequalities (3). Then we have to check each solution of (3) for being a solution for PACDP. In practical implementation we would like to check each solution individually, rather than to collect all the solutions in memory. This leads to the following algorithm for solving PACDP.

Algorithm 5 (PACDP)

Require: N_1 and N_2 — natural numbers, $N_1 > N_2$; c, D — method's real parameters
Ensure: $\mathcal{W} \subset \mathbb{N} \times \mathbb{Z}$ — a set of pairs (A, Δ), such that $N_1 = AB_1$, $N_2 = AB_2 + \Delta$ and $A \geq D > \sqrt{\frac{N_1|\Delta|}{c}}$

1: **procedure** SolvePACDP(N_1, N_2, c, D)
2: $\quad$ $\mathbf{A} \leftarrow \left(\begin{smallmatrix} N_2 & 1 \\ -N_1 & 0 \end{smallmatrix}\right)$, $\mathbf{v_1} \leftarrow (cD, 0)$, $\mathbf{v_2} \leftarrow (cD, \frac{N_1}{D})$
3: $\quad$ $\mathcal{W} \leftarrow \varnothing$
4: $\quad$ $\mathbf{B} \leftarrow$ Reduce($\mathbf{A}$, ${v_2}^{(1)} - {v_1}^{(1)}$)
5: $\quad$ $\mathbf{b} \leftarrow \lfloor v_2^{(1)} / b_2^{(1)} \rfloor \cdot \mathbf{b_2}$
6: $\quad$ $\mathbf{b} \leftarrow$ StripJumpPoint($\mathbf{B}$, $\mathbf{b}$, $v_1^{(1)}$, $v_2^{(1)}$, $v_1^{(2)}$)
7: $\quad$ **for** $i = 1, 2, \ldots, 2c$ **do**
8: $\quad\quad$ **if** $b^{(2)} | N_1$ **then**
9: $\quad\quad\quad$ $B_1 \leftarrow b^{(2)}$, $B_2 \leftarrow \frac{N_2 B_1 - b^{(1)}}{N_1}$, $A \leftarrow \frac{N_1}{B_1}$, $\Delta \leftarrow N_2 - AB_2$
10: $\quad\quad\quad$ $\mathcal{W} \leftarrow \mathcal{W} \cup \{(A, \Delta)\}$
11: $\quad\quad$ $\mathbf{b} \leftarrow$ StripNextPoint($\mathbf{B}$, $\mathbf{b}$, $v_1^{(1)}$, $v_2^{(1)}$)
12: $\quad$ **return** $\mathcal{W}$

Algorithm 5 is correct for inputs such that Assumption 1 holds. For such inputs Algorithm 5 finds all the pairs (A, Δ) satisfying condition (1). Let $n = \max\{[\ln N_1],\ [\ln N_2]\}$. The complexity of step 4 is $O(n^2)$ binary operations (the Euclidean algorithm). The complexity of loop 7–11 is $O(cn \ln n \ln\ln n)$ binary operations. Therefore the total complexity is estimated as follows.

Proposition 3. *If $n = O(c)$ then the complexity of Algorithm 5 is*

$$O(cn \ln n \ln\ln n)$$

binary operations, where $n = \max\{[\ln N_1],\ [\ln N_2]\}$.

5 Experements

We have implemented Algorithm 5 using the MPIR library [5] for bignum arithmetic. We used the Microsoft Visual C++ compiler (64-bit). We ran our program on a single core of an Intel Xeon processor (3.5 GHz).

In Table 1 l_A denotes the length of the binary representation of A, l_Δ denotes the length of Δ, l_B denotes the lengths of B_1 and B_2, and the parameter c satisfies condition the $l_A \geq l_B + l_\Delta + 2 - \log_2 c$.

Table 1: Program implementation of Algorithm 5

l_A	l_B	$\log_2 c$	l_Δ	Time, sec
2795	277	27	2543	167
		30	2546	1340
3819	277	21	3561	3.5
		30	3570	1689

The values from the first and the third lines of Table 1 are on the bound of applicability of S. Sarkar's and S. Maitra's method. The values from the second and the forth lines are out of the range of S. Sarkar's and S. Maitra's method applicability. In each case the computing a common divisor with our implementation took the time showed in the last column of Table 1. In our experiments we've gained a speed up at least by a factor of 5 as compared with the experiments of [4].

References

[1] J. Franke and T. Kleinjung. *Continued Fractions and Lattice Sieving.* In Proceedings SHARCS 2005. `http://www.hyperelliptic.org/tanja/SHARCS/talks/FrankeKleinjung.pdf`

[2] N. Howgrave-Graham. *Approximate integer common divisors.* In Proceedings of CaLC'01, Lecture Notes in Computer Science. **2146** (2001), 51 – 66.

[3] S. Sarkar, S. Maitra. *Approximate integer common divisor problem relates to implicit factorization.* IEEE Trans. Inform. Theory. **57** (2011), 4002 – 4013.

[4] K. D. Zhukov. *Approximate common divisor problem and continued fractions.* Mathematical Aspects of Cryptography **7**:2 (2016), 61–70

[5] The Multiple Precision Integers and Rationals Library. `http://www.mpir.org/`

A Correctness of Algorithm 3

Proof. Denote $(\mu, \nu) = (0, D)\mathbf{A}^{-1}$. After step 3 the equalities $s = \lfloor \mu \rfloor$, $t = \lfloor \nu \rfloor$ hold. Hence, after step 3 the estimates $I_1 - {b_2}^{(1)} \leq b^{(1)} < I_2 - {b_1}^{(1)}$ and $b^{(2)} < J$ hold.

Suppose $b^{(1)} \geq I_1 - {b_1}^{(1)}$. We are going to show that the algorithm is correct in this case (the correctness in the case $b^{(1)} < I_2 - {b_2}^{(1)}$ can be proved in the same way). After step 5 the following inclusion holds $\mathbf{b} \in \mathcal{L}(\mathbf{B}) \cap \mathcal{S}(I_1, I_2)$. Consider the cases $b^{(2)} > J$ and $b^{(2)} \leq J$ separately.

If after step 5 the inequality $b^{(2)} > J$ holds, then for

$$l = \min \left\{ \lfloor (b^{(1)} - I_1)/{b_2}^{(1)} \rfloor, \lfloor (b^{(2)} - J)/{b_2}^{(2)} \rfloor \right\}$$

the vector $\mathbf{b} - l\mathbf{b_2}$ has the second coordinate minimal with property $b^{(2)} - lb_2^{(2)} \leq J$. To see this consider the vector $\mathbf{b}'$ which has the second coordinate maximal with property $b'^{(2)} < b^{(2)} - lb_2^{(2)}$. According to Algorithm 2 the vector $\mathbf{b}'$ is the difference of $\mathbf{b} - l\mathbf{b_2}$ and either $\mathbf{b_1}$ either $\mathbf{b_2}$ or $\mathbf{b_1} + \mathbf{b_2}$. If $\mathbf{b}' = \mathbf{b} - (l+1)\mathbf{b_2}$, then $b^{(2)} \leq J$ by the definition of l. Else $\mathbf{b}' = \mathbf{a} + s\mathbf{b_1} + (t - l')\mathbf{b_2}$ ($l' = l$ or $l' = l + 1$), hence $b^{(2)} \leq J$.

If after step 5 the inequality $b^{(2)} \leq J$ holds, then for

$$l \leftarrow \min \left\{ \lfloor (I_1 - b^{(1)})/{b_1}^{(1)} \rfloor, \lfloor (J - b^{(2)})/{b_1}^{(2)} \rfloor \right\}$$

the vector $\mathbf{b} + l\mathbf{b_1}$ has the second coordinate maximal with property $b^{(2)} + lb_1^{(1)} \leq J$. To see this consider the vector $\mathbf{b}''$ which has second coordinate minimal with property $b''^{(2)} > b^{(2)} + lb_1^{(2)}$. According to Algorithm 2 the vector $\mathbf{b}''$ is the sum of $\mathbf{b} + l\mathbf{b_1}$ and $\mathbf{b_1}$ or $\mathbf{b_2}$ (or both). If $\mathbf{b}'' = \mathbf{b} + (l+1)\mathbf{b_1}$, then $b^{(2)} > J$ by the definition of l. Else $\mathbf{b}'' = \mathbf{a} + (s + 1 + l')\mathbf{b_1} + (t+1)\mathbf{b_2}$, ($l' = l$ or $l' = l + 1$), hence $b''^{(2)} > J$. Notice that in this case algorithm outputs $\mathbf{b}''$.

Consider step 18. By the transition condition we have $b^{(1)} + b_1^{(1)} < I_1$. Since the matrix $\mathbf{B}$ is FK-reduced with parameter $(I_2 - I_1)$, then $b_1^{(1)} > I_1 - I_2$. Hence, $b^{(1)} < I_2$. In the same manner we get an estimate $b^{(1)} > I_1$. Therefore $b \in \mathcal{L}(\mathbf{B}) \cap \mathcal{S}(I_1, I_2)$. Using Algorithm 2 we calculate the next point as on step 19. After step 19 the estimate $b^{(2)} > a^{(2)} + D$ holds.

Estimates of extremal codeword weights of random linear codes over $\mathbf{F}_p$

Vasily Kruglov, Andrey Zubkov

Abstract

We propose two-sided estimates for the typical values of minimal and maximal non-zero codeword weights in random equiprobable linear code over $\mathbf{F}_p$.

Keywords: random linear codes, weight spectrum, minimal and maximal codeword weights

Let p be any fixed prime number. By $\mathbf{F}_p^N = \{X = (x_1, \ldots, x_N)\colon x_1, \ldots, x_N \in \mathbf{F}_p\}$ we denote the N-dimensional linear space over the prime field $\mathbf{F}_p$. Any k-dimensional subspace $L \subset \mathbf{F}_p^N$ we understood as k-dimensional linear code.

Weight of the vector $X = (x_1, \ldots, x_N) \in \mathbf{F}_p^N$ is defined as the number $w(X) = \sum_{k=1}^N I\{x_k \neq 0\}$ of its non-zero coordinates.

By $\left(\mathbf{F}_p^N\right)_s$ we denote the set of vectors of fixed weight s. For a linear code L we denote by $v_s(L) = |L \cap \left(\mathbf{F}_p^N\right)_s|$ the number of codewords in L having the weight s, the set of all $v_s(L)$ is called weight spectrum of code L. Then

$$v_{\leqslant s}(L) = \sum_{u=1}^{s} v_s(L) \quad \text{and} \quad v_{\geqslant s}(L) = \sum_{u=s}^{N} v_s(L)$$

are the numbers of non-zero codewords with the weight not exceeding s and not smaller than s correspondingly.

The following statement follows from the results in [8].

Theorem 1. *If $L \subset \mathbf{F}_p^N$ is a random linear k-dimensional code in $\mathbf{F}_p^N$ having the uniform distribution on the set of all k-dimensional codes in $\mathbf{F}_p^N$, then*

$$\mathbf{E}v_{\leqslant s}(L) = \frac{p^k-1}{p^N-1}\sum_{u=1}^{s} C_N^u (p-1)^u,$$

$$\mathbf{E}v_{\geqslant s}(L) = \frac{p^k-1}{p^N-1}\sum_{u=s}^{N} C_N^u (p-1)^u.$$

If $\mu_*(L) = \min\{w(X)\colon X \in L\backslash\{0\}\}$ and $\mu^*(L) = \max\{w(X)\colon X \in L\}$ are the minimal and maximal weights of non-zero codewords in L, then

$$\frac{1}{1+\frac{p^N-p^k}{p^N-1}\,(p-1)(\mathbf{E}v_{\leqslant s}(L))^{-1}} \leqslant \mathbf{P}\{\mu_*(L) \leqslant s\} \leqslant \mathbf{E}v_{\leqslant s}(L), \tag{1}$$

$$\frac{1}{1+\frac{p^N-p^k}{p^N-1}\,(p-1)(\mathbf{E}v_{\geqslant s}(L))^{-1}} \leqslant \mathbf{P}\{\mu^*(L) \geqslant s\} \leqslant \mathbf{E}v_{\geqslant s}(L). \tag{2}$$

Note that Poisson limit theorems for random variables $v_s(L)$ under somewhat another assumptions on the distributions of random codes were proved in [6], [4].

Equalities for $\mathbf{E}v_{\leqslant s}(L)$ and $\mathbf{E}v_{\geqslant s}(L)$ involve binomial sums. The inequality

$$\begin{gathered}\mathbf{E}v_{\leqslant s+1}(L) \geqslant \frac{p^k-1}{p^N-1}\sum_{u=2}^{s+1} C_N^u (p-1)^u = \\ = \frac{p^k-1}{p^N-1}\sum_{u=1}^{s} C_N^u (p-1)^u\,\frac{N-u}{u+1}(p-1) \geqslant \frac{N-s}{s+1}(p-1)\mathbf{E}v_{\leqslant s}(L)\end{gathered} \tag{3}$$

shows that the sequence $\mathbf{E}v_{\leqslant s}(L)$ is growing almost geometrically if $s < cN\,\frac{p-1}{p}$ where N is large and $c \in (0,1)$ is separated from 1.

The binomial sums may be estimated by means of inequalities proved in [7].

Theorem 2. *Let $H(x,r) = x\ln\frac{x}{r} + (1-x)\ln\frac{1-x}{1-r}$, $\operatorname{sgn}(x) = \frac{x}{|x|}$ for $x \neq 0$ and $\operatorname{sgn}(0) = 0$, let $\{C_{N,r}(m)\}_{m=0}^N$ be increasing sequences defined as follows:*

$C_{N,r}(0) = (1-r)^N,\ C_{N,r}(N) = 1 - r^N,$

$$C_{N,r}(m) = \Phi\left(\operatorname{sgn}\left(\tfrac{m}{N} - r\right)\sqrt{2NH\left(\tfrac{m}{N}, r\right)}\right),\ 1 \leqslant m < N.$$

Then for every $m = 0, 1, \ldots, N-1$ *and for every* $r \in (0,1)$

$$C_{N,r}(m) \leqslant \sum_{u=0}^{m} C_N^u r^u (1-r)^{N-u} \leqslant C_{n,r}(m+1), \tag{4}$$

and inequalities become equalities only for $m = 0$ *or* $m = N-1$.

If $r \in (0,\ 1)$ is fixed, then the function $H(x,r)$ is convex ($H''_x(x,r) = \frac{1}{x(1-x)} > 0$); it decreases monotonically from $H(0,r) = -\ln(1-r)$ to $H(r,r) = 0$ and further increases to $H(1,r) = -\ln r$.

It follows from the Theorem 2 that for $0 < s < \frac{p-1}{p}$

$$\Phi\left(-\sqrt{2NH\left(\tfrac{s}{N}, \tfrac{p-1}{p}\right)}\right) < \frac{1}{p^N}\sum_{u=0}^{s} C_N^u (p-1)^u < \Phi\left(-\sqrt{2NH\left(\tfrac{s+1}{N}, \tfrac{p-1}{p}\right)}\right). \tag{5}$$

Further, it follows from (1) that $\mathbf{P}\{\mu_*(L) \leqslant s\}$ is close to 1 if $\mathbf{E}v_{\leqslant s}(L)$ is large and $\mathbf{P}\{\mu_*(L) \leqslant s\}$ is close to 0 if $\mathbf{E}v_{\leqslant s}(L)$ is small. So, typical values of $\mu_*(L)$ for the random uniformly distributed linear code L correspond to the interval where $\mathbf{P}\{\mu_*(L) \leqslant s\}$ increases from small values to values close to 1 and $\mathbf{E}v_{\leqslant s}(L)$ increases from small to large values.

In view of Theorem 1 and inequalities (5) we define values $z_k > 0$ and $x_k < 1 - \frac{1}{p}$ as solutions of equations

$$\Phi\left(-\sqrt{2z_k}\right) = \tfrac{1}{p^k}, \quad NH\left(x_k, 1 - \tfrac{1}{p}\right) = z_k. \tag{6}$$

Statement 1. *If* $k \geqslant 4$ *and* $k\ln p \geqslant 2\ln(4e\pi k \ln p)$, *then*

$$|z_k - (k\ln p - \ln(4\pi k \ln p))| < 1. \tag{7}$$

and

$$\mathbf{E}v_{\leqslant Nx_k - 1}(L) < 1 < \mathbf{E}v_{\leqslant Nx_k + 1}(L). \tag{8}$$

Proof. To estimate the value of z_k we use well-known (see [3]) inequalities

$$\left(1-\frac{1}{x^2}\right)\frac{1}{\sqrt{2\pi}|x|}\,e^{-x^2/2}<\Phi(x)<\frac{\varphi(x)}{|x|}=\frac{1}{\sqrt{2\pi}|x|}\,e^{-x^2/2},\quad x<0.$$

Then for the first equation in (6) we have

$$\left(1-\frac{1}{2z_k}\right)\frac{1}{\sqrt{4\pi z_k}}\,e^{-z_k}<\Phi\left(-\sqrt{2z_k}\,\right)=\frac{1}{p^k}<\frac{1}{\sqrt{4\pi z_k}}\,e^{-z_k}. \tag{9}$$

If $k\geqslant 4$, then $z_k>1$ and $\ln\left(1-\frac{1}{2z_k}\right)>-1$, so it follows from (9) that

$$z_k+\frac{1}{2}\,\ln(4\pi z_k)<k\ln p<z_k+\frac{1}{2}\,\ln(4\pi z_k)+1\,.$$

According to the left inequality $z_k<k\ln p$; substituting this estimate into the right inequality we obtain $k\ln p<z_k+\ln(4\pi k\ln p)+1$, or

$$z_k>k\ln p-\ln(4\pi k\ln p)-1.$$

Using this estimate along with the left inequality, we find that

$$z_k<k\ln p-\ln\left(4\pi(k\ln p-\ln(4\pi k\ln p)-1)\right)=$$

$$=k\ln p-\ln(4\pi k\ln p)-\ln\left(1-\frac{\ln(4e\pi k\ln p)}{k\ln p}\right)<k\ln p-\ln(4\pi k\ln p)+1$$

if $k\ln p\geqslant 2\ln(4e\pi k\ln p)$. This proves (7).

Further, if $NH\left(x_k,1-\frac{1}{p}\right)=z_k$, then

$$\Phi\left(-\sqrt{2NH\left(\tfrac{[Nx_k]}{N},1-\tfrac{1}{p}\right)}\right)\leqslant$$

$$\leqslant\Phi\left(-\sqrt{2NH\left(x_k,1-\tfrac{1}{p}\right)}\right)=\frac{1}{p^k}<\Phi\left(-\sqrt{2NH\left(\tfrac{[Nx_k]+1}{N},1-\tfrac{1}{p}\right)}\right)$$

and

$$\frac{1}{p^k}<\Phi\left(-\sqrt{2NH\left(\tfrac{[Nx_k]+1}{N},1-\tfrac{1}{p}\right)}\right)<\frac{1}{p^N}\sum_{u=0}^{[Nx_k]+1}C_N^u(p-1)^u,$$

$$\frac{1}{p^N}\sum_{u=0}^{[Nx_k]-1}C_N^u(p-1)^u<\Phi\left(-\sqrt{2NH\left(\tfrac{[Nx_k]}{N},1-\tfrac{1}{p}\right)}\right)\leqslant\frac{1}{p^k}\,.$$

This proves (8). □

So, approximate typical values $s < \frac{N(p-1)}{p}$ of $\mu_*(L)$ for random uniformly distributed k-dimensional code L in $\mathbf{F}_p^N$ satisfy the equation

$$H\left(\tfrac{s}{N}, \tfrac{p-1}{p}\right) = \tfrac{1}{N}\ (k\ln p - \ln(4\pi k \ln p))\,. \qquad (10)$$

In particular, if dimension k of random code L and dimension N of the space $\mathbf{F}_p^N$ are growing proportionally, then the typical value s of the minimal nonzero codeword weight also is growing proportionally to N. Upper bounds of typical values of $\mu_*(L)$ are close to lower ones because according to (3) values of $\mathbf{E}v_{\leqslant s}(L)$ for typical values of $\mu_*(L)$ are increasing as geometric progression.

The case of maximal codeword weight of random linear k-dimensional code may be considered analogously. In this case instead of equations (6) we should consider equations

$$\Phi\left(\sqrt{2z_k}\,\right) = 1 - \tfrac{1}{p^k}, \quad NH\left(x_k, 1-\tfrac{1}{p}\right) = z_k,\ x_k > 1 - \tfrac{1}{p}\,. \qquad (11)$$

So, the solution z_k of the first equation in (11) satisfy the same conditions (7) and approximate typical values $s > \frac{N(p-1)}{p}$ of $\mu^*(L)$ satisfy the equation

$$H\left(\tfrac{s}{N}, \tfrac{p-1}{p}\right) = \tfrac{1}{N}\ (k\ln p - \ln(4\pi k \ln p))\,. \qquad (12)$$

Let us illustrate this results by some numerical and graphical examples.

Fig. 1 presents lower and upper bounds for probabilities $\mathbf{P}\{\mu_*(L) \leqslant s\}$ for $p = 2$, $L \subset \mathbf{F}_2^{128}$ and some $k = \dim L$. Same bounds for $L \subset \mathbf{F}_2^{1024}$ are presented on fig. 2.

The equation (10) leads to the following estimates of the typical values of $\mu_*(L)$ for considered values of N and k :

	$k = 3N/4$	$k = N/2$	$k = N/4$
$N = 128$	7.628	17.293	32.176
$N = 1024$	45.533	116.727	225.669

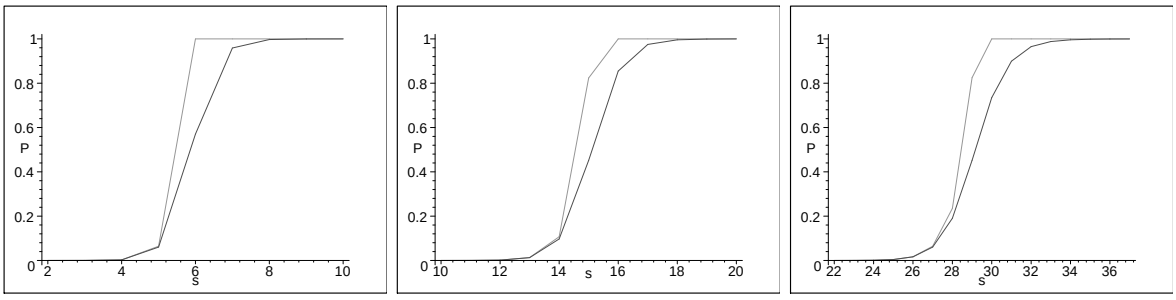

Figure 1: From left to right: $k = 3N/4$, $k = N/2$, $k = N/4$.

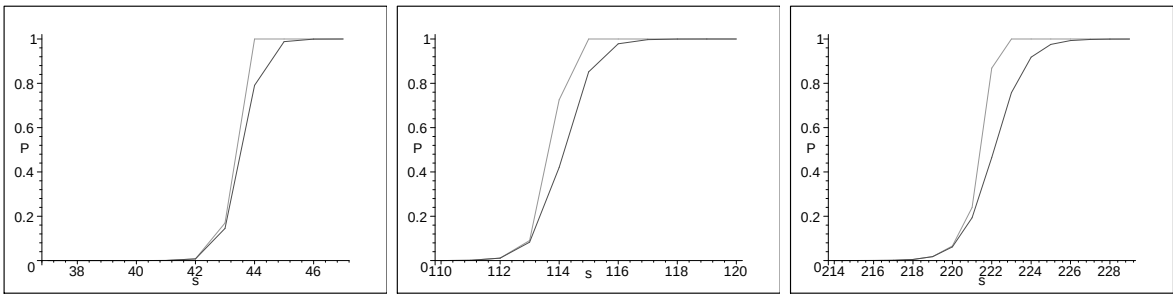

Figure 2: From left to right: $k = 3N/4$, $k = N/2$, $k = N/4$.

Fig. 3 and fig. 4 present lower and upper bounds for probabilities $\mathbf{P}\{\mu^*(L) \geqslant s\}$ for $L \subset \mathbf{F}_2^{128}$ and for $L \subset \mathbf{F}_2^{1024}$ correspondingly.

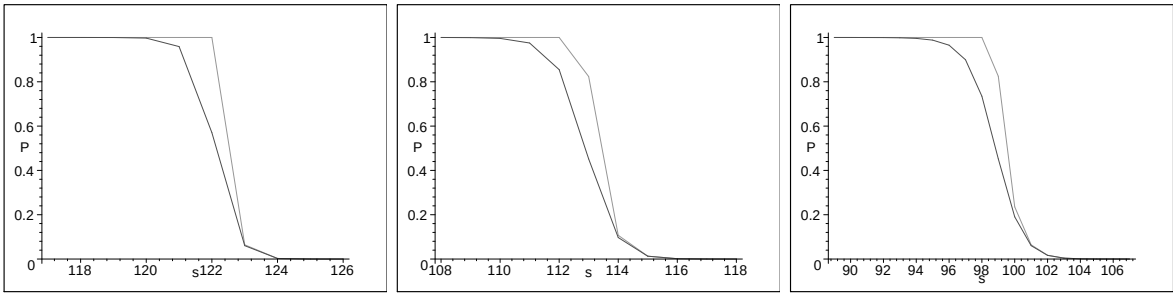

Figure 3: From left to right: $k = 3N/4$, $k = N/2$, $k = N/4$.

The equation (12) leads to the following estimates of the typical values of $\mu^*(L)$:

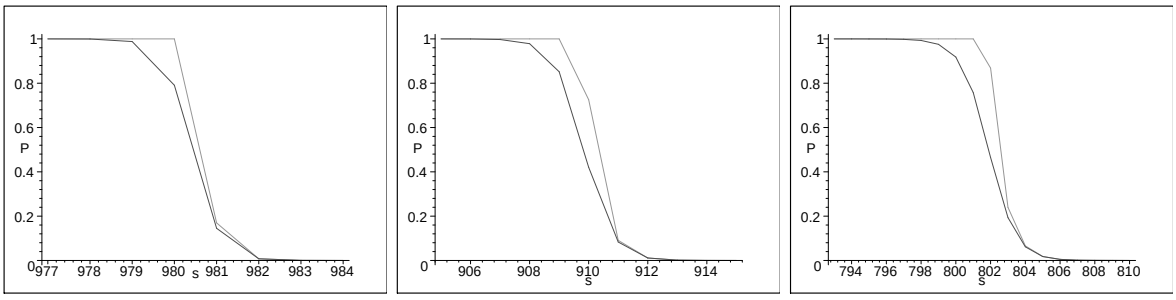

Figure 4: From left to right: $k = 3N/4$, $k = N/2$, $k = N/4$.

	$k = 3N/4$	$k = N/2$	$k = N/4$
$N = 128$	120.371	110.706	95.823
$N = 1024$	978.466	907.272	798.330

Comparing inequalities (1) and (2) we can note that

$$\mathbf{E}v_{\geqslant s}(L) = \frac{2^k - 1}{2^N - 1} \sum_{r=s}^{N} C_N^r = \frac{2^k - 1}{2^N - 1} \sum_{r=0}^{N-s} C_N^r = \mathbf{E}v_{\leqslant N-s}(L) + \frac{2^k - 1}{2^N - 1},$$

and thus lower and upper bounds for probability $\mathbf{P}\{\mu^*(L) \geqslant s\}$ differ only very slightly from lower and upper bounds for probability $\mathbf{P}\{\mu_*(L) \leqslant N - s\}$.

The estimates of the minimal weights of nonzero codewords may be used to choose the parameters of McEliece cryptosystem [5]. If McEliece cryptosystem is used and someone has intercepted single encrypted message, he or she faces the problem of decoding for seemingly random linear code and such a problem is equivalent ([2], p. 368) to a problem of finding codeword of minimal weight in extended code with dimension increased by 1. It was shown in [1] that general problem of determining minimal weight of codeword for given code is NP-hard.

According to estimate (10) for the case of binary spaces $\mathbf{F}_2^N$ and linear codes of dimension $k = N/2$, the typical values of minimal weight $\mu_*(L)$ of non-zero codeword for random linear code L may be approximated by value $0.110023\ldots \cdot N$ that corresponds to propositions of well-known paper [2] where probabilistic algorithm for finding vectors of minimal weight in random binary (n, k)-code with minimal Hamming distance d between elements of

code is suggested. One may compare values of parameters n, k, d from chapter 3.3.1 of [2] and numerical estimates from (10):

n	k	d	$\mu_*(L)$ estim.
64	32	7	10.043
128	64	15	17.293
256	128	29	31.634
512	256	57	60.088
768	384	85	88.431
1024	512	113	116.727
1536	768	170	173.244
2048	1024	226	229.710

References

[1] Berlekamp E.R., McEliece R.J., van Tilborg H.C.A. On the inherent intractability of certain coding problems. — IEEE Trans. Inf. Theory, 1978, **24**, p. 384–386.

[2] Canteaut A., Chabaud F. A new algorithm for finding minimum-weight words in a linear code: application to McEliece's cryptosystem and to narrow-sense BCH codes of length 511. — IEEE Trans. Inf. Theory, 1988, **44**:1, p. 367–378.

[3] Feller W. An Introduction to Probability Theory and its Applications, Vol. 1 (3rd ed.). — N.Y. e.a., J.Wiley & Sons, 1970.

[4] Kopyttsev V. A., Mikhailov V. G. Poisson-type theorems for the number of special solutions of a random linear inclusion. — Discrete Math. Appl., 2010, **20**:2, p. 191–211.

[5] McEliece R.J. A public-key cryptosystem based on algebraic coding theory. — Jet Propulsion Lab. DSN Progress Report 42-44, 1978.

[6] Mikhailov V. G. Limit theorems for the number of solutions of a system of random linear equations belonging to a given set. — Discrete Math. Appl., 2007, **17**:1, pp. 13–22.

[7] Zubkov A. M., Serov A. A., A complete proof of universal inequalities for distribution function of binomial law. — Theory of Probab. and its Appl., 2012, **57**:3, pp. 597–602.

[8] Zubkov A. M., Kruglov V. I. Statistical characteristics of weight spectra of random linear codes over GF(p). — Matem. Voprosy Kriptografii, 2014, **5**:1, pp. 27–38 (in Russian).

Poisson Approximation for Non-Decreasing Runs in Markov Chains

Alexander Minakov

Abstract

We consider a sequence $X_1,\ X_2,\ \ldots, X_n$ of random variables generated by a stationary irreducible and aperiodic Markov chain with state space $\mathcal{A} = \{1,\ \ldots,\ N\}$, $N \geq 1$. We study the non-overlapping appearances of non-decreasing runs in the sequence $X_1,\ X_2,\ \ldots, X_n$. By means of Stein's method we estimate the total variation distance between the distribution of the number of non-overlapping appearances of non-decreasing monotone runs and Poisson distribution. As corollary for this result we prove the appropriate limit theorem.

Keywords: non-decreasing runs, Poisson approximation, Stein's method, Markov chain, Jordan normal form, upper triangular matrix.

1 Introduction

Let $X = (X_1,\ X_2,\ \ldots, X_n)$ be a segment of stationary irreducible and aperiodic Markov chain $\{X_a\}_{a\in\mathbb{Z}}$ on a finite space $\mathcal{A} = \{1,\ \ldots,\ N\}$, $N \geq 1$, with transition probability matrix $\mathbb{P} = (p_{ij})_{(i,j)\in\mathcal{A}\times\mathcal{A}}$ and stationary distribution $\vec{\pi} = (\pi_i)_{i\in\mathcal{A}}$.

A non-decreasing run length s $(s \geq 2)$ with the beginning at local minimum at time $t+1$ is called an event

$$E_{t,s} = \{X_t > X_{t+1} \leq \ldots \leq X_{t+s}\}.$$

Let $\mathrm{I}\{B\}$ be a indicator of the event B. We define the random variable

$$\xi_n(s) = \sum_{t=1}^{n} \mathrm{I}\{E_{t,s}\}, \tag{1}$$

which enumerates the non-overlapping appearances of non-decreasing runs with length $s \geq 2$, that begin at local minimum in the sequence X_1, X_2, ..., X_n.

In order to avoid edge effects and to facilitate calculations, we assume that doubly infinite sequence $\{X_a\}_{a\in\mathbb{Z}}$ is observed. Therefore at the beginning or at the end of $X_1,\ X_2,\ \ldots,\ X_n$ the structure of runs is the same.

In papers [4, 10], the non-overlapping appearances of increasing runs were studied in random permutations. In the paper [9] the Poisson limit theorem was proved for the number of increasing runs with length larger or equal to some fixed length. The Poisson limit theorem for increasing runs of fixed length was proved in the paper [3]. The author of the paper [7] formulated multinomial normal theorem for the number of increasing runs with fixed length. In the paper [8] the compound Poisson limit theorem was proved for the number of non-decreasing runs. All theorems was formulated for sequences of independent identically distributed random variables.

In this paper we construct an estimate of the variation distance between distribution of random variable $\xi_n(s)$ and Poisson distribution in the stationary irreducible and aperiodic Markov chain. As corollary for this result we prove the appropriate Poisson limit theorem.

2 Main results

Let us recall some definitions. By $d(\Phi, \Psi)$ denote the variation distance between distributions Φ and Ψ. For the distribution Φ and Ψ on the set $\{0,\ 1,\ \ldots\}$ (see the book [2])

$$d(\Phi, \Psi) = \frac{1}{2}\sum_{m=0}^{\infty} |\Psi\{m\} - \Phi\{m\}|.$$

Let $L(\zeta)$ denote a distribution of random variable ζ.

Let $\mathbb{M} = (m_{ij})_{(i,j)\in\mathcal{A}\times\mathcal{A}}$ be the matrix such that all the entries below the main diagonal are zero and the other entries are equal to appropriate elements of the matrix $\mathbb{P}$ ($m_{ij} = p_{ij}$, for $i \geq j$). The matrix $\mathbb{M}$ defines transition probabilities from $i \in \mathcal{A}$ to $j \in \mathcal{A}$ under the condition $i \leq j$.

By the Jordan canonical form theorem (see the book [5]) there exists invertible matrix $\mathbb{T} = (t_{ij})_{(i,j)\in\mathcal{A}\times\mathcal{A}}$ $\left(\text{and } \mathbb{T}^{-1} = \left(t'_{ij}\right)_{(i,j)\in\mathcal{A}\times\mathcal{A}}\right)$ such that

$$\mathbb{M} = \mathbb{T}\cdot\mathbb{J}\cdot\mathbb{T}^{-1}, \tag{2}$$

where

$$\mathbb{J} = \mathrm{Diag}\left\{\mathbb{J}_{d_1}\left(p_{i_1,i_1}\right), \mathbb{J}_{d_2}\left(p_{i_2,i_2}\right), \ \ldots, \ \mathbb{J}_{d_q}\left(p_{i_q,i_q}\right)\right\} \tag{3}$$

is a block diagonal matrix and $\mathbb{J}_{d_1}\left(p_{i_1,i_1}\right)$, ..., $\mathbb{J}_{d_q}\left(p_{i_q,i_q}\right)$ are Jordan blocks of order $d_1, \ldots, d_q$ and $d_1 + \ldots + d_q = N$. The matrix $\mathbb{J}$ has $q \in \{1, \ldots, N\}$ blocks.

Choose the matrix $\mathbb{T}$ such that the Jordan blocks are arranged from greatest eigenvalue to least and then the subdiagonal 1 blocks are arranged from longest to shortest inside each Jordan block. Suppose $b \in \{1, \ldots, q\}$ first Jordan blocks have greatest eigenvalue that equals p:

$$p_{i_1,i_1} = \ldots = p_{i_b,i_b} \equiv p. \tag{4}$$

Let

$$\mathbb{K} = \mathbb{P} - \mathbb{M}. \tag{5}$$

The matrix $\mathbb{K}$ defines transition probabilities from $i \in \mathcal{A}$ to $j \in \mathcal{A}$ under the condition $i > j$.

Suppose

$$\vec{\alpha} = (\alpha_1, \ \ldots, \alpha_N) = \vec{\pi}\mathbb{K}\mathbb{M}^{s-1}. \tag{6}$$

The sum of vector's components $\vec{\alpha}$ equals the probability of the non-decreasing run length s with the beginning at local minimum. Therefore, for any $t \in \{1, \ \ldots, \ n\}$ and $s \geq 2$ we have

$$\mathbf{P}\{E_{t,s}\} = \alpha_1 + \ldots + \alpha_N. \tag{7}$$

Let λ be the expectation value of random value $\xi_n(s)$. From (7) it follows that

$$\lambda = \mathbf{E}\xi_n(s) = \sum_{t=1}^{n} \mathbf{P}\{E_{t,s}\} = n\cdot\mathbf{P}\{E_{t,s}\} = n\cdot(\alpha_1 + \ldots + \alpha_N). \tag{8}$$

Let $p_{ij}^{(t)}$ denote the t-order transition probability of the irreducible and aperiodic Markov chain $\{X_a\}_{a\in\mathbb{Z}}$ and $p_{(R)ij}^{(t)} = \pi_j p_{ji}^{(t)} {\pi_i}^{-1}$ the t-order transition probability of the reversed Markov chain $\{X_a^R\}_{a\in\mathbb{Z}}$, for $t \in \mathbb{N}$. Following a coupling argument (see the paper [6]), we deduce

$$\max_{x\in\mathcal{A}} \max\left\{ \sum_{y\in\mathcal{A}} \left| p_{xy}^{(t)} - \pi_y \right|, \sum_{y\in\mathcal{A}} \left| p_{(R)xy}^{(t)} - \pi_y \right| \right\} \leq 2\varrho^t, \forall t \geq 1, \tag{9}$$

where

$$\varrho = 1 - \min\left\{ \sum_{y\in\mathcal{A}} \min_{x\in\mathcal{A}} p_{xy}, \sum_{y\in\mathcal{A}} \min_{x\in\mathcal{A}} p_{(R)xy} \right\}. \tag{10}$$

Now we are ready to state the first main result of this work. Using Stein's method and results of the paper [1] we prove the following theorem.

Theorem 1. *Let $X_1, X_2, \ldots, X_n$ be a segment of stationary irreducible and aperiodic Markov chain $\{X_a\}_{a\in\mathbb{Z}}$ on a finite space $\mathcal{A} = \{1, \ldots, N\}$, $N \geq 1$, with transition probability matrix $\mathbb{P} = (p_{ij})_{(i,j)\in\mathcal{A}\times\mathcal{A}}$ and stationary distribution $\vec{\pi} = (\pi_i)_{i\in\mathcal{A}}$, then*

$$d\Big(\mathcal{L}\big(\xi_n(s)\big), Pois(\lambda)\Big) \leq \frac{\left(1 - e^{-\lambda}\right)(2s + 2r + 3)\,\lambda}{n} + 2\lambda\varrho^{r+1}\left(2 + 2\varrho^{s+r+1} + \varrho^{2s+r+1}\right), \tag{11}$$

where λ defined in (8), ϱ defined in (10), and $r \geq 0$.

Follows from Theorem 1 we give limit theorem for random variable (1).

Theorem 2. *Let $X_1, X_2, \ldots, X_n$ be a segment of stationary irreducible and aperiodic Markov chain $\{X_a\}_{a\in\mathbb{Z}}$ on a finite space $\mathcal{A} = \{1, \ldots, N\}$, $N \geq 1$, with transition probability matrix $\mathbb{P} = (p_{ij})_{(i,j)\in\mathcal{A}\times\mathcal{A}}$ and stationary distribution $\vec{\pi} = (\pi_i)_{i\in\mathcal{A}}$. if the following conditions hold*

$$n, s \to \infty\ ,\ \ \frac{s}{n} \to 0\ \ np^{s-2N}\frac{(s-1)^{2N-1}}{(2N-1)!} \to Q \in (0, \infty), \tag{12}$$

then

$$\mathcal{L}\big(\xi_n(s)\big) \to Pois(\lambda).$$

3 Proof of Theorems 1 and 2

Following the paper [1] we consider the sum $W = \sum_{i\in I} Y_i$ of random indicators $\{Y_i | i \in I\}$. Let Γ_i be an arbitrary finite set of indices for any $i \in I$. Γ_i consist of $j \neq i$ such that Y_j depends on Y_i strongly. Let us define the following variables

$$A^* = \sum_{i=1}^{n} \sum_{j\in\Gamma_i\cup\{i\}} \mathbf{P}\{Y_i = 1\}\mathbf{P}\{Y_j = 1\},$$

$$B^* = \sum_{i=1}^{n} \sum_{j\in\Gamma_i} \mathbf{P}\{Y_i = 1, Y_j = 1\},$$

$$C^* = \sum_{i=1}^{n} \mathbf{E}\Big|\mathbf{E}\Big\{\big(Y_i - \mathbf{P}\{Y_i = 1\}\big) : \sigma\big(Y_j, j \notin \Gamma_i\big)\Big\}\Big|,$$

where $\sigma\big(Y_j, j \in M\big)$ is a σ-algebra generated by $\big\{Y_j, j \in M\big\}$.

In the paper [1], the estimate of the total variation distance between the sum W and Poisson distribution with expectation value $\theta = \mathbf{E}W$ was constructed:

$$d\Big(\mathcal{L}\big(W\big), Pois\,(\theta)\Big) \leq \frac{1 - e^{-\theta}}{\theta}\,(A^* + B^*) + \max\{1; 1.4\theta^{-1/2}\}C^*. \qquad (13)$$

Let us use this estimate for random value (1). For any $i \in \{1, \ldots, n\}$ and $r \geq 0$ choose the set $\Gamma_i = \{i - s - r, \ldots, i + s + r\}\backslash\{i\}$ and values

$$A^* = \sum_{i=1}^{n} \sum_{j\in\Gamma_i\cup\{i\}} \mathbf{P}\{E_{i,s}\}\mathbf{P}\{E_{j,s}\},$$

$$B^* = \sum_{i=1}^{n} \sum_{j\in\Gamma_i} \mathbf{P}\{E_{i,s}, E_{j,s}\},$$

$$C^* = \sum_{i=1}^{n} \mathbf{E}\Big|\mathbf{E}\Big(\big(I\{E_{i,s}\} - \mathbf{P}\{E_{i,s}\}\big)|\sigma\big(I\{E_{j,s}\}, j \notin \Gamma_i\big)\Big)\Big|.$$

The set Γ_i is selected such that $E_{i,s}$ and $E_{j,s}$ are "weakly" dependent whenever $j \geq i + s + r$ or $i \geq j + s + r$ for sufficiently large r. We note that

r results from Markovian structure of the problem, which necessitates a larger neighborhood that in of independent identically distributed random variables $X_1, \ldots, X_n$.

Let the vector

$$\vec{\beta}^{(h)} = \left(\beta_1^{(h)}, \ \ldots, \beta_N^{(h)}\right) = \vec{\pi}\mathbb{K}\mathbb{M}^{s-1}\mathbb{P}^h\mathbb{K}\mathbb{M}^{s-1}. \tag{14}$$

The sum of vector's components $\beta_1^{(h)} + \ldots + \beta_N^{(h)}$ equals to probability the non-overlapping appearances of two non-decreasing runs with length s at a distance h between the end of the one run and the beginning of the other run.

We note that two non-decreasing runs with the beginning at local minimum have not common elements. In other words if $I\{E_{i,s}\} = 1$ then $I\{E_{j,s}\} = 0$ for any $j \in \{i - s + 1, \ldots, i + s - 1\}$. Hence

$$\mathbf{P}\{E_{i,s}, E_{j,s}\} = 0 \tag{15}$$

for any $j \in \{i - s + 1, \ldots, i + s - 1\}$.

If $j \leq i - s$ or $j \geq i + s$ then

$$\mathbf{P}\{E_{i,s}, E_{j,s}\} = \beta_1^{(h)} + \ldots + \beta_N^{(h)}. \tag{16}$$

Using (7), (14), (15), and (16) we get

$$A^* = n\,(2s+1)\,(\alpha_1 + \ldots + \alpha_N)^2\,, \tag{17}$$

$$B^* = 2n \sum_{h=0}^{r} \left(\beta_1^{(h)} + \ldots + \beta_N^{(h)}\right) \leq 2n\,(r+1)\,(\alpha_1 + \ldots + \alpha_N)^2\,, \tag{18}$$

where $r \geq 0$ results from Markovian structure of the problem.

Next, we turn our attention to the term C^*. Define an event $E_{t,s}\,(x, y) = \{X_t > X_{t+1} \leq \ldots \leq X_{t+s} > X_{t+s+1} | X_t = x, X_{t+s+1} = y\}$, $x, y \in \mathcal{A}$. It holds true that $\sigma\big(I\{E_{j,s}\} | j \notin \Gamma_i\big) \subseteq \sigma\big(X_1, \ldots, X_{i-s-1}, X_{i+s+1}, \ldots, X_n\big)$, and us-

ing the Markov property and definition (1), we obtain

$$C^* \leq \sum_{i=1}^{n} \mathbf{E}\Big|\mathbf{E}\Big(\big(I\{E_{i,s}\} - \mathbf{P}\{E_{i,s}\}\big)|\sigma\big(X_1, ..., X_{i-s-r-1}, X_{i+s+r+1}, ..., X_n\big)\Big)\Big|$$

$$\leq \sum_{i=1}^{n} \sum_{x,y\in\mathcal{A}} \mathbf{E}\Big|\mathbf{E}\big(I\{E_{i,s}(x,y)\}|\sigma\big(X_{i-s-r-1}, X_{i+s+r+1}\big)\big) - \mathbf{P}\{E_{i,s}(x,y)\}\Big|$$

$$= \sum_{i=1}^{n} \sum_{z,w\in\mathcal{A}} \sum_{x,y\in\mathcal{A}} \Big|\mathbf{P}\big\{I\{E_{i,s}(x,y)\} = 1, X_{i-s-r-1} = z, X_{i+s+r+1} = w\big\}$$

$$- \mathbf{P}\{E_{i,s}(x,y)\}\mathbf{P}\{X_{i-s-r-1} = z, X_{i+s+r+1} = w\}\Big|. \quad (19)$$

We have (19):

$$\mathbf{P}\{X_{i-s-r-1} = z, X_{i+s+r+1} = w\} = \pi_z p_{zw}^{(2s+2r+2)}$$

and

$$\mathbf{P}\big\{I\{E_{i,s}(x,y)\} = 1, X_{i-s-r-1} = z, X_{i+s+r+1} = w\big\}$$

$$= p_{(R)xz}^{(s+r+1)} \mathbf{P}\{E_{i,s}(x,y)\} p_{yw}^{(r+1)},$$

where $p_{(R)xz}^{(t)} = \pi_z p_{zx}^{(t)}/\pi_x$ is the transition probability of order t of the reversed Markov chain. Furthermore, if we set

$$\varepsilon_{xy}^{(t)} = \left|p_{xy}^{(t)} - \pi_y\right| \qquad \varepsilon_{(R)xy}^{(t)} = \left|p_{(R)xy}^{(t)} - \pi_y\right|,$$

then

$$\Big|\mathbf{P}\big\{I\{E_{i,s}(x,y)\} = 1, X_{i-s-r-1} = z, X_{i+s+r+1} = w\big\}$$

$$- \mathbf{P}\{E_{i,s}(x,y)\}\mathbf{P}\{X_{i-s-r-1} = z, X_{i+s+r+1} = w\}\Big|$$

$$= \mathbf{P}\{E_{i,s}(x,y)\}\left|p_{(R)xz}^{(s+r+1)} p_{yw}^{(r+1)} - \pi_z p_{zw}^{(2s+2r+2)}\right|$$

$$\leq \mathbf{P}\{E_{i,s}(x,y)\}\left(\varepsilon_{(R)xz}^{(s+r+1)}\varepsilon_{yw}^{(r+1)} + \pi_w\varepsilon_{(R)xz}^{(s+r+1)} + \pi_z\varepsilon_{yw}^{(r+1)} + \pi_z\varepsilon_{zw}^{(2s+2r+2)}\right),$$

and substituting this to relation (19), we get

$$C^* \le \sum_{i=1}^{n} \sum_{x,y\in\mathcal{A}} \mathbf{P}\{E_{i,s}(x,y)\} \times \left(\sum_{z,w\in\mathcal{A}} \varepsilon_{(R)xz}^{(s+r+1)} \varepsilon_{yw}^{(r+1)} + \sum_{z\in\mathcal{A}} \varepsilon_{(R)xz}^{(s+r+1)} + \sum_{w\in\mathcal{A}} \varepsilon_{yw}^{(r+1)} + \sum_{z,w\in\mathcal{A}} \pi_z \varepsilon_{zw}^{(2s+2r+2)} \right).$$

Using (9), we obtain

$$\begin{aligned} C^* &\le \sum_{i=1}^{n} \sum_{x,y\in\mathcal{A}} \mathbf{P}\{E_{i,s}(x,y)\} \left(4\varrho^{s+2r+2} + 2\varrho^{s+r+1} + 2\varrho^{r+1} + 2\varrho^{2s+2r+2}\right) \\ &\le \sum_{i=1}^{n} \sum_{x,y\in\mathcal{A}} \mathbf{P}\{E_{i,s}(x,y)\} 2\varrho^{r+1} \left(2 + 2\varrho^{s+r+1} + \varrho^{2s+r+1}\right) \\ &= 2\lambda\varrho^{r+1} \left(2 + 2\varrho^{s+r+1} + \varrho^{2s+r+1}\right), \end{aligned} \tag{20}$$

where λ is defined in (8).

Now we formulate the lemma in which we calculate the expectation value of random value $\xi_n(s)$.

Lemma 1.

$$\lambda = \mathbf{E}\xi_n(s) = n \cdot \sum_{l=1}^{N} \sum_{k=1}^{N-1} w_{k,l} \sum_{t=k+1}^{N} \pi_t p_{t,k}, \tag{21}$$

where for any $k \le l$:

$$w_{k,l} = \sum_{r=1}^{q} \sum_{j=d_0+d_1+\ldots+d_{r-1}}^{d_1+\ldots+d_r} t'_{j,l} \sum_{i=1}^{j} t_{k,i} p_{i_r,i_r}^{s-j+i-d_0-d_1-\ldots-d_{r-1}} \binom{s-1}{s-j+i-d_0-d_1-\ldots-d_{r-1}},$$

where $d_0 \equiv 1$, and for any $k > l$:

$$w_{k,l} = 0.$$

Proof of Lemma 1 you can see in Appendix.

If we combine (17), (18), (20), and the results of Lemma 1 with (13), we get the estimate (11). This completes the proof of Theorem 1.

Let us prove that under conditions of Theorem 2 λ is bounded above.

$$\lambda = n \cdot \sum_{l=1}^{N} \sum_{k=1}^{N-1} \sum_{t=k+1}^{N} \pi_t p_{t,k} \sum_{r=1}^{q} \sum_{j=d_0+d_1+\ldots+d_{r-1}}^{d_1+\ldots+d_r} t'_{j,l} \times \sum_{i=1}^{j} t_{k,i} p_{i_r,i_r}^{s-j+i-d_0-d_1-\ldots-d_{r-1}} \binom{s-1}{s-j+i-d_0-d_1-\ldots-d_{r-1}}$$

$$\leq n\binom{s-1}{s-2N} \cdot \sum_{r=1}^{q} p_{i_r,i_r}^{s-2N} \cdot \sum_{l=1}^{N} \sum_{k=1}^{N-1} \sum_{t=k+1}^{N} \pi_t p_{t,k} \sum_{j=d_0+d_1+\ldots+d_{r-1}}^{d_1+\ldots+d_r} t'_{j,l} \sum_{i=1}^{j} t_{k,i} \equiv \lambda'.$$

Using (4) we isolate summands depend on eigenvalue p:

$$\lambda' = n\binom{s-1}{s-2N} p^{s-2N} b \cdot \sum_{r=1}^{b} \sum_{l=1}^{N} \sum_{k=1}^{N-1} \sum_{t=k+1}^{N} \pi_t p_{t,k} \sum_{j=d_0+d_1+\ldots+d_{r-1}}^{d_1+\ldots+d_r} t'_{j,l} \sum_{i=1}^{j} t_{k,i}$$

$$+ n\binom{s-1}{s-2N} \cdot \sum_{r=b+1}^{q} p_{i_r,i_r}^{s-2N} \cdot \sum_{l=1}^{N} \sum_{k=1}^{N-1} \sum_{t=k+1}^{N} \pi_t p_{t,k} \sum_{j=d_0+d_1+\ldots+d_{r-1}}^{d_1+\ldots+d_r} t'_{j,l} \sum_{i=1}^{j} t_{k,i}.$$

Note that under conditions $s \to \infty$ and $G = \text{const} > 0$ we have

$$\binom{s+G}{s} = \binom{s+G}{G} = \frac{(s+G)^G}{G!} (1 + o(1)). \tag{22}$$

Using (22) and under conditions (12), we obtain

$$\lambda' = n \cdot \tfrac{(s-1)^{2N-1}}{(2N-1)!} \Big(p^{s-2N} b \cdot \sum_{r=1}^{b} \sum_{l=1}^{N} \sum_{k=1}^{N-1} \sum_{t=k+1}^{N} \pi_t p_{t,k} \sum_{j=d_0+d_1+\ldots+d_{r-1}}^{d_1+\ldots+d_r} t'_{j,l} \sum_{i=1}^{j} t_{k,i}$$

$$+ \sum_{r=b+1}^{q} p_{i_r,i_r}^{s-2N} \cdot \sum_{l=1}^{N} \sum_{k=1}^{N-1} \sum_{t=k+1}^{N} \pi_t p_{t,k} \sum_{j=d_0+d_1+\ldots+d_{r-1}}^{d_1+\ldots+d_r} t'_{j,l} \sum_{i=1}^{j} t_{k,i} \Big) (1 + o(1))$$

$$\to Q \cdot b \cdot \sum_{r=1}^{b} \sum_{l=1}^{N} \sum_{k=1}^{N-1} \sum_{t=k+1}^{N} \pi_t p_{t,k} \sum_{j=d_0+d_1+\ldots+d_{r-1}}^{d_1+\ldots+d_r} t'_{j,l} \sum_{i=1}^{j} t_{k,i} \in (0, \infty)$$

then $\lambda \in (0, \infty)$.

In the definition of Γ_i we choose r such that $r \,/\, s \to 1$ under $s \to \infty$ (for example, let $r = s$). Therefore using the estimate (11) and the inequality $\varrho < 1$, we obtain

$$\begin{aligned} & d\Big(\mathcal{L}\big(\xi_n(s)\big), Pois(\lambda)\Big) \\ & \quad \le \frac{\left(1 - e^{-\lambda}\right)(2s + 2r + 3)\,\lambda}{n} + 2\lambda\varrho^{r+1}\left(2 + 2\varrho^{s+r+1} + \varrho^{2s+r+1}\right) \\ & \quad \le \frac{\left(1 - e^{-\lambda'}\right)(2s + 2r + 3)\,\lambda'}{n} + 10\lambda'\varrho^{r+1} \\ & \quad = O\left((s + r)\,p^{s-2N}\,(s-1)^{2N-1}\right) = o(1)\,. \end{aligned}$$

The proof of Theorem 2 is complete.

References

[1] Barbour A.D., Holst L., Janson S. Poisson approximation, volume 2 of Oxford Studies in Probability. The Clarendon Press Oxford University Press, New York, 1992. Oxford Science Publication.

[2] Billingsley P. Convergence of probability Measures. Moscow: Science, 1977, p.306.

[3] Chryssaphinou O., Papastavridis S., Vaggelatou E. Poisson limit theorems for the appearances of attributes.In: Steins method and applications. /Ed. by A.D.Barbour, L. H. Y. Chen. Singapore: Singapore Univ. Press, 2005, p. 1935.

[4] Goncharov V.L. From combinatorial analysis. Proc. AS USSR. Ser. math., 1944, v.8, N.1, p.3-48.

[5] Horn R.A., Johnson C.R. Matrix Analysis. World, 1989 , 656 p.

[6] Lindvall T. Lectures on the Coupling Method, Wiley, New York, 1992, p.96.

[7] Mezhennaya N.M., Multinomial normal theorem for number of monotone tuples with fixed length in a random equiprobable sequence. Review of Applied and Industrial Mathematics, 2007, v.14, N.3, p.503-505

[8] Minakov A.A. Compound Poisson approximation for the distribution of the number of monotone tuples in random sequence. Applied discrete mathematics. N.2 (28), 2015, p. 21-29.

[9] Pittel B.G. Limiting behavior of a process of runs, Ann. Prob., 9, 1981, p.119-129.

[10] Wolfowitz J. Asymptotics distribution of runs up and down. Ann. Math. Statist. 1944. v.15, p.163-172.

4 Appendix

Proof Lemma 1.

First we calculate components of vector (6) and then we substitute into (8).

Using (2) and properties of Jordan matrices the following equalities is true:

$$\mathbb{M}^{s-1} = \mathbb{T} \cdot \mathbb{J}^{s-1} \cdot \mathbb{T}^{-1}$$

and

$$\mathbb{J}^{s-1} = \mathrm{Diag}\left\{ \mathbb{J}_{d_1}^{s-1}\left(p_{i_1,i_1}\right), \mathbb{J}_{d_2}^{s-1}\left(p_{i_2,i_2}\right), \ \ldots, \mathbb{J}_{d_q}^{s-1}\left(p_{i_q,i_q}\right) \right\}.$$

For brevity denote $\mathbb{J}^{s-1}$ by $\mathbb{Y}$. Then multiply $\mathbb{Y}$ by $\mathbb{T}$, and the result denote by $\mathbb{F}$:

$$\mathbb{T} \cdot \mathbb{Y} = \left(f_{k,l}\right)_{(k,l)\in\mathcal{A}\times\mathcal{A}} \equiv \mathbb{F}.$$

Taking into account that $\mathbb{Y}$ is upper triangular matrix, we get the expression for (k,l)-th element of $\mathbb{F}$:

$$f_{k,l} = \sum_{i=1}^{l} t_{k,i} y_{i,l}.$$

Now multiplying $\mathbb{F}$ by $\mathbb{T}^{-1}$ we obtain the matrix $\mathbb{M}^{s-1}$:

$$\mathbb{M}^{s-1} = \mathbb{F} \cdot \mathbb{T}^{-1}.$$

Denote $\mathbb{M}^{s-1}$ by $\mathbb{W}$. Using fact that $\mathbb{W}$ is upper triangular matrix, we get (k,l)-th element of $\mathbb{W}$ under conditions $k \leq l$:

$$w_{k,l} = \sum_{j=1}^{N} t'_{j,l} \sum_{i=1}^{j} t_{k,i} y_{i,j}, \tag{23}$$

and under conditions $k > l$:

$$w_{k,l} = 0.$$

Now let's find the formula for each Jordan block of the matrix $\mathbb{Y}$. Suppose r-th Jordan block in degree $s-1$:

$$\mathbb{J}_{d_r}^{s-1}(p_{i_r,i_r}) = (p_{i_r,i_r}\mathbb{E} + \mathbb{H})^{s-1}, \quad r \in \{1,\ldots,q\},$$

where $\mathbb{E}_{d_r}$ is the identity matrix of size $d_r \times d_r$, and $\mathbb{H}_{d_r}$ is the matrix of size $d_r \times d_r$ such that superdiagonal of $\mathbb{H}_{d_r}$ consists of 1 and the other entries are zeros:

$$\mathbb{H} = \begin{pmatrix} 0 & 1 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 1 & \ldots & 0 & 0 \\ & & & \ddots & & \\ 0 & 0 & 0 & \ldots & 0 & 1 \\ 0 & 0 & 0 & \ldots & 0 & 0 \end{pmatrix},$$

It follows from [5, § 3.2.5] that:

$$\mathbb{J}_{d_r}^{s-1}(p_{i_r,i_r}) = \sum_{h=s-d_r}^{s-1} \binom{s-1}{h} p_{i_r,i_r}^{h} \mathbb{H}^{s-h-1}.$$

Therefore the (i,j)-th $\big(i,j \in \{1,\ldots,d_r\}\big)$ element of $\mathbb{J}_{d_r}^{s-1}(p_{i_r,i_r})$ equals to $p_{i_r,i_r}^{s-j+i-1}\binom{s-1}{s-j+i-1}$. Notice that the matrix $\mathbb{Y}$ has block structure, we

substitute the formula for $y_{i,j}$ into (23). Under condition $k \le l$ we have:

$$w_{k,l} = \sum_{j=d_0}^{d_1} t'_{j,l} \sum_{i=1}^{j} t_{k,i} y_{i,j} + \sum_{j=d_0+d_1}^{d_1+d_2} t'_{j,l} \sum_{i=1}^{j} t_{k,i} y_{i,j} + \dots$$

$$+ \sum_{j=d_0+d_1+\dots+d_{q-1}}^{d_1+\dots+d_q} t'_{j,l} \sum_{i=1}^{j} t_{k,i} y_{i,j}$$

$$= \sum_{r=1}^{q} \sum_{j=d_0+d_1+\dots+d_{r-1}}^{d_1+\dots+d_r} t'_{j,l} \sum_{i=1}^{j} t_{k,i} y_{i,j}$$

$$= \sum_{r=1}^{q} \sum_{j=d_0+d_1+\dots+d_{r-1}}^{d_1+\dots+d_r} t'_{j,l} \sum_{i=1}^{j} t_{k,i} p_{i_r,i_r}^{s-j+i-d_0-d_1-\dots-d_{r-1}} \binom{s-1}{s-j+i-d_0-d_1-\dots-d_{r-1}},$$

where $d_0 \equiv 1$.

Now we multiply the vector $\vec{\pi}$ by $\mathbb{K}$ is defined in (5):

$$\vec{\pi} \cdot \mathbb{K} = \left(\sum_{l=2}^{N} \pi_l p_{l,1}, \sum_{l=3}^{N} \pi_l p_{l,2}, \dots, \pi_N p_{N,N-1}, 0 \right).$$

And multiply the vector $\vec{\pi}\mathbb{K}$ by $\mathbb{W}$:

$$\vec{\alpha} = \vec{\pi}\mathbb{K}\mathbb{W} = \sum_{k=1}^{N-1} \vec{\mathbb{W}}_k \sum_{l=k+1}^{N} \pi_l p_{l,k}.$$

Now let us write the formula for the entries of $\vec{\alpha}$:

$$\alpha_l = \sum_{k=1}^{N-1} w_{k,l} \sum_{t=k+1}^{N} \pi_t p_{t,k}, \tag{24}$$

where $l \in \{1, \dots, N\}$.

Using (8) and (24), we get:

$$\lambda = n \cdot \sum_{l=1}^{N} \sum_{k=1}^{N-1} w_{k,l} \sum_{t=k+1}^{N} \pi_t p_{t,k}.$$

This completes the proof of Lemma 1.

Limit theorem for the image size of a subset under compositions of random mappings

Andrey Zubkov, Alexandr Serov

Abstract

Let $\mathcal{X}_N$ be a set consisting of N elements and $F_1, F_2, \ldots$ be a sequence of random independent equiprobable mappings $\mathcal{X}_N \to \mathcal{X}_N$. For a subset $S_0 \subset \mathcal{X}_N$, $|S_0| = n$, we consider a sequence of its images $S_t = F_t(\ldots F_2(F_1(S_0))\ldots)$, $t = 1, 2\ldots$ The conditions on n, t, $N \to \infty$ under which the distribution of image sizes $|S_t|$ is asymptotically normal are presented.

Keywords: random equiprobable mappings, compositions of random mappings, image sizes

1 Introduction

One of the well-known computationally hard problem is the search for solution of the equation

$$G(x) = a, \tag{1}$$

where G is a mapping of the finite set $\mathcal{X}_N = \{X_1, \ldots, X_N\}$ to itself such that all known methods of computation of the value $G^{-1}(a)$ have complexity comparable with the exhaustive search over the entire set $\mathcal{X}_N$, i. e. with $O(N)$ as $N \to \infty$.

M. E. Hellman [2] had proposed the universal (independent on the type of the function G) method for searching the solutions of the equation (1) permitting (after the preliminary construction of tables having volume smaller than $O(N)$ in time $O(N)$) to find a solution of equation (1) with a high probability in time of the order smaller than $O(N)$. This approach had been called the time-memory tradeoff. Let $R\colon \mathcal{X}_N \to \mathcal{X}_N$ be some one-to-one mapping and $F(x) = R(G(x))$, $x \in \mathcal{X}_N$. At the preliminary stage

of the Hellman method and its later modifications the tables containing in total $O(N/t)$ pairs of the form $(x, F^t(x))$, where $F^t(x)$ is the t-fold iteration of the mapping $F(x) = R(G(x))$. These tables allow at the main stage to find the solution of the equation (1) for any $a \in \mathcal{X}_{\mathcal{N}}$ by the computation of $O(N/n)$ values of $R(G(x))$. If n and t are of the order $O(N^{1/3})$, then the size of tables constructed on the preliminary stage has the order $O(N^{2/3})$, and at the main stage the search of a solution is performed by $O(N^{2/3})$ computations of values $R(G(x))$ (all these estimates are given up to the logarithmic factors).

We consider the version of the time-memory tradeoff method which is called "rainbow" table method. This method was proposed in [3], and its simplified mathematical model is as follows: an initial subset $S_0 \subset \mathcal{X}_{\mathcal{N}}$, $|S_0| = n$, is chosen and its images

$$S_1 = F_1(S_0),\ S_2 = F_2(F_1(S_0)), \ldots, S_t = F_t(F_{t-1}(\ldots(F_1(S_0))\ldots)), \qquad (2)$$

are calculated, where $F_1, \ldots, F_t$ are independent random mappings of the set $\mathcal{X}_{\mathcal{N}}$ into itself having uniform distribution on the set Σ_N, $|\Sigma_N| = N^N$, of all mappings $\mathcal{X}_{\mathcal{N}} \to \mathcal{X}_{\mathcal{N}}$. Obviously, the sequences $\{S_t\}$ and $\{|S_t|\}$ are Markov chains with non-increasing trajectories.

2 Main results

Assertion 1. *If the images of the initial subset $S_0 \subset \mathcal{X}_N$, $|S_0| = n$, are calculated according to the formulas (2), then the following identities are true:*

$$\mathbf{P}\left\{|S_t| = n \,\middle|\, |S_0| = n\right\} = \left(\prod_{q=1}^{n-1}\left(1 - \frac{q}{N}\right)\right)^t, \qquad (3)$$

$$\mathbf{P}\left\{|S_t| = n-1 \,\middle|\, |S_0| = n\right\} = \frac{n}{2}\left(1 - \left(1 - \frac{n-1}{N}\right)^t\right)\left(\prod_{q=1}^{n-2}\left(1 - \frac{q}{N}\right)\right)^t. \qquad (4)$$

Assertion 2. *If $n = CN^{1/3}$, then*

$$1 - \frac{C^2}{2N^{1/3}} \le \mathbf{P}\left\{|S_i| = n \,\middle|\, |S_{i-1}| = n\right\} \le e^{-\frac{n(n-1)}{2N}} < \\ < 1 - \frac{C^2}{2N^{1/3}} + \frac{C}{2N^{1/3}} \cdot \frac{C^3+4}{4N^{1/3}}\,, \quad (5)$$

$$\frac{C^2}{2N^{1/3}}\left(1 - \frac{C^2+2/C}{2N^{1/3}}\right) \le \mathbf{P}\left\{|S_i| = n-1 \,\middle|\, |S_{i-1}| = n\right\} \le \\ \le \frac{C^2}{2N^{1/3}}\left(1 - \frac{C^2}{2N^{1/3}} + \frac{C(C^3+16)}{8N^{2/3}} - \frac{C^3}{2N}\right)\,, \quad (6)$$

$$\mathbf{P}\left\{|S_i| < n-1 \,\middle|\, |S_{i-1}| = n\right\} \le \frac{C\left(C^3+2\right)}{4N^{2/3}}\,. \quad (7)$$

Let

$$p_0(n) = \mathbf{P}\left\{|S_1| = n \,\middle|\, |S_0| = n\right\}, \quad p_1(n) = \mathbf{P}\left\{|S_1| = n-1 \,\middle|\, |S_0| = n\right\}, \\ p_2(n) = \mathbf{P}\left\{|S_1| < n-1 \,\middle|\, |S_0| = n\right\}. \quad (8)$$

Assertion 3. *If $n = CN^{1/3}$, then*

$$1 - p_0(2) < 1 - p_0(3) < \ldots < 1 - p_0(n) \leqslant \frac{C^2}{2N^{1/3}}\,, \quad (9)$$

$$\frac{C\left(C^3+2\right)}{4N^{2/3}} \ge p_2(n) > p_2(n-1) > \ldots > p_2(3) > p_2(2). \quad (10)$$

Theorem 1. *If n, m, t, $N \to \infty$ in such a way that n has the order $N^{1/4}$ and $m = o(n)$, then for any fixed $x \in \mathbb{R}$ and*

$$t = 2N\left(\frac{1}{m} - \frac{1}{n}\right) + (1+o(1))x\,\frac{2N}{\sqrt{3}}\sqrt{\left(\frac{1}{m^3} - \frac{1}{n^3}\right)},$$

the following relation is true:

$$\mathbf{P}\left\{|S_t| \le \frac{2n\,N}{n\,t+2N}\right\} \to \Phi(x)\,,$$

where $\Phi(x)$ is the standard normal distribution function.

3 Proofs

Proof of assertion 1. Since the sequence $|S_0|, |S_1|, \ldots$ is non-increasing, then

$$\{|S_t| = n,\, |S_0| = n\} = \bigcap_{i=0}^{t-1} \{|S_{i+1}| = n,\, |S_i| = n\}\ .$$

As $\mathbf{P}\left\{|S_{i+1}| = n \,\middle|\, |S_i| = n\right\} = \prod\limits_{q=1}^{n-1} \left(1 - \frac{q}{N}\right)$ for any $i = 0, 1, \ldots$, then (3) follows from the independence of random mappings $F_1, \ldots, F_t$.

In order to prove the equality (4) we consider the random variable $\tau_n = \min\{j\colon |S_j| < n\}$ supposing that $\{|S_0| = n\}$. According to the total probability law, taking into account that under condition $\{|S_0| = n\}$ the events $\{\tau_n = i\}$, $i = 1, 2 \ldots$, are incompatible and the sequence $|S_i|$ is non-increasing, we find:

$$\mathbf{P}\left\{|S_t| = n-1 \,\middle|\, |S_0| = n\right\} = \sum_{i=1}^{t} \mathbf{P}\left\{\tau_n = i,\, |S_t| = n-1 \,\middle|\, |S_0| = n\right\}\ . \tag{11}$$

Due to the Markov structure of the sequence $|S_t|$ the items in (11) have the form

$$\begin{aligned}\mathbf{P}\left\{\tau_n = i,\, |S_t| = n-1 \,\middle|\, |S_0| = n\right\} &= \mathbf{P}\left\{|S_{i-1}| = n \,\middle|\, |S_0| = n\right\} \times \\ &\times \mathbf{P}\left\{|S_i| = n-1 \,\middle|\, |S_{i-1}| = n\right\} \mathbf{P}\left\{|S_t| = n-1 \,\middle|\, |S_i| = n-1\right\}. \end{aligned} \tag{12}$$

The first and the last factors may be calculated by the formula (3).

To find $\mathbf{P}\left\{|S_i| = n-1 \,\middle|\, |S_{i-1}| = n\right\}$ we will assume without loss of generality that $S_{i-1} = \{1, \ldots, n\}$. Then

$$\begin{aligned}&\mathbf{P}\left\{|S_i| = n-1 \,|\, |S_{i-1}| = n\right\} = \mathbf{P}\big\{|F_i(\{1, \ldots, n\})| = n-1\big\} = \\ &= \mathbf{P}\left\{\bigcup_{1 \leqslant k < j \leqslant n} \{F_i(k) = F_i(j),\, F_i(u) \neq F_i(v),\, 1 \leq u < v \leq n,\, (u, v) \neq (k, j)\}\right\}.\end{aligned}$$

Here we have a union of C_n^2 incompatible events, the probabilities of these events do not depend on k and j. Further,

$$\mathbf{P}\{F_i(1) = F_i(2),\ |\{F_i(2), \dots, F_i(n)\}| = n-1\} = \frac{1}{N}\prod_{q=1}^{n-2}\left(1 - \frac{q}{N}\right),$$

therefore

$$\mathbf{P}\left\{|S_i| = n-1 \,\middle|\, |S_{i-1}| = n\right\} = \frac{n(n-1)}{2N}\prod_{q=1}^{n-2}\left(1 - \frac{q}{N}\right). \tag{13}$$

Then, according to (11), (12) and (13),

$$\begin{aligned}
&\mathbf{P}\left\{|S_t| = n-1 \,\middle|\, |S_0| = n\right\} = \\
&= \sum_{i=1}^{t}\left(\prod_{q=1}^{n-1}\left(1 - \frac{q}{N}\right)\right)^{i-1}\left(\prod_{q=1}^{n-2}\left(1 - \frac{q}{N}\right)\right)\frac{n(n-1)}{2N}\left(\prod_{q=1}^{n-2}\left(1 - \frac{q}{N}\right)\right)^{t-i} = \\
&= \frac{n(n-1)}{2N}\left(\prod_{q=1}^{n-2}\left(1 - \frac{q}{N}\right)\right)^{t}\sum_{i=1}^{t}\left(1 - \frac{n-1}{N}\right)^{i-1} = \\
&= \frac{n}{2}\left(\prod_{q=1}^{n-2}\left(1 - \frac{q}{N}\right)\right)^{t}\left(1 - \left(1 - \frac{n-1}{N}\right)^{t}\right).
\end{aligned}$$

Thus the equality (4) is proved.

□

Proof of assertion 2. To prove two-sided estimate (5) we use inequalities

$$\begin{aligned}
&1 - \sum\nolimits_{i=1}^{k} x_i \leqslant \prod_{i=1}^{k}(1 - x_i) \leqslant \exp\left\{-\sum\nolimits_{i=1}^{k} x_i\right\} \leqslant \\
&\leqslant 1 - \sum\nolimits_{i=1}^{k} x_i + \frac{1}{2}\left(\sum\nolimits_{i=1}^{k} x_i\right)^2, \quad x_1, \dots, x_k \in [0, 1),
\end{aligned} \tag{14}$$

and equalities (3), (4) for $t = 1$:

$$\mathbf{P}\left\{|S_i| = n \,\middle|\, |S_{i-1}| = n\right\} = \prod_{q=1}^{n-1}\left(1 - \frac{q}{N}\right), \tag{15}$$

$$\mathbf{P}\left\{|S_i| = n-1 \,\middle|\, |S_{i-1}| = n\right\} = \frac{n(n-1)}{2N}\prod_{q=1}^{n-2}\left(1 - \frac{q}{N}\right). \tag{16}$$

Then it follows from (14) and (15) that

$$\mathbf{P}\left\{|S_i| = n \,\middle|\, |S_{i-1}| = n\right\} \geq 1 - \sum_{q=1}^{n-1}\frac{q}{N} = 1 - \frac{n(n-1)}{2N} > 1 - \frac{C^2}{2N^{1/3}} \tag{17}$$

and

$$\mathbf{P}\left\{|S_i| = n \,\middle|\, |S_{i-1}| = n\right\} \leq \exp\left\{-\sum_{q=1}^{n-1}\frac{q}{N}\right\} = \exp\left\{-\frac{n(n-1)}{2N}\right\} <$$
$$< 1 - \frac{(n-1)n}{2N} + \frac{(n-1)^2n^2}{8N^2} < 1 - \frac{C^2}{2N^{1/3}} + \frac{C}{2N^{2/3}} + \frac{C^4}{8N^{2/3}}.$$

Thus, two-sided inequality (5) is proved.

According to (16) and (14)

$$\mathbf{P}\left\{|S_i| = n-1 \,\middle|\, |S_{i-1}| = n\right\} \geq \frac{n(n-1)}{2N}\left(1 - \sum_{q=1}^{n-2}\frac{q}{N}\right) =$$
$$= \frac{n(n-1)}{2N} - \frac{n(n-1)^2(n-2)}{4N^2} > \frac{n^2}{2N} - \frac{n}{2N} - \frac{n^4}{4N^2} = \tag{18}$$
$$= \frac{C^2}{2N^{1/3}} - \frac{C(2+C^3)}{4N^{2/3}}.$$

Again using (16) and (14), we get the upper estimate:

$$\mathbf{P}\left\{|S_i| = n-1 \,\middle|\, |S_{i-1}| = n\right\} \leq \frac{n(n-1)}{2N}\, e^{-\frac{1}{N}\sum\limits_{q=1}^{n-2} q} = \frac{n(n-1)}{2N}\, e^{-\frac{(n-1)(n-2)}{2N}} <$$
$$< \frac{n(n-1)}{2N}\left(1 - \frac{(n-1)(n-2)}{2N}\left(1 - \frac{(n-1)(n-2)}{4N}\right)\right).$$

For $n = CN^{1/3}$ we have

$$\frac{(n-1)(n-2)}{N} > \frac{(n-2)^2}{N} > \frac{n^2}{N} - \frac{4n}{N} = C^2 N^{-1/3} - 4CN^{-2/3}$$

and

$$\frac{(n-1)(n-2)}{N} < \frac{n^2}{N} = C^2 N^{-1/3},$$

so

$$\begin{aligned}\mathbf{P}\left\{|S_i| = n-1 \,\middle|\, |S_{i-1}| = n\right\} \leq \\ \leq \frac{C^2}{2N^{1/3}}\left(1 - \left(\frac{C^2}{2N^{1/3}} - \frac{2C}{N^{2/3}}\right)\left(1 - \frac{C^2}{4N^{1/3}}\right)\right) = \\ = \frac{C^2}{2N^{1/3}}\left(1 - \frac{C^2}{2N^{1/3}} - \frac{C^3}{2N} + \frac{2C}{N^{2/3}} + \frac{C^4}{8N^{2/3}}\right).\end{aligned}$$

This proves the inequality (6).

The last inequality follows from (5), (6) and total probability law:

$$\begin{aligned}\mathbf{P}\left\{|S_i| = n \,\middle|\, |S_{i-1}| = n\right\} + \mathbf{P}\left\{|S_i| = n-1 \,\middle|\, |S_{i-1}| = n\right\} + \\ + \mathbf{P}\left\{|S_i| < n-1 \,\middle|\, |S_{i-1}| = n\right\} = 1\,.\end{aligned}$$

□

Proof of assertion 3. Indeed, for any $i = 2, 3, \ldots, n-1$ we have according to (3)

$$1 - p_0(i) = 1 - \prod_{q=1}^{i-1}\left(1 - \frac{q}{N}\right) < 1 - \prod_{q=1}^{i}\left(1 - \frac{q}{N}\right) = 1 - p_0(i+1)\,.$$

For $n = CN^{1/3}$ the last inequality in (9) follows from (5).

The first inequality in (10) coincides with the inequality (7) of the assertion 2. Further, since $p_2(k) = 1 - p_0(k) - p_1(k)$ for any $k = n, n-1, \ldots, 3$,

we have

$$p_2(k) - p_2(k-1) = p_0(k-1) - p_0(k) + p_1(k-1) - p_1(k) =$$
$$= \frac{k-1}{N} \prod_{q=1}^{k-2} \left(1 - \frac{q}{N}\right) + \frac{(k-1)}{2N} \left(\frac{k(k-2)}{N} - 2\right) \prod_{q=1}^{k-3} \left(1 - \frac{q}{N}\right) =$$
$$= \frac{(k-1)(k-2)}{N^2} \left(\frac{k}{2} - 1\right) \prod_{q=1}^{k-3} \left(1 - \frac{q}{N}\right) > 0.$$

That completes the proof of the assertion 3.

□

Proof of theorem 1. Consider the event

$$A_{n,t} = \left\{ |S_0| = n, \bigcap_{k=0}^{t-1} \{|S_{k+1}| \geq |S_k| - 1\} \right\}.$$

From (17) and (18) it follows, for example, that

$$\mathbf{P}\{A_{n,t}\} > (1 - p_2(n))^t = (p_0(n) + p_1(n))^t >$$
$$> \left(1 - \frac{n(n-1)^2(n-2)}{2N^2}\right)^t > 1 - \frac{t\, n(n-1)^2(n-2)}{2N^2} > 1 - \frac{tn^4}{2N^2}$$

for $t\, n^4 \leq 2N^2$. Thus, if $n, t, N \to \infty$, $n < CN^{1/4}$, $t = o(N)$, then $\mathbf{P}\{A_{n,t}\} \to 1$.

Consider the auxiliary Markov chain $\{S_k^*\}_{k=0}^{\infty}$ with $S_0^* = |S_0| = n$ and transition probabilities

$$\mathbf{P}\{S_{k+1}^* = j \,|\, S_k^* = j\} = p_0(j),$$
$$\mathbf{P}\{S_{k+1}^* = j-1 \,|\, S_k^* = j\} = 1 - p_0(j) = \mathbf{P}\{|S_{k+1}| \leq j-1 \,|\, |S_k| = j\} \geq$$
$$\geq \mathbf{P}\{|S_{k+1}| = j-1 \,|\, |S_k| = j\},\ j = 2, \ldots, n.$$

So, for any nonincreasing sequence $n_0 = n \geq n_1 \geq \ldots \geq n_t \geq 1$ such that $\max_{0 \leq k < t}(n_k - n_{k+1}) \leq 1$ we have

$$\mathbf{P}\{S_k^* = n_k\,(1 \leq k \leq t) \,|\, S_0^* = n\} \geq \mathbf{P}\{|S_k| = n_k\,(1 \leq k \leq t) \,|\, |S_0| = n\},$$

and $\mathbf{P}\{S_k^* = n_k\,(1 \le k \le t) \,|\, S_0^* = n\} = 0$ otherwise. Thus

$$\sum_{n_0=n\ge n_1\ge\ldots\ge n_t\ge 1} |\mathbf{P}\{S_k^* = n_k\,(1 \le k \le t) \,|\, S_0^* = n\} - \\ - \mathbf{P}\{|S_k| = n_k\,(1 \le k \le t) \,|\, |S_0| = n\}| = \\ = 2\mathbf{P}\left\{\max_{0\le k<t}(|S_k| - |S_{k+1}|) > 1\right\} = 2(1 - \mathbf{P}\{A_{n,t}\}),$$

that is if n, t and N tend to ∞ in such a way that $\mathbf{P}\{A_{n,t}\} \to 1$, then the total variation distance between the distributions of trajectories of Markov chains $\{|S_k|\}_{k=0}^t$ and $\{S_k^*\}_{k=0}^t$ tends to 0. Consequently, the total variation distance between the distributions of any functions of these trajectories tends to 0. Therefore in order to prove Theorem 1 we may consider the chain $\{S_k^*\}_{k=0}^t$ instead of the chain $\{|S_k|\}_{k=0}^t$.

Further, from the Markov property of the sequence $\{S_j^*\}$ it follows that the random variables $T_m = \min\{k \geqslant 1\colon S_k^* = m\}$, $m = 1, \ldots, n$, are defined correctly, the differences $\delta_m = T_{m-1} - T_m$, $m = 2, 3, \ldots, n$, are independent and according to the definition of the chain $\{S_k^*\}$ have the geometric distribution

$$\mathbf{P}\{\delta_j = k\} = \mathbf{P}\{S_k^* = j-1, S_{k-1}^* = j \,\big|\, S_0^* = j\} = \lambda_j^{k-1}(1-\lambda_j),$$

where in view of (14)

$$\lambda_j = \prod_{v=1}^{j-1}\left(1 - \frac{v}{N}\right) \in \left(1 - \tfrac{j(j-1)}{2N}\,, 1 - \tfrac{j(j-1)}{2N} + \tfrac{j^2(j-1)^2}{8N^2}\right),\ j = 1, 2, \ldots, n.$$

So,

$$\mathbf{E}\delta_j = \frac{1}{1-\lambda_j} \in \left(\frac{2N}{j(j-1)}\,, \frac{2N}{j(j-1)\left(1 - \frac{j(j-1)}{4N}\right)}\right), \tag{19}$$

$$\mathbf{D}\delta_j = \frac{\lambda_j}{(1-\lambda_j)^2} \in \left(\frac{4N^2(1 - \frac{j(j-1)}{2N})}{j^2(j-1)^2}\,, \frac{4N^2}{j^2(j-1)^2\left(1 - \frac{j(j-1)}{4N}\right)^2}\right). \tag{20}$$

Since $\mathbf{E}(\delta_j - \mathbf{E}\delta_j)^4 = \mathbf{E}\delta_j^4 - 4\,\mathbf{E}\delta_j^3\mathbf{E}\delta_j + 6\,\mathbf{E}\delta_j^2\,(\mathbf{E}\delta_j)^2 - 3\,(\mathbf{E}\delta_j)^4$ and the moment-generating function of δ_j has the form

$$g_{\delta_j}(e^z) = \mathbf{E}e^{z\delta_j} = \sum_{v=1}^{\infty} e^{zv}\lambda_j^{v-1}(1-\lambda_j) = \frac{e^z\,(1-\lambda_j)}{1-e^z\,\lambda_j}\,,$$

$$\mathbf{E}\delta_j^k = g_{\delta_j}^{(k)}(e^z)\,\big|_{z=0},\ k = 1, 2, \ldots,$$

we find that $\mathbf{E}\delta_j^3 = \frac{\lambda_j^2+4\lambda_j+1}{(1-\lambda_j)^3}$, $\mathbf{E}\delta_j^4 = \frac{(1+\lambda_j)(\lambda_j^2+10\lambda_j+1)}{(1-\lambda_j)^4}$ and

$$\mathbf{E}(\delta_j - \mathbf{E}\delta_j)^4 = \frac{\lambda_j(\lambda_j^2+7\lambda_j+1)}{(1-\lambda_j)^4} < \frac{9}{(1-\lambda_j)^4}\,.$$

Using the Lyapunov inequality we obtain an estimate of the third absolute central moment of δ_j

$$\mathbf{E}|\delta_j - \mathbf{E}\delta_j|^3 \leqslant (\mathbf{E}(\delta_j - \mathbf{E}\delta_j)^4)^{3/4} < \frac{3^{3/2}}{(1-\lambda_j)^3} < \frac{42N^3}{j^3(j-1)^3\left(1-\frac{j(j-1)}{4N}\right)^3}\,. \tag{21}$$

Thus, if $n, m, N \to \infty$ in such a way that n is of the order $N^{1/4}$ and $m = o(n)$, then for $T_m = \sum_{j=m+1}^{n}\delta_j$ we have

$$\begin{aligned}
\mathbf{E}T_m &= \sum_{j=m+1}^{n}\frac{1}{1-\lambda_j} = 2N\left(\frac{1}{m}-\frac{1}{n}\right)\left(1+O\left(\frac{n^2}{N}\right)\right),\\
\mathbf{D}T_m &= \sum_{j=m+1}^{n}\frac{\lambda_j}{(1-\lambda_j)^2} = \frac{4N^2}{3}\left(\frac{1}{m^3}-\frac{1}{n^3}\right)(1+o(1))\,, \qquad (22)\\
C_3(m,n) &= \sum_{j=m+1}^{n}\mathbf{E}|\delta_j - \mathbf{E}\delta_j|^3 < 10N^3\left(\frac{1}{m^5}-\frac{1}{n^5}\right).
\end{aligned}$$

Remark 1. The estimate of $\mathbf{E}T_m$ is the consequence of (19) and identities

$$\sum_{j=m+1}^{n}\frac{2N}{j(j-1)} = 2N\sum_{j=m+1}^{n}\left(\frac{1}{j-1}-\frac{1}{j}\right) = 2N\left(\frac{1}{m}-\frac{1}{n}\right),$$

$$1 \leq \frac{1}{1-\frac{j(j-1)}{4N}} \leq \frac{1}{1-\frac{n^2}{4N}} = 1+O\left(\frac{n^2}{4N}\right).$$

The estimate of $\mathbf{D}T_m$ is the consequence of (20), the identities

$$\sum_{j=m+1}^{n} \frac{4N^2(1-\frac{j(j-1)}{2N})}{j^2(j-1)^2} = \sum_{j=m+1}^{n} \left(\frac{4N^2}{j^2(j-1)^2} - \frac{2N}{j(j-1)} \right),$$

$$\sum_{j=m+1}^{n} \frac{4N^2}{j^2(j-1)^2\left(1-\frac{j(j-1)}{4N}\right)^2} = \sum_{j=m+1}^{n} \frac{4N^2\left(1+\frac{j(j-1)}{4N}\right)^2}{j^2(j-1)^2\left(1-\frac{j^2(j-1)^2}{16N^2}\right)^2}$$

and the inequalities

$$\sum_{j=m+1}^{n} \frac{4N^2}{j^2(j-1)^2} \geq \int_{m+\frac{1}{2}}^{n+\frac{1}{2}} \frac{4N^2}{j^4}\,dj = \frac{4N^2}{3}\left(\frac{1}{(m+\frac{1}{2})^3} - \frac{1}{(n+\frac{1}{2})^3} \right)$$
$$\geq \frac{4N^2}{3}\left(\frac{1}{(m+1)^3} - \frac{1}{n^3} \right),$$

$$\sum_{j=m+1}^{n} \frac{4N^2}{j^2(j-1)^2} \leq \int_{m+\frac{1}{2}}^{n+\frac{1}{2}} \frac{4N^2}{(j-1)^4}\,dj = \frac{4N^2}{3}\left(\frac{1}{(m-\frac{1}{2})^3} - \frac{1}{(n-\frac{1}{2})^3} \right)$$
$$\leq \frac{4N^2}{3}\left(\frac{1}{(m-1)^3} - \frac{1}{n^3} \right).$$

Finally, the estimate $C_3(m,n)$ is the consequence of (21) and

$$\sum_{j=m+1}^{n} \frac{42N^3}{j^3(j-1)^3} \leq \int_{m+\frac{1}{2}}^{n+\frac{1}{2}} \frac{42N^3}{(j-1)^6}\mathrm{d}j = \frac{42N^3}{5}\left(\frac{1}{(m-\frac{1}{2})^5} - \frac{1}{(n-\frac{1}{2})^5} \right)$$
$$\leq \frac{42N^3}{5}\left(\frac{1}{(m-1)^5} - \frac{1}{n^5} \right).$$

If $0 < \varepsilon < \frac{m}{n} < 1-\varepsilon$, then the Lyapunov ratio (see, for example, [1, p. 188])

$$\frac{C_3(m,n)}{(\mathbf{D}T_m)^{3/2}} = O(m^{3\cdot 3/2-5}) = O(m^{-1/2}) \tag{23}$$

tends to 0 as $N,n,m \to \infty$, $m = o(n)$. According to the Lyapunov theorem the distribution of T_m is asymptotically normal with parameters $(\mathbf{E}T_m, \mathbf{D}T_m)$.

The equalities $\{S_t^* \le m\} = \{T_m \le t\}$ allow to find the asymptotic behavior of distribution of S_t^* for $N, n \to \infty, m = o(n)$, since

$$\begin{aligned} \mathbf{P}\left\{\frac{T_m - \mathbf{E}T_m}{\sqrt{\mathbf{D}T_m}} \le x\right\} &= \mathbf{P}\left\{T_m \le \mathbf{E}T_m + x\sqrt{\mathbf{D}T_m}\right\} \\ &= \mathbf{P}\{S^*_{\mathbf{E}T_m + x\sqrt{\mathbf{D}T_m}} \le m\} \to \Phi(x), \end{aligned} \tag{24}$$

where $\Phi(x)$ is the standard normal distribution function.

Denote by

$$u(n, m, x) = \mathbf{E}T_m + x\sqrt{\mathbf{D}T_m} \tag{25}$$

(in what follows, for brevity, we shall simply write u). The right part of (25) is an increasing function of m. Let

$$m = f(u) \tag{26}$$

be the solution of the equation (25). Then

$$\mathbf{P}\{|S_u^*| \le f(u)\} \to \Phi(x).$$

So, it remains to find the required representation (26). To this aim we use in (25) the values of $\mathbf{E}T_m$, $\mathbf{D}T_m$ from (22), denoting the remainder terms in these representations by c_1, c_2 respectively:

$$u = 2N\left(\frac{1}{m} - \frac{1}{n}\right)(1 + c_1) + x\,\frac{2N}{\sqrt{3}}\sqrt{\left(\frac{1}{(m^3} - \frac{1}{n^3}\right)(1 + c_2)}\,.$$

Note that if $N, n, m \to \infty$, $m = o(n)$, then $N = o(un)$.

Sequential identity transformations of this formula give the cubic equation for the unknown $\frac{1}{m}$:

$$\left(\frac{1}{m}\right)^3 - \frac{3(1+c_1)^2}{x^2(1+c_2)}\left(\frac{1}{m}\right)^2 + \frac{1}{x^2}\left(\frac{6(1+c_1)^2}{(1+c_2)n} + \frac{3u(1+c_1)}{(1+c_2)N}\right)\left(\frac{1}{m}\right) - \frac{1}{n^3} - \frac{3}{(1+c_2)x^2}\left(\frac{u}{2N} + \frac{1+c_1}{n}\right)^2 = 0\,.$$

We are interested in a root of this equation satisfying condition $m \to \infty$, i. e. $\frac{1}{m} \to 0$ as $n, N \to \infty$. This equation may be represented in the

equivalent form

$$\frac{3}{x^2(1+c_2)}\left((1+c_1)\left(\frac{1}{m}\right)-\left(\frac{u}{2N}+\frac{1+c_1}{n}\right)\right)^2=\frac{1}{m^3}-\frac{1}{n^3}\,.$$

So, if $m,n\to\infty$, $m=o(n)$, then the solution should have the form

$$m=\frac{2nN(1+c_1)}{u\ n+2(1+c_1)N}(1+o(1)). \tag{27}$$

Here $N=o(un)$, so

$$m=\frac{2nN}{un+2N}\,(1+o(1)).$$

The theorem is proved.

□

References

[1] Borovkov A. A., *Probability Theory*, New York: Gordon & Breach, 1998, 474 pp.

[2] Hellman M.E., "A cryptanalytic time–memory trade-off", *IEEE Trans. Inf. Theory*, 1980, 401–406.

[3] Oechslin P., "Making a faster cryptanalytic time-memory trade-off", *Lect. Notes Comput. Sci.*, **2729** (2003), 617–630.

The permutation group insight on diffusion property of linear mappings

Dmitry Burov, Boris Pogorelov

Abstract

In this paper we investigate the properties of linear mappings related to the structures of the group, generated by s-box layer and the group of key addition layer, i.e. the translation group of a vector space. We propose new parameters which characterize diffusion properties of linear mapping. We give a new characterization of MDS linear mappings. Moreover MDS linear mappings which are used in connection with resistance to linear and differential methods can be derived from permutation group point of view. This fact shows that permutation group strategy is a general approach for design block cipher primitives.

Keywords: block cipher, linear mapping, wreath product, exponentiation, structures for permutation groups, systems of blocks, metrics.

1 Introduction

Linear and differential methods are the most well known methods of block cipher cryptanalysis. These methods had an influence on constructing of block ciphers. In particular, differential characteristic and linear characteristic are one of the main properties of s-boxes. Linear and differential branch numbers are the main characteristics of a linear mapping. Linear and differential branch numbers are associated with linear and differential methods.

Furthermore, we consider XSL block ciphers. The round function of XSL block ciphers consists of the following layers: key addition layer, nonlinear s-box layer, linear layer. Let G_{XS} be the group generated by s-box layer and the group of key addition layer, i.e. the translation group of a vector space, G_{XL} be the group generated by linear mapping and the group of key addition layer. We show that the group G_{XS} is a subgroup of the wreath product of symmetric permutation groups and a subgroup of exponentiation of the

symmetric permutation group [11], [6]. Moreover there exist G_{XS}-invariant partitions and G_{XS}-invariant metrics. Linear mapping is intended to diffuse these structures in order to avoid some attacks. G_{XL}-invariant metrics are studied by Pogorelov B.A. and Pudovkina M.A. [13]. In this paper we study the properties of a linear mapping from the point of view of diffusion of structures of the group G_{XS}. In particular we show that a linear mapping optimally diffuses the partitions if only if it is MDS mapping. Nevertheless even MDS mapping can be reducible. This leads to an imprimitivity of the group G_{XL}. The reducibility of a linear mapping are used in attacks on Khazad block cipher [2], [3]. Implicitly the reducibility of a linear mapping is used in attacks on Print [9], Zorro, Robin, iSCREAM [10], Midori [7].

2 Preliminaries

Recall some definitions. Let K and H be groups and suppose H acts on the nonempty set Ω. Denote by $\mathrm{Fun}(\Omega, K)$ the set of all functions from Ω into K. The set $\mathrm{Fun}(\Omega, K)$ with respect to a product:

$$(fg)(\gamma) = f(y)\, g(\gamma),\ \forall f, g \in \mathrm{Fun}(\Omega, K), \gamma \in \Omega$$

is a group. The wreath product of K by H with respect to this action is defined to be the semiderect product $\mathrm{Fun}(\Omega, K) \leftthreetimes H$ where H acts on the group $\mathrm{Fun}(\Omega, K)$ via

$$f^g(\gamma) = f\left(\gamma^{g^{-1}}\right), \quad \forall f \in \mathrm{Fun}(\Omega, K), \gamma \in \Omega, g \in H.$$

Denote this group by $K\ wr\ H$.

The wreath product $K\ wr\ H$ has two actions. Namely, if K acts on a set Δ, then we can define an action of $K\ wr\ H$ on $\Delta \times \Omega$ by

$$(\delta, \gamma)^{(f,u)} = \left(\delta^{f(\gamma)}, \gamma^u\right), \quad \forall (\delta, \gamma) \in \Delta \times \Omega,$$

where $(f, u) \in K\ wr\ H$. Denote the group $(K\ wr\ H, \Delta \times \Omega)$ by $K \wr H$.

Suppose K acts on a set Δ. Define the action of the group $K\ wr\ H$ on the set $\mathrm{Fun}(\Omega,\Delta)$ as follows:

$$\varphi(\gamma)^{(f,x)} = \varphi\Big(\gamma^{x^{-1}}\Big)^{f\left(\gamma^{x^{-1}}\right)}, \quad \forall \gamma \in \Omega.$$

Denote the group $(K \ wr \ H, \mathrm{Fun}\,(\Omega,\Delta))$ by $K \uparrow H$.

Denote the symmetric permutation group on a finite set Ω of n elements by $S_n = S\,(\Omega)$.

Let us do some remarks on the properties of these actions. Suppose

$$\mathbf{W}=\{W_1,\ldots,W_r\}$$

is a partition of the set Ω and $|W_i| = w,\ i = 1,\ldots,r$. Then the group $G = \{g \in S\,(\Omega)\,|\mathbf{W}^g = \mathbf{W}\}$ is equal to the group $S\,(W)\wr S_r$ (strictly speaking G is permutation isomorphic to $S\,(W)\wr S_r$).

Suppose Ω is equal A^n for any set A, $n \in \mathbb{N}$. Then the isometry group of Hamming metric on the set A^n is equal to the group $S_{|A|} \uparrow S_n$ (strictly speaking the isometry group is permutation isomorphic to $S_{|A|} \uparrow S_n$). For other properties of the wreath product actions we refer to [11], [6].

Denote by V_n the vector space of dimension n over the field $GF\,(2)$. Suppose $v_\alpha : x \mapsto x \oplus \alpha$ for all $x \in V_n$, where binary operation $\oplus$ is the operation of vector addition. Denote by V_n^+ the translation group of the vector space V_n, i.e.

$$V_n^+ = \{v_\alpha | \alpha \in V_n\}\,.$$

Let $s \in S\,(V_n)$ be a parallel action of substitutions $s_1,\ldots,s_m \in S\,(V_d)$, $n = md$, i.e.

$$s : (\alpha_1,\ldots,\alpha_m) \mapsto (\alpha_1^{s_1},\ldots,\alpha_m^{s_m})\,,\ \alpha_i \in V_d,\ i = 1,\ldots,m.$$

Mapping s is used in XSL ciphers as s-box layer.

Let $GL_n\,(2)$ be the group of invertible $(n \times n)$-matrixes. Each matrix $h \in GL_n\,(2)$ induces the bijective linear mapping of the vector space V_n by multiplying vector-rows by h. Denote this linear mapping also by h. Further we use special submatrixes of matrix h. Namely, suppose $n = md$, $t, r \in \{1,\ldots,m\}$, $1 \le i_1 < \cdots < i_t \le m$, $1 \le j_1 < \cdots < j_r \le m$. Let

$h\begin{pmatrix} i_1,\ldots,i_t \\ j_1,\ldots,j_r \end{pmatrix}$ be a $(td \times rd)$-submatrix of the matrix h which is obtained by deleting rows

$$\{1,\ldots,n\} \setminus \{(i_1-1)\,d+1,\ldots,(i_1-1)\,d+d,\ldots,(i_t-1)\,d+d\}$$

and columns

$$\{1,\ldots,n\} \setminus \{(j_1-1)\,d+1,\ldots,(j_1-1)\,d+d,\ldots,(j_r-1)\,d+d\}\,.$$

Denote the vector $(0,\ldots,0) \in V_d$ by 0_d. For $\alpha=(\alpha_1,\ldots,\alpha_m) \in V_n$, $\alpha_i \in V_d$, $i=1,\ldots,m$, suppose $\mathrm{wt}(\alpha) = |\{i \in \{1,\ldots,m\} \,|\alpha_i \neq 0_d\}|$. Usually diffusion property of linear mappings is characterized by the differential branch number $\mathrm{bn}_d(h)$ and linear branch number $\mathrm{bn}_l(h)$ [5] defined as

$$\mathrm{bn_d}(h) = \min\left\{\mathrm{wt}(\alpha) + \mathrm{wt}\left(\alpha^h\right) |\alpha \in V_n \setminus \{0\}\right\}\,,$$

$$\mathrm{bn}_l(h) = \min\left\{\mathrm{wt}(\alpha) + \mathrm{wt}\left(\alpha^{{}^th}\right) |\alpha \in V_n \setminus \{0\}\right\}\,.$$

Here th is a transpose matrix. It is easily to show that $\mathrm{bn}(h) \leq m+1$. If $\mathrm{bn_d}(h) = m+1$, then h is called MDS linear mapping. Besides if $\mathrm{bn_d}(h) = m+1$, then $\mathrm{bn}_l(h) = m+1$ [4]. The differential and linear branch numbers have influence on resistance of block ciphers with respect to differential and linear cryptanalysis, respectively.

3 Structures of the group $\mathbf{G_X}_S$

Recall that G_{XS} is equal to $\langle V_n^+, s\rangle$. Suppose

$$V(i_1,\ldots,i_t) = \left\{ \left(\overbrace{\underbrace{0_d,\ldots,0_d}_{i_1-1},\alpha_1,0_d,\ldots,0_d}^{i_t-1},\alpha_t,0_d,\ldots,0_d \right) |\alpha_1,\ldots,\alpha_t \in V_d \right\}.$$

Let $\mathbf{V}(i_1,\ldots,i_t)$ be a partition of cosets of $V(i_1,\ldots,i_t)$ in the vector space V_n. Since V_n^+ is a regular group and $V(i_1,\ldots,i_t) < V_n$, the partition $\mathbf{V}(i_1,\ldots,i_t)$ is a system of blocks for the group V_n^+. It is easily to show that

the partition $\mathbf{V}(i_1,\ldots,i_t)$ is s-invariant. Therefore $\mathbf{V}(i_1,\ldots,i_t)$ is system of blocks for the group G_{XS}. Maximal subgroup of symmetric group $S(V_n)$ preserving partition $\mathbf{V}(i_1,\ldots,i_t)$ is the wreath product $S(V(i_1,\ldots,i_t)) \wr S_{2^{n-dt}}$. Hence, G_{XS} is a subgroup of the wreath product $S(V(i_1,\ldots,i_t)) \wr S_{2^{n-dt}}$ for all $t = 1,\ldots,m-1$, $1 \le i_1 < \cdots < i_t \le m$. Therefore, we have the following proposition.

Proposition 1. *For all $m \ge 2$, $d \ge 2$, $1 \le i_1 < \cdots < i_t \le m$, $t = \{1,\ldots,m-1\}$ the group G_{XS} is imprimitive and $\mathbf{V}(i_1,\ldots,i_t)$ is a system of blocks for the group G_{XS}.*

Let $\mathbf{A} = \{A_1,\ldots,A_u\}$ be a partition of the set $\{1,\ldots,m\}$, $|A_i| = \frac{m}{u}$, $i = 1,\ldots,u$. Define a weight of the vector $\alpha = (\alpha_1,\ldots,\alpha_m) \in V_n$, $\alpha_i \in V_d$, $i = 1,\ldots,m$, with respect to partition $\mathbf{A}$ by

$$\mathrm{wt}_{\mathbf{A}}(\alpha) = |\{i \in \{1,\ldots,u\} \,|\exists j \in A_i,\ \alpha_j \ne 0\}|.$$

Note if $A_i = \{i\}$, $i = 1,\ldots,m$, then $\mathrm{wt}_{\mathbf{A}}(\alpha)$ is equal to $\mathrm{wt}(\alpha)$ for all $\alpha \in V_n$. Let $\chi_{\mathbf{A}} : V_n \times V_n \to \{0,\ldots,u+1\}$ be a metric defined as $\chi_{\mathbf{A}}(\alpha,\beta) = \mathrm{wt}_{\mathbf{A}}(\alpha \oplus \beta)$. The isometric group $G_{\mathbf{A}} < S(V_n)$ of metric $\chi_{\mathbf{A}}$ is permutation isomorphic to the exponentiation $S_{2^{\frac{n}{u}}} \uparrow S_u$. It is clearly that $\chi_{\mathbf{A}}(\alpha,\beta) = \chi(\alpha^s,\beta^s)$ and $\chi_{\mathbf{A}}(\alpha,\beta) = \chi_{\mathbf{A}}(\alpha \oplus \gamma, \beta \oplus \gamma)$ for all $\alpha,\ \beta,\gamma \in V_n$. Hence, we have the following proposition.

Proposition 2. *For all partitions $\mathbf{A} = \{A_1,\ldots,A_u\}$ of the set $\{1,\ldots,m\}$, $|A_i| = \frac{m}{u}$, $i = 1,\ldots,u$, the metric $\chi_{\mathbf{A}}$ is G_{XS}-invariant.*

For any $p \in S_m$ denote by $\hat{p} \in S(V_n)$ the action permutation p on V_n, i.e. for all $\alpha = (\alpha_1,\ldots,\alpha_m) \in V_n$, $\alpha_i \in V_d$, $i = 1,\ldots,\ m$, we have

$$\hat{p} : (\alpha_1,\ldots,\alpha_m) \mapsto \left(\alpha_{1^{p^{-1}}},\ldots,\alpha_{m^{p^{-1}}}\right).$$

Proposition 3. *Let $\mathbf{A} = \{A_1,\ldots,A_u\}$ be a partition of the set $\{1,\ldots,m\}$, $A_i = \left\{i_1,\ldots,i_{\frac{m}{u}}\right\} \subset \{1,\ldots,m\}$, $i = 1,\ldots,u$. Suppose g belongs to $G_{\mathbf{A}}$. Then there exists a permutation $p_g \in S_m$ such that $g\hat{p}_g \in S\left(V\left(i_1,\ldots,i_{\frac{m}{u}}\right)\right) \wr S_{2^{n-dt}}$.*

Proof. There exists $p \in S_m$ such that

$$A_i^p = \left\{(i-1)\frac{m}{u}+1, \ldots, (i-1)\frac{m}{u}+\frac{m}{u}\right\}.$$

Let $\mathbf{A}' = \{A'_1, \ldots, A'_u\}$ be a partition of the set $\{1, \ldots, m\}$ and $A'_i = A_i^p$. Hence, $\chi_{\mathbf{A}'}$ is the Hamming metric,

$\chi_{\mathbf{A}}(\alpha, \beta) = \chi_{\mathbf{A}'}\left(\alpha^{\hat{p}}, \beta^{\hat{p}}\right)$, $G_{\mathbf{A}} = \hat{p} G_{\mathbf{A}'} \hat{p}^{-1}$. Since $G_{\mathbf{A}'}$ is the group of isometric of the Hamming metric, for any $f \in G_{\mathbf{A}}$ there exist $f_1, \ldots, f_u \in S\left(V_{d^{\frac{m}{u}}}\right)$, $r \in S_u$ such that

$$f : (\beta_1, \ldots, \beta_u) \to \left(\beta_{1^{r^{-1}}}^{f_{1^{r^{-1}}}}, \ldots, \beta_{u^{r^{-1}}}^{f_{u^{r^{-1}}}}\right), \quad \beta_i \in V_{d^{\frac{m}{u}}}, i = 1, \ldots, u.$$

Hence, for any $f \in G_{\mathbf{A}'}$ there exist $f_1, \ldots, f_u \in S\left(V_{d^{\frac{m}{u}}}\right)$, $q \in S_m$ such that

$$f : (\beta_1, \ldots, \beta_u) \to \left(\beta_1^{f_1}, \ldots, \beta_u^{f_u}\right)^{\hat{q}}$$

and $f\hat{q}^{-1} \in \underbrace{S\left(V_{d^{\frac{m}{u}}}\right) \times \cdots \times S\left(V_{d^{\frac{m}{u}}}\right)}_{u \text{ times}} < S\left(V\left(1, \ldots, \frac{m}{u}\right)\right) \wr S_{2^{n-\frac{n}{u}}}$. Suppose $q' \in S_m$ such that $p^{-1}q' = qp^{-1}$.

Thus, for any $g \in G_{\mathbf{A}}$ there exist $f \in G_{\mathbf{A}'}$, $q, q' \in S_m$ such that

$$g\hat{q}' = \hat{p} f \hat{p}^{-1} \hat{q}' = \hat{p} f \hat{q} \hat{p}^{-1} \in \hat{p} S\left(V\left(1, \ldots, \tfrac{m}{u}\right)\right) \wr S_{2^{n-\frac{n}{u}}} \hat{p}^{-1} =$$
$$= S\left(V\left(i_1, \ldots, i_{\frac{m}{u}}\right)\right) \wr S_{2^{n-\frac{n}{u}}}$$

□

4 Diffusion properties of linear mapping

Since partitions $\mathbf{V}(i_1, \ldots, i_t)$ are G_{XS}-invariant, diffusion properties of linear mapping h are important. Suppose $\mathbf{W}{=}\mathbf{V}(i_1, \ldots, i_t)$, $\mathbf{W}'{=}\mathbf{V}(j_1, \ldots, j_r)$. Denote by $\rho(\mathbf{a}, \mathbf{b})$ the Euclid distance between real matrixes $\mathbf{a}, \mathbf{b} \in \mathbb{R}_{\mathbf{n}\times\mathbf{m}}$, i.e. $\rho(\mathbf{a}, \mathbf{b}) = \sqrt{\sum_{i,j}(a_{i,j} - b_{i,j})^2}$. Diffusion property of linear mapping h with respect to the pair of partitions $(\mathbf{W}, \mathbf{W}')$ is characterized by $\left(2^{(m-t)d} \times 2^{(m-r)d}\right)$-matrix $\mathbf{c}_{\mathbf{W},\mathbf{W}'}(h) = \|c_{i,j}(h)\|$, where

$$c_{i,j}(h) = \left|W_i^h \cap W'_j\right|.$$

More precisely, we study the Euclid distance between matrix $\mathbf{c}_{\mathbf{W},\mathbf{W}'}(h)$ and the uniform $\left(2^{(m-t)d}\times 2^{(m-r)d}\right)$-matrix $\left\|2^{(t+r-m)d}\right\|$. The Euclid distance is less the property linear mapping h is better. For example

$$\rho\left(\mathbf{c}_{\mathbf{W},\mathbf{W}}(h),\left\|2^{(t+r-m)d}\right\|\right)$$

is maximum, i.e. $2^{-td}\mathbf{c}_{\mathbf{W},\mathbf{W}}(h)$ is a substitution matrix, if and only if $h\in S(V(i_1,\ldots,i_t))\wr S_{2^{(m-t)d}}$, i.e. partition $\mathbf{W}$ is h-invariant. On the other hand, h has optimal diffusion property under pair of partitions $(\mathbf{W},\mathbf{W}')$ if

$$\rho\left(\mathbf{c}_{\mathbf{W},\mathbf{W}'}(h),\left\|2^{(t+r-m)d}\right\|\right)$$

is the least. In the next proposition the lower bound for

$$\rho\left(\mathbf{c}_{\mathbf{W},\mathbf{W}'}(h),\left\|2^{(t+r-m)d}\right\|\right)$$

is derived.

Proposition 4. *Let $h\in GL_n(2)$, $\mathbf{W}=\mathbf{V}(i_1,\ldots,i_t)$, $\mathbf{W}'=\mathbf{V}(j_1,\ldots,j_r)$, $1\leq i_1<\cdots<i_t\leq m$, $1\leq j_1<\cdots<j_r\leq m$, $n=md$, $m,d\geq 2$. Then we have*

$$\rho\left(\mathbf{c}_{\mathbf{W},\mathbf{W}'}(h),\left\|2^{(t+r-m)d}\right\|_{2^{(m-t)d},2^{(m-r)d}}\right)\geq\begin{cases}0, & if\ t+r\geq m,\\ \sqrt{2^{md}-2^{(t+r)d}}, & if\ t+r<m;\end{cases}$$

$$\mathbf{c}_{i,j}(h)\in\left\{0,2^{td-\mathrm{rang}\ h\begin{pmatrix}i_1,\ldots,i_t\\ j'_1,\ldots,j'_{m-r}\end{pmatrix}}\right\},$$

where $\{j'_1,\ldots,j'_{m-r}\}=\{1,\ldots,m\}\setminus\{j_1,\ldots,j_r\}$, $1\leq j'_1<\cdots<j'_{m-r}\leq m$.

Proof. By definition we have

$$c_{i,j}(h)=\left|W_i^h\cap W'_j\right|=\left|(V(i_1,\ldots,i_t)\oplus\delta)^h\cap(V(j_1,\ldots,j_r)\oplus\gamma)\right|$$

for appropriate $\delta = (\delta_1, \ldots, \delta_m)$, $\gamma = (\gamma_1, \ldots, \gamma_m) \in V_n$, $\delta_l, \gamma_l \in V_d$, $l = 1, \ldots, m$. Therefore $c_{i,j}(h)$ is equal to the number of solutions of the linear system

$$\alpha h \begin{pmatrix} i_1, \ldots, i_t \\ j'_1, \ldots, j'_{m-r} \end{pmatrix} = \left(\gamma_{j'_1}, \ldots, \gamma_{j'_{m-r}}\right).$$

Hence, $c_{i,j}(h) \in \left\{ 0, 2^{td - \operatorname{rang} h \begin{pmatrix} i_1, \ldots, i_t \\ j'_1, \ldots, j'_{m-r} \end{pmatrix}} \right\}$.

It is easily to show that $\rho\left(\mathbf{c}_{\mathbf{W},\mathbf{W}'}(h), \left\|2^{(t+r-m)d}\right\|\right)$ is minimal if and only if $\operatorname{rang} h \begin{pmatrix} i_1, \ldots, i_t \\ j'_1, \ldots, j'_{m-r} \end{pmatrix} = \min\{td, (m-r)d\}$.

Suppose $t + r \geq m$. If $\operatorname{rang} h\left(\, i_1, \ldots, i_t, j'_1, \ldots, j'_{m-r} \,\right) = (m-r)d$, then $c_{i,j}(h) = 2^{(t+r-m)d}$ for all i, j. Hence $\mathbf{c}_{\mathbf{W},\mathbf{W}'}(h) = \left\|2^{(t+r-m)d}\right\|$ and $\rho\left(\mathbf{c}_{\mathbf{W},\mathbf{W}'}(h), \left\|2^{(t+r-m)d}\right\|\right) = 0$.

Suppose $t + r < m$. Then we have

$$\left|\left\{(i,j) \in \left\{1, \ldots, 2^{(m-t)d}\right\} \times \left\{1, \ldots, 2^{(m-r)d}\right\} | c_{i,j}(h) = 1\right\}\right| = 2^{md},$$

$$\left|\left\{(i,j) \in \left\{1, \ldots, 2^{(m-t)d}\right\} \times \left\{1, \ldots, 2^{(m-r)d}\right\} | c_{i,j}(h) = 0\right\}\right| = 2^{(2m-(t+r))d} - 2^{md}.$$

Hence,

$$\rho\left(\mathbf{c}_{\mathbf{W},\mathbf{W}'}(h), \left\|2^{(t+r-m)d}\right\|\right) = \sqrt{2^{dt}\left(1 - 2^{(t+r-m)d}\right)^2 + \left(2^{d(m-r)} - 2^{dt}\right)\left(2^{(t+r-m)d}\right)^2} = \sqrt{2^{md} - 2^{(t+r)d}}.$$

□

Therefore $\rho\left(\mathbf{c}_{\mathbf{W},\mathbf{W}'}(h), \left\|2^{(t+r-m)d}\right\|_{2^{n-td}, 2^{n-tr}}\right)$ depends on rang of appropriate submatrix only.

Suppose

$$\varphi_m(t,r) = \begin{cases} 0, & if\ t + r \geq m, \\ \sqrt{2^{md} - 2^{(t+r)d}}, & if\ t + r < m. \end{cases}$$

Corollary 5. *The equalities*

$$\rho\left(\mathbf{c}_{\mathbf{R},\mathbf{R}'}(h), \left\|2^{(t+r-m)d}\right\|_{2^{(m-t)d},2^{(m-r)d}}\right) = \varphi_m(t,r)$$

hold for all $t, r \in \{1,\ldots,m\}$, $1 \le i_1 < \cdots < i_t \le m$, $1 \le j_1 < \cdots < j_r \le m$, *if and only if* h *is MDS linear mapping.*

Proof. It is know [1] that $h \in GL_n(2)$ is MDS linear mapping if and only if for all $t \in \{1,\ldots,m\}$, $1 \le i_1 < \cdots < i_t \le m$, $1 \le j_1 < \cdots < j_t \le m$ we have

$$\operatorname{rang} h\begin{pmatrix} i_1,\ldots,i_t \\ j_1,\ldots,j_t \end{pmatrix} = t.$$

Therefore h is MDS linear mapping if and only if for all $t, r \in \{1,\ldots,m\}$, $1 \le i_1 < \cdots < i_t \le m$, $1 \le j_1 < \cdots < j_r \le m$, we have

$$\operatorname{rang} h\begin{pmatrix} i_1,\ldots,i_t \\ j_1,\ldots,j_{m-r} \end{pmatrix} = \min\{td, (m-r)d\}\ .$$

Hence, h is MDS linear mapping if and only if the equalities

$$\rho\left(\mathbf{c}_{\mathbf{R},\mathbf{R}'}(h), \left\|2^{(t+r-m)d}\right\|_{2^{(m-t)d},2^{(m-r)d}}\right) = \varphi_m(t,r)$$

hold for all $t, r \in \{1,\ldots,m\}$, $1 \le i_1 < \cdots < i_t \le m$, $1 \le j_1 < \cdots < j_r \le m$. □

Corollary 5 gives a new characterization of MDS linear mappings. Moreover MDS linear mappings which are used in the connection with resistance to linear and differential methods can be derived from permutation group point of view. This fact shows that permutation group strategy is a general approach for design block cipher primitives.

For any linear mapping $h \in GL_n(2)$ suppose

$$B_h = \left\{ (t,r) \;\middle|\; \begin{array}{l} t + r \le m, \exists\ 1 \le i_1 < \cdots < i_t \le m, \quad 1 \le j_1 < \cdots < j_r \le m, \\ \rho\left(\mathbf{c}_{\mathbf{V}(i_1,\ldots,i_t),\mathbf{V}(j_1,\ldots,j_r)}(h), \left\|2^{(t+r-m)d}\right\|_{2^{(m-t)d},2^{(m-r)d}}\right) > \varphi_m(t,r) \end{array} \right\}.$$

In the following proposition the differential branch number of a linear mapping h is derived using the set B_h.

Proposition 6. *For any $h \in GL_n(2)$ we have*

$$\mathrm{bn_d}(h) = \begin{cases} \min\{t+r | (t,r) \in B_h\}, \ if \ B_h \neq \emptyset, \\ m+1, \ if \ A_h = \emptyset. \end{cases}$$

Proof. Let $\mathrm{bn_d}(h) = b$ and $b < m+1$. Then there exist $t, r \in \mathbb{N}$, $\alpha \in V_n$ such that $t + r = b$, $\mathrm{wt}(\alpha) = t$, $\mathrm{wt}\left(\alpha^h\right) = r$. Therefore, there exist $1 \leq i_1 < \cdots < i_t \leq m$, $\left|V(i_1,\ldots,i_t)^h \cap V(j_1,\ldots,j_r)\right| > 1$ and $\rho\left(\mathbf{c}_{\mathbf{V}(i_1,\ldots,i_t),\mathbf{V}(j_1,\ldots,j_r)}(h), \left\|2^{(t+r-m)d}\right\|_{2^{(m-t)d},2^{(m-r)d}}\right) > \varphi_m(t,r)$. Hence, we have that $(t,r) \in B_h$ and $\mathrm{bn_d}(h) \geq \min\{t+r | (t,r) \in B_h\}$. On the other hand, if $(t_0,r_0) \in B_h$, then there exists $\alpha \in V_n$ such that $\mathrm{wt}(\alpha) = t_0$, $\mathrm{wt}\left(\alpha^h\right) = r_0$. Hence, $\mathrm{bn_d}(h) \geq t_0 + r_0 \geq \min\{t+r | (t,r) \in B_h\}$. □

Note that the linear branch number can be expressed similarly to the differential branch number using matrix ${}^t h$.

5 The distances from the linear mappings to the groups $S(V(i_1,\ldots,i_t)) \wr S_{2^{n-dt}}$ and $S_{2^{\frac{n}{u}}} \uparrow S_u$

Since G_{XS} is a subgroup of the wreath product $S(V(i_1,\ldots,i_t)) \wr S_{2^{n-dt}}$ and $S_{2^{\frac{n}{u}}} \uparrow S_u$ it is importantly to study the Hamming distances from h to the groups $S(V(i_1,\ldots,i_t)) \wr S_{2^{n-dt}}$ and $S_{2^{\frac{n}{u}}} \uparrow S_u$. If the Hamming distance is little, then it may be used for construction a distinguishing of cipher function.

Suppose

$$\chi_{\mathbf{W}}(h) = \min\{\chi(h,g) | g \in S(W) \wr S_{2^{n-dt}}\},$$

where $\chi(h,g)$ is the Hamming distance between g and h, i.e. $\chi(g,h) = 2^n - \sum_{\alpha \in V_n} \mathrm{Ind}\left\{\alpha^g = \alpha^h\right\}$.

Suppose $\mathbf{c}_{\mathbf{W}}(h) = c_{\mathbf{W},\mathbf{W}}(h)$, $\mathbf{W} = \left\{W_i | i = 1,\ldots,\ 2^{n-dt}\right\}$ is a partition of the vector space V_n and $|W_i| = 2^{dt}$, $i = 1,\ldots,2^{n-dt}$. It is known [12] that for all substitution $h \in S(V_n)$ we have

$$\chi_{\mathbf{W}}(h) = 2^n - \max\left\{ \sum_{i=1}^{2^{n-dt}} c_{i,i^f}(h) \,|\, f \in S_{2^{dt}} \right\},$$

$$\chi_{\mathbf{W}}(h) \leq 2^n - 2^{n-dt} \left\lceil 2^{2dt-n} \right\rceil.$$

In the following proposition we obtain $\chi_{\mathbf{W}}(h)$ for linear mapping $h \in GL_n$ using the rang of its submatrix.

Proposition 7. *Let $h \in GL_n(2)$, $t \in \{1, \ldots, m-1\}$, $1 \leq i_1 < \cdots < i_t \leq m$, $\mathbf{W} = \mathbf{V}(i_1 \ldots, i_t)$, $n = md$. Then we have*

$$\chi_{\mathbf{W}}(h) = 2^n - 2^{n-\text{rang } h\begin{pmatrix} i_1, \ldots, i_t \\ i'_1, \ldots, i'_{m-t} \end{pmatrix}},$$

where $\{i'_1, \ldots, i'_{m-t}\} = \{1, \ldots, m\} \setminus \{i_1, \ldots, i_t\}$, $1 \leq i'_1 < \cdots < i'_{m-t} \leq m$.

Proof. Using proposition 4, we get $c_{i,j}(h) \in \left\{ 0, 2^{td-\text{rang } h\begin{pmatrix} i_1, \ldots, i_t \\ i'_1, \ldots, i'_{m-t} \end{pmatrix}} \right\}$, $i, j = 1, \ldots, 2^{n-dt}$. It is easily to show that the matrix

$$2^{\text{rang } h\begin{pmatrix} i_1, \ldots, i_t \\ i'_1, \ldots, i'_{m-t} \end{pmatrix} - td} \mathbf{c}_{\mathbf{W}}(h)$$

is a doubly stochastic matrix. Therefore, there exists $g \in S_{2^{dt}}$ such that

$$c_{i,i^g}(h) = 2^{td-\text{rang } h\begin{pmatrix} i_1, \ldots, i_t \\ i'_1, \ldots, i'_{m-t} \end{pmatrix}}, \; i = 1, \ldots, 2^{dt}.$$

Hence, we have

$$\chi_{\mathbf{W}}(h) = 2^n - \max\left\{ \textstyle\sum_{i=1}^{2^{n-dt}} c_{i,i^f}(h) \,|\, f \in S_{2^{dt}} \right\} =$$

$$= 2^n - \textstyle\sum_{i=1}^{2^{n-dt}} c_{i,i^g}(h) = 2^n - 2^{n-\text{rang } h\begin{pmatrix} i_1, \ldots, i_t \\ i'_1, \ldots, i'_{m-t} \end{pmatrix}}.$$

□

Since rang $h\begin{pmatrix} i_1,\ldots,i_t \\ i'_1,\ldots,i'_{m-t} \end{pmatrix} = \min\{td,(m-t)\,d\}$, we have the following corollary.

Corollary 8. *Let $h \in GL_n(2)$ be an MDS linear mapping. Then for all $t \in \{1,\ldots,m-1\}$, $1 \le i_1 < \cdots < i_t \le m$, we have that $\chi_{\mathbf{W}}(h)$ achieves the maximum possible value, i.e.*

$$\chi_{\mathbf{W}}(h) = 2^n - 2^{n-dt}\left\lceil 2^{2dt-n}\right\rceil.$$

In the following proposition we obtain lower bounds of a distances from a MDS linear mappings to the groups $G_{\mathbf{A}}$.

Proposition 9. *Let $h \in GL_{md}$ be a MDS linear mapping, $\mathbf{A} = \{A_1,\ldots,A_u\}$ be a partition of the set $\{1,\ldots,m\}$, $|A_i| = \frac{n}{u}$. Then we have*

$$\chi(h, G_{\mathbf{A}}) \ge 2^n - 2^{n-\frac{n}{u}}.$$

Proof. Suppose $\chi(h, G_{\mathbf{A}})$ is equal to r, $g \in G_{\mathbf{A}}$ and $\chi(h,g) = r$. Taking into account proposition 3, we obtain that there exists permutation $p \in S_m$ such that $g\hat{p} \in S\left(V\left(i_1,\ldots,i_{\frac{n}{u}}\right)\right) \wr S_{2^{n-\frac{n}{u}}}$. Notice that $h\hat{p}$ is MDS linear mapping. Using corollary 8, we get

$$\chi(h,g) = \chi(h\hat{p}, g\hat{p}) \ge \chi_{\mathbf{W}}(h\hat{p}) = 2^n - 2^{\frac{n}{u}}.$$

□

From proposition 9 it follows that there are no good approximations for MDS linear mapping in the group $G_{\mathbf{A}}$.

References

[1] Blaum M., Roth R.M. On the lowest density MDS codes // IEEE transactions on information theory. – 1999. – V. 45 – P. 46-59.

[2] Burov D.A., Pogorelov B.A. An attack on 6 rounds of Khazad // 4rd Workshop on Current Trends in Cryptology (CTCrypt 2015). – 2015. – P. 318-329.

[3] Burov D.A., Pogorelov B.A. The influence of linear mapping reducibility on choice of round constants // 5rd Workshop on Current Trends in Cryptology (CTCrypt 2016). – 2016. – P. 7-22.

[4] Daemen J. Cipher and hash function design strategies based on linear and differential cryptanalysis // PhD thesis. – K.U. Leuven. – 1995.

[5] Daemen J., Rijmen V. The design of Rijndael: AES – The Advanced Encryption Standard // Springer. – 2002.

[6] Dixon J.D., Mortimer B. Permutation groups // Springer, 1996. – 346 p.

[7] Guo J., Jean J., Nicolic I., Qiao K., Sasaki Y., Meng Sim S. Invariant subspace attack against full Midory64 [Electronic resource] // Cryptology ePrint Archive. Report 2015/1189. Mode of access: http://eprint.iacr.org/2015/1189.

[8] Kaluznin L.A., Lkin M.H., Sushchanskii V.I. The operation of exponentiation of permutation groups // I. Izvesiya VUZov. Seriya Matematicheskaya. – 1979. – V. 8. – 26-33. (in Russian).

[9] Leander G., Abdelraheem M., Alkhzaimi H., Zenner E. A Cryptanalysis of PRINT cipher: The Invariant Subspace Attack. Advances in Cryptology // CRYPTO'11. LNCS. — 2011. — V. 6841. — P. 206-221.

[10] Leander G., Minaud B., Sonjom S. A generic approach to invariant subspace attacks: cryptanalysis of Robin, iSCREAM and Zorro // Eurocrypt'15. LNCS. — 2015. — V. 9056. — P. 254-283.

[11] Pogorelov B.A. Permutation groups. Part 1, general questions // Moscow, 1986. — 316 p. (in Russian)

[12] Pogorelov B.A., Pudovkina M.A. On the distance from permutations to imprimitive groups for a fixed system of imprimitivity // Discrete Mathematics and Applications. – 2014. – V. 24, issue 2. – P. 95-108. (in Russian).

[13] Pogorelov B.A., Pudovkina M.A. Combinatorical characterization of XL-layers // Mathematical Aspects of Cryptography. – 2013. – V. 4, no. 3. – P. 99-129 (in Russian).

The branch numbers of linear transformations in encryption algorithms

Andrey Erokhin, Fedor Malyshev, Andrey Trishin

Abstract

We study the properties of linear medium of cipher related to resistance against multidimensional linear cryptanalysis. We define the branch number of linear medium of cipher transformation and the branch number of nonsingular matrix.

Keywords: symmetric encryption, branch number, multidimensional linear cryptanalysis

Introduction

C. Shannon [1] stated essential requirements of confusion and diffusion for encryption algorithms. These requirements were necessary to ensure a practical security of a secret communication. By the 60th years of the twentieth century his recommendations were implemented in emerging of SP-networks ([2], [3]). The SP-network contains interleaving transformations with "good" local confusion and "weak" diffusion properties (S-boxes) and transformations with "good" diffusion and "weak" confusion properties (overall permutations of bits).

The linear cryptanalysis and its dual differential cryptanalysis (see [4]) led to emergence of new block ciphers including XSL-ciphers [5]. The linear and differential cryptanalysis techniques allowed deeper understanding of an essence of diffusion property. A new diffusion criteria was found. The diffusion property began to be measured by a numerical characteristic called *the linear medium's coefficient of diffusion* (LMCD) of cipher transformation [6]. Actually the linear medium of a cipher has two coefficients of diffusion associated with the linear and differential cryptanalysis. The related terms are

the linear medium's linear coefficient of diffusion (LMLCD) and *the linear medium's differential coefficient of diffusion* (LMDCD) of cipher transformation. In this paper we will consider only the LMLCD (in short, LMCD). Its clear definition will be given later (in section 2.1).

Usually the LMLCD and the LMDCD may be estimated relatively easily. So these coefficients are a good supplement to parameters such as the length of encryption key and the number of rounds in a cipher [6], [5]. This parameters characterize security of a cipher. Linear transformations (and their matrices) in ciphers are characterized by the branch numbers called the linear and differential branch numbers. The linear and differential branch numbers are induced by the LMLCD and the LMDCD respectively. The MDS matrices [7] have a maximum value of the branch numbers among all nonsingular matrices of the same size.

Unfortunately, the linear and differential branch numbers do not distinguish substitution matrices used in SP-networks. All permutations have a minimal value of the branch number equal to 2, but different permutations have different diffusion properties (according to Shannon) as is demonstrated by the avalanche effect. This paper aims to eliminate this disadvantage using the multidimensional linear cryptanalysis [8].

This paper is organized as follows: Section 1 contains a description of the s-dimensional linear cryptanalysis, $s \geqslant 1$, considered in [4], [6] for arbitrary functional schemes defining mappings of binary vector spaces; in Section 2 we define (in connection with the multidimensional linear cryptanalysis) the LMCD of cipher transformation. Also in Section 2 for nonsingular matrices we define *the matrix's linear characteristic of diffusion* (MLCD). The linear branch number is a special case of MLCD.

1 Multidimensional linear cryptanalysis

The main aim of multidimensional linear cryptanalysis is to construct the s-dimensional linear relation linking bits of plaintext and ciphertext. Cipher transformations will be given further by the functional schemes. We give below necessary notations related to the functional schemes.

1.1. **The functional scheme** defining the cipher transformation

$$F : V_N \times V_K \to V_M,\ F(a, z) = b,\ (a, z) \in V_N \times V_K, \tag{1}$$

may be represented by the command sequence of its program implementation. Here $V_K = GF(2)^K$ is a set of encryption keys, V_N is a set of plaintexts, and V_M is a set of ciphertexts.

In the case of block ciphers $M = N$. Then $a \in V_N$ is a block of plaintext, $b \in V_M$ is a block of ciphertext, $z \in V_K$ is a secret key. Then vectors $(a, z) \in V_N \times V_K$ and $b \in V_M$ are the input and output of the functional scheme, respectively.

Let

$$f_i : V_{n_i} \to V_{m_i},\ i = 1, \ldots, k, \tag{2}$$

be nonlinear functional elements of the functional scheme defining the cipher transformation (1). The argument (input) of the map f_i is denoted by $x_i \in V_{n_i}$, the image (output) of the map f_i is denoted by $y_i \in V_{m_i},\ y_i = f_i(x_i),\ i = 1, \ldots, k$. All other operations of the functional scheme are linear. These linear operations form the linear medium of cipher transformation.

We may assume that the mappings (2) are arranged in such way that outputs of f_i may be inputs of f_j (perhaps indirectly, after linear operations) only if $i < j$. As a result,

$$x_j = c_j(z, a, y_1, \ldots, y_{j-1}) = zc_{*j} + ac_{0j} + y_1c_{1j} + \ldots + y_{j-1}c_{j-1,j}, \tag{3}$$

$$b = c_{k+1}(z, a, y_1, \ldots, y_k) = zc_{*,k+1} + ac_{0,k+1} + y_1c_{1,k+1} + \ldots + y_kc_{k,k+1}, \tag{4}$$

$$\text{where } c_j : V_{K+N+\sum_{i=1}^{j-1} m_i} \to V_{n_j},\ j = 1, \ldots, k, k+1, \tag{5}$$

are linear (over $GF(2)$) mappings. Here $c_{ij},\ i = 0, 1, \ldots, j-1,\ j = 1, \ldots, k, k+1$, are $m_i \times n_j$ matrices, $m_0 = N,\ n_{k+1} = M$. Further, $c_{*j}, j = 1, \ldots, k, k+1$, are $K \times n_j$ matrices. Linear mappings and their matrices are denoted by the same symbols.

Suppose round keys are added to intermediate blocks of text by means of bitwise XOR operation. The linear mappings (5) are combined into united linear mapping

$$C : V_{K+N+\sum_{i=1}^{k} m_i} \to V_{\sum_{i=1}^{k} n_i + M}$$

and C is a $(K+N+\sum_{i=1}^{k} m_i) \times (\sum_{i=1}^{k} n_i + M)$ matrix. The uppermost "row" of matrix C is the identity matrix $C_0 = (c_{*1}, \dots, c_{*k}, c_{*,k+1})$. Denote submatrix of matrix C consisting of "rows" $(c_{i1}, \dots, c_{ik}, c_{i,k+1})$, $i = 0, 1, \dots, k$, by $\widetilde{C}$. (Suppose $c_{ij} = 0$ for $i \geqslant j$.) The equations (3) and (4) may be written in the form $(z, a, y_1, \dots, y_k)\, C = (x_1, \dots, x_k, b)$, or

$$(a, y)\widetilde{C} + zC_0 = (z, a, y)\, C = (x, b),$$

where $(y_1, \dots, y_k) = y$, $(x_1, \dots, x_k) = x$.

1.2. Additive method for constructing multidimensional linear relations. Suppose $s \geqslant 1$. As in the articles [4], [6] a multidimensional linear relation of cipher transformation (1) is given by linear mappings $L' : V_N \to V_s$, $L : V_K \to V_s$, $L'' : V_M \to V_s$ and is represented as

$$bL'' = aL' + zL + \eta, \tag{6}$$

where $b = F(a, z)$, $L'' \neq 0$, $\eta = \eta(a, z) \in V_s$ is a random vector of "discrepancy", $a \in V_N$ is a uniformly distributed random vector, $z \in V_K$ is a fixed key.

An efficiency of the relation (6) is characterized by the probability distribution of the vector η on the set V_s. The closer this distribution is to a degenerate distribution, the more relation (6) is effective. Consider an entropy $H(\eta)$ [1] of probability distribution of the vector η on the set V_s as a measure of uncertainty of the vector η.

The relation (6) is obtained by summing the local s-dimensional probability linear relations of mappings (2) for all $i = 1, \dots, k$, namely

$$y_i l_i'' = x_i l_i' + \eta_i, \tag{7}$$

where $y_i = f_i(x_i)$, $\eta_i = \eta_i(x_i) \in V_s$ is a random vector of "discrepancy".

The entire set of relations (7) is given by the set $\mathfrak{L} = ((l_i', l_i''), i = 1, \dots, k)$ which consists of binary $n_i \times s$ and $m_i \times s$ matrices defining linear mappings $l_i' : V_{n_i} \to V_s$, $l_i'' : V_{m_i} \to V_s$, $i = 1, \dots, k$, such that $V_s = \sum_{i=1}^{k} \mathrm{Im} l_i''$. We will call this set $\mathfrak{L}$ as the system of the local s-dimensional probability linear relations of cipher transformation (1).

The mappings l_i', l_i'', $i = 1, ..., k$, must satisfy two requirements. The first requirement is to move the distribution of vectors $\eta_i = x_i l_i' + y_i l_i''$ near to

degenerate distribution, that is to make these vectors more specific, therefore, in particular $\mathrm{Im}l'_i \subseteq \mathrm{Im}l''_i$. The entropy $H(\eta_i)$ of the probability distribution of the vector η_i on the set V_s is a measure of uncertainty of each random vector $\eta_i \in V_s$, $i = 1, \ldots, k$. A probability distribution of η_i is calculated under the assumption that $x_i \in V_{n_i}$ are uniformly distributed.

The second requirement is the conformity of the system $\mathfrak{L} = ((l'_i, l''_i), i = 1, \ldots, k)$ that is using (3), (4) (but without using equations $y_i = f_i(x_i)$, $i = 1, \ldots, k$). We can reduce the sum

$$\eta_{\mathfrak{L}} = \sum_{i=1}^{k} \eta_i = \sum_{i=1}^{k} (x_i l'_i + y_i l''_i)$$

to the form

$$\eta = aL' + zL + bL'',$$

where $L' : V_N \to V_s$, $L : V_K \to V_s$, $L'' : V_M \to V_s$ are some linear mappings. This is equivalent to the solvability of equation

$$\widetilde{C} \begin{pmatrix} l' \\ L'' \end{pmatrix} = \begin{pmatrix} L' \\ l'' \end{pmatrix} \tag{8}$$

with respect to $N \times s$ and $M \times s$ matrices L', L''. In equation (8) the matrices l', l'' consist of stacked matrices l'_i, l''_i, $i = 1, \ldots, k$, respectively.

If the system $\mathfrak{L}$ is conformal then we suppose $L = C_0 \begin{pmatrix} l' \\ L'' \end{pmatrix}$. Using (8) and the first requirement for mappings l'_i, l''_i, $i = 1, \ldots, k$, we get $\mathrm{Im}L' \subseteq \mathrm{Im}L'' = V_s$. The set $\mathfrak{W}_s$ of all conformal systems $\mathfrak{L} = (l', l'')$ is a vector space over the field $GF(2)$. If the system $\mathfrak{L} = (l', l'')$ is conformal then

$$\eta_{\mathfrak{L}} = xl' + yl'' = \sum_{i=1}^{k} (x_i l'_i + y_i l''_i) = \sum_{i=1}^{k} \eta_i = aL' + zL + bL'' = \eta. \tag{9}$$

1.3. About key recovery. If there exist the conformal system $\mathfrak{L} = (l', l'')$ and a set of plaintext and ciphertext pairs $\left(a^{(j)}, b^{(j)}\right) \in V_N \times V_M$, $j = 1, \ldots, T$ (zL is fixed) then using (9) we get T realizations of a random vector η in the form

$$\sum_{i=1}^{k} \eta_i^{(j)} = \sum_{i=1}^{k} \left(x_i^{(j)} l'_i + y_i^{(j)} l''_i\right) = a^{(j)} L' + b^{(j)} L'' + zL,\ j = 1, \ldots, T. \tag{10}$$

Here $b^{(j)} = F(a^{(j)}, z)$; $x_i^{(j)}$ and $y_i^{(j)}$ are the input and the output of mapping f_i, $i = 1, \ldots, k$, if $a^{(j)}$ is the input of functional scheme; $\eta_i^{(j)} = x_i^{(j)} l_i' + y_i^{(j)} l_i''$. If zL is correct (i.e. $zL = z_0 L$, where z_0 is a true key) then the set of vectors (10) must correspond to the probability distribution of random vector η owing to the fact that cryptanalyst can get some key information. Usually several first and/or last operations of entire cipher are not included in the cipher transformation (1). In this case vectors $a^{(j)}, b^{(j)}$, $j = 1, \ldots, T$, are expressed through plaintext, ciphertext and certain subkeys. These subkeys can also be determined.

The efficiency of a key recovery depends on a value $\sigma = \sum_{v \in V_s} \varepsilon_v^2 \approx \ln 2 \cdot (s - H(\eta))/2^{s-1}$, where $\left\{p_v = \frac{1}{2^s} + \varepsilon_v,\ v \in V_s\right\}$ is a probability distribution of vector η taking values in V_s. This probability distribution is estimated under the assumption that random summands η_i, $i = 1, \ldots, k$, are statistically independent and $x_i \in V_{n_i}$, $i = 1, \ldots, k$, are uniformly distributed. Each of these assumptions may be wrong. We do not have any theoretical results confirming the closeness of computed (under our assumptions) distributions $\left\{\frac{1}{2^s} + \widetilde{\varepsilon}_v,\ v \in V_s\right\}$ to true distributions. Therefore the closeness of $\widetilde{\varepsilon}_v$ to ε_v, $v \in V_s$, requires an experimental verification.

The smaller is the uncertainty of $H(\eta)$ (or the greater σ) the less amount of data is needed for a key recovery attack. Therefore we are interested in numbers $i \in \{1, \ldots, k\}$ such that $x_i l_i' = f_i(x_i) l_i''$ for all $x_i \in V_{n_i}$, particularly $l_i' = 0, l_i'' = 0$. For such numbers a random vector η_i does not introduce an uncertainty into η. Thus conformal systems $\mathfrak{L}$ having a minimal value of $\theta_{\mathfrak{L}} = |\{i \in \{1, \ldots, k\} \,|\, l_i'' \neq 0\}|$ are preferred.

2 The linear medium's coefficient of diffusion of cipher transformations and the branch numbers of nonsingular matrices

2.1. In the case $s = 1$ the LMCD of cipher transformation (1) with linear medium C is determined by the formula [6]

$$\theta_1(C) = \min_{\mathfrak{L} \in \mathfrak{W}_1 \setminus \{0\}} \theta_{\mathfrak{L}}. \tag{11}$$

The coefficient $\theta_1(C)$ allows to compare different nonsingular linear transformations of the space $V_n = GF(2)^n$ by their cryptographic features.

Let the matrix $\Lambda \in GL(n,2)$ be used in so-called canonical XSL-cipher action on V_n, $n = m \cdot \kappa$. One round of this cipher consists of two transformations. The first transformation S is nonlinear, $S = (\underbrace{\pi, \ldots, \pi}_{\kappa})$, $\pi \in S_{V_m}$. The second transformation is a multiplication of vectors from V_n by the matrix Λ from the right. Round keys of the canonical XSL-cipher are equal to 0. In the case of the canonical XSL-cipher a set $\mathfrak{W}_1$ in (11) is replaced by

$$\mathfrak{W}_1^{(0)} = \{\mathfrak{L} = ((l'_i, l''_i),\, i = 1, \ldots, k) \,|\, \forall i \in \{1, \ldots, k\} : (l'_i = 0 \Leftrightarrow l''_i = 0)\}.$$

By $C_t(\Lambda)$ we denote the linear medium of the canonical XSL-cipher with t rounds. The branch number of matrix Λ is defined as $\rho_{1,2}(\Lambda) = \theta_1(C_2(\Lambda))$. If the column $l \in V_n^*$ is composed of columns $l_1, \ldots, l_\kappa \in V_m^*$ and $w(l) = \big|\{j \in \{1, \ldots, \kappa\} \big| \, l_j \neq 0\}\big|$ then we can see that

$$\rho_{1,2}(\Lambda) = \min_{l \in V_n^* \setminus \{0\}} \left(w(l) + w(\Lambda l)\right). \tag{12}$$

Indices 1 and 2 in the notation $\rho_{1,2}(\Lambda)$ correspond to $s = 1$ и $t = 2$. The branch number $\rho_{1,2}(\Lambda)$ refers to the matrix Λ divided into κ strips of m rows in each strip and into κ groups of m columns in each group. So the linear branch number of nonsingular matrix is equal to LMCD of related XSL-cipher with 2 rounds. Note that the equation (12) is contained in [5] as a definition of the linear branch number of linear mappings.

Matrices $\Lambda \in GL(n,2)$ having large branch number $\rho_{1,2}(\Lambda)$ are preferable from the viewpoint of a cryptographic design. Using definition (12) we get

$$2 \leqslant \rho_{1,2}(\Lambda) \leqslant \kappa + 1. \tag{13}$$

2.2. Let $P \in GL(n,2)$ be a substitution matrix, then according to (12) $\rho_{1,2}(P) = 2$. Using (13) we can see that for nonsingular matrices 2 is the smallest value of this characteristic. Thus the branch number $\rho_{1,2}(P)$ does not distinguish substitution matrices. This fact is a main disadvantage of $\rho_{1,2}(P)$.

Also this disadvantage takes place for sparse matrices, particularly for matrices Λ such that Λ and Λ^{-1} have the same small number of 1 in each

row and in each column [9]. These matrices are used in different block ciphers [10].

A set of characteristics $\rho_{1,\tau}(P) = \theta_1(C_\tau(P)) = \tau$, $\tau \geqslant 2$, does not change the situation. But distinguishing of different permutations $P \in S_n$ according to the degree of diffusion (in the sense of Shannon) took place in a cryptographic practice. Originally the diffusion properties of permutations P are defined by the avalanche effect. Theorem 1 below describes optimal in this sense permutations P. For these permutations pairs $(i, P(i))$, $i = 1, \ldots, n$, are edges of the generalized de Bruijn graphs [11].

To formulate the Theorem 1 we must introduce some notations. The substitution $\pi : V_n \to V_n$, $x = (x_1, \ldots, x_n) \mapsto (y_1, \ldots, y_n) = y$, is called *significant* if the following conditions hold: (i) y_j depends significantly on x_i for all $i, j \in \{1, \ldots, n\}$; (ii) x_i depends significantly on y_j for all i, j. The SP-networks comprising significant substitutions are called *canonical*.

Other concept refers to directed graphs [11]. If there exists only one directed path from i to j for any vertices $i, j \in \{1, \ldots, n\}$ and this path contains r edges then directed graph Γ on $n > 1$ vertices is called *∂-graph of order $r \geqslant 1$*. The graph Γ^+ dual to Γ is the ∂-graph of order $r + 1$. Recall [12] that vertices of Γ^+ are edges of Γ; graph Γ^+ contains an edge (α, β) if in graph Γ the end of edge α coincides with the beginning of edge β. If we change the direction of all edges of ∂-graph Γ then we get ∂-graph $\overline{\Gamma}$ of the same order. For example, the de Bruijn graph on $n = m^r$ vertices is ∂-graph. The de Bruijn graph on $n = m^r$ $(r > 1)$ vertices is dual to the de Bruijn graph on $n = m^{r-1}$ vertices.

The theorem stated below refers to canonical SP-network on a set V_n, $n = m\kappa$. Further set $\{1, \ldots, n\}$ is divided into m-subsets

$$N(j) = \{(j-1)m + 1, (j-1)m + 2, \ldots, jm\}, j = 1, \ldots, \kappa.$$

The substitutions π of transformation S act on vectors from V_m; the components of these vectors have numbers $N(j), j = 1, \ldots, \kappa$. Further the permutation $P : V_n \to V_n$ is associated with directed graph $\Gamma(P)$ on a set of vertices $\{1, \ldots, n\}$; each number $i \in \{1, \ldots, n\}$ is a beginning of m edges. Ends of these edges form a set $N\left(\left\lceil \frac{P(i)}{m} \right\rceil\right)$ containing $P(i)$.

Theorem 1. *If $n = m^r$ in the canonical SP-network and the graph $\Gamma(P)$ is ∂-graph of order r, then the substitution $(SP)^r : V_n \to V_n$ is significant.*

2.3. The results of previous sections mean that if we want to measure diffusion properties of matrices $\Lambda \in GL(m\kappa, 2)$ by the LMCD of canonical XSL-ciphers with τ rounds and linear medium C_τ then similarly to (11) we must consider the LMCD $\theta_s(C_\tau)$ corresponding to multidimensional (s-dimensional) linear cryptanalysis. Definition of $\theta_{\mathfrak{L}}$ from section 1.3 must be modified. The result is a 2-parameter set of branch numbers $\rho_{s,\tau}(\Lambda) = \theta_s(C_\tau)$, $s \geqslant 1, \tau \geqslant 2$.

The substitution matrices are poorly differ with branch numbers $\rho_{1,\tau}, \tau \geqslant 2$. In the same time according to Theorem 1, permutations $P \in S_n$, $n = m^r$, whose graph $\Gamma(P)$ is a ∂-graph of order r are better with respect to qualitative idea of diffusion.

In this section numbers $i \in \{0, 1, \ldots, n-1\}, n = m^r$, represented as

$$i = (i_0, i_1, \ldots, i_{r-1}) = i_0 + i_1 m + \ldots + i_{r-1} m^{r-1},$$

$i_0, i_1, \ldots, i_{r-1} \in \{0, 1, \ldots, m-1\}$, will be numbers of components of vectors from V_n. Suppose edges of the de Bruijn graph Γ_0 on a set $\{0, 1, \ldots, m^r - 1\}$ are $(i_0, i_1, \ldots, i_{r-1}) \to (i_1, \ldots, i_{r-1}, j)$, $i_0, i_1, \ldots, i_{r-1}, j \in \{0, 1, \ldots, m-1\}$. Then we get $\{0, 1, \ldots, m^r - 1\} = \coprod\limits_{j \in \{0,1,\ldots,m^{r-1}-1\}} N(j)$, where $j = (j_0, j_1, \ldots, j_{r-2})$ and $N(j) = \{jm, jm+1, \ldots, jm+m-1\} = \{(i_0, j_0, j_1, \ldots, j_{r-2}) \,\big|\, i_0 = 0, 1, \ldots, m-1\}$. Thus $i \in N\big(\big[\frac{i}{m}\big]\big)$.

Further, in the canonical SP-network we use the permutation $P_0 \in S_{m^r}$ defined by the equation $P_0(i) = P_0(i_0, i_1, \ldots, i_{r-1}) = (i_1, \ldots, i_{r-1}, i_0)$. Notice that $\Gamma(P_0) = \Gamma_0$.

In this paper we propose a 2-parameter set of branch numbers $\rho_{s,\tau}(\Lambda)$, $s = 1, \ldots, m$, $\tau = 2, \ldots, \lceil \log_m n \rceil$, as a diffusion characteristic of matrix $\Lambda \in GL(m\kappa, 2)$ divided into κ strips of m rows in each strip and into κ groups of m columns in each group. We call this characteristic a *the matrix's linear characteristic of diffusion* (MLCD). Branch numbers corresponding to small values of s, τ are more important. That is we use branch numbers $\rho_{s,\tau}(\Lambda_1)$ and $\rho_{s,\tau}(\Lambda_2)$ to compare diffusion properties of matrices Λ_1, $\Lambda_2 \in GL(n, 2)$ only if we can not do it using $\rho_{s',\tau'}(\Lambda_1)$ and $\rho_{s',\tau'}(\Lambda_2)$, $s' \leqslant s$, $\tau' \leqslant \tau$, $(s', \tau') \neq$

(s, τ), for example, if $\rho_{s',\tau'}(\Lambda_1) = \rho_{s',\tau'}(\Lambda_2)$ for all these s', τ'. Authors do not know which of the two numbers $\rho_{s_1,\tau_1}(\Lambda)$ and $\rho_{s_2,\tau_2}(\Lambda)$, $s_1 > s_2$, $\tau_1 < \tau_2$, is preferable.

A value $\lceil \log_m n \rceil$ is the upper bound for τ. This bound is realized for the special examples of substitutions $P \in S_{m^r}$, $n = m^r$, such that $(SP)^r$ (unlike $(SP_0)^r$) is not significant, but $\rho_{m,\tau}(P) = \rho_{m,\tau}(P_0)$ for $\tau < r$ and $\rho_{m,r}(P) < \rho_{m,r}(P_0)$ (see Theorem 2). Authors don't know such examples for an upper bound for s. Maybe inequality $\rho_{m',r}(P) < \rho_{m',r}(P_0)$ holds if $m' < m$.

If the set $\{(s, \tau) | \, 1 \leqslant s \leqslant m, \, 2 \leqslant \tau \leqslant \lceil \log_m n \rceil\}$ is linearly ordered ((1, 2) is the first element, $(m, \lceil \log_m n \rceil)$ is the last element) with respect to decreasing of priority $\rho_{s,\tau}(\Lambda)$ then values $\rho_{s,\tau}(\Lambda)$, $1 \leqslant s \leqslant m$, $2 \leqslant \tau \leqslant \lceil \log_m n \rceil$, may be interpreted as numbers after comma in decimal representation of real numbers. Maybe we can use only $\rho_{1,2}(\Lambda)$ for matrices Λ which is not sparse. There is need to use other branch numbers for sparse matrices. According to the Theorem 2 below we must use the "last" branch number $\rho_{m,\lceil \log_m n \rceil}(\Lambda)$ for substitution matrices.

When we determined the MLCD we were limited to consideration of linear medium C_τ of canonical XSL-ciphers with $\tau \leqslant \lceil \log_m n \rceil$ rounds and values $s \leqslant m$. Further, suppose nonzero $m \times s$ matrices l'_{ij}, l''_{ij}, $i = 1, \ldots, \tau$, $j = 1, \ldots, \kappa$ from conformal system $\mathfrak{L} \in \mathfrak{W}_s$ have a maximal rank equal to s. Otherwise (according to definitions below) $\theta_s(C_\tau) = \theta_1(C_\tau)$, and we do not get additional opportunities for distinguishing of substitutions and distinguishing of sparse matrices $\Lambda \in GL(n, 2)$.

Consider the conformal system

$$\mathfrak{L} = ((l'_i, l''_i), i = 1, \ldots, \tau) = ((l'_{ij}, l''_{ij}), \, i = 1, \ldots, \tau, \, j = 1, \ldots, \kappa) \in \mathfrak{W}_s,$$

where $l''_i = \Lambda l'_{i+1}$, $i = 1, \ldots, \tau - 1$. This system may be defined by the set $\widehat{\mathfrak{L}} = (l'_1, l'_2, \ldots, l'_{\tau-1}, l'_\tau; l''_\tau)$ consisting of $n \times s$ matrices.

Similarly to (11) we put

$$\theta_s(C) = \min_{\mathfrak{L} \in \mathfrak{W}_s^{(0)} \setminus \{0\}} \theta_{\mathfrak{L}},$$

where $\mathfrak{W}_s^{(0)} = \{\mathfrak{L} \in \mathfrak{W}_s \, \big| \, l'_{ij} = 0 \Leftrightarrow (\Lambda l'_{i+1})_j = 0, \, i = 1, \ldots, \tau - 1, \, j = 1, \ldots, \kappa\}$, $\theta_{\mathfrak{L}} = \big|\big\{(i, j) \in \{1, \ldots, \tau\} \times \{1, \ldots, \kappa\} | \, l'_{ij} \neq 0\big\}\big|$.

Theorem 2. *Suppose* $n = m^r$, $P_j \in S_{m^r}$, $j = 0, 1, \ldots, r-2$,

$$P_j(i_0, i_1, \ldots, i_{r-j-1}, i_{r-j}, \ldots, i_{r-1}) = (i_1, \ldots, i_{r-j-1}, i_0, i_{r-j}, \ldots, i_{r-1}),$$

$i_0, i_1, \ldots, i_{r-1} \in \{0, 1, \ldots, m-1\}$. *Then*

$$\rho_{m,\tau}(P_j) = \tau m^{\tau-1} \text{ for } j + \tau \le r \text{ and } \rho_{m,\tau}(P_j) = \tau m^{r-1-j} \text{ for } j + \tau > r.$$

Corresponding to the Theorem 2 we get $\rho_{m,r-1}(P_1) = \rho_{m,r-1}(P_0)$, but $\rho_{m,r}(P_1) < \rho_{m,r}(P_0)$. Notice that substitution $(SP_0)^r$ is significant, but $(SP_1)^r$ is not significant.

The Theorem 2 holds for an identical permutation P_{r-1}, that is $\rho_{m,\tau}(P_{r-1}) = \tau$, $\tau = 2, \ldots, r$.

Substitutions $(SP_j)^{r-j} \in S_{V_{m^r}}$, $j = 0, 1, \ldots, r-1$, are significant for blocks of size m^{r-j}. Substitutions $(SP_j)^\tau$, $\tau \in \mathbb{Z}$, act on these m^j blocks independently and identically. Permutations $P_0, P_1, \ldots, P_{r-1}$ are ranked completely with respect to qualitative idea of diffusion (according to the avalanche effect), and permutations P_j are preferred for small values j. In the same time

$$\rho_{m,\tau}(P_{r-1}) < \rho_{m,\tau}(P_{r-2}) < \ldots < \rho_{m,\tau}(P_{r-\tau+2}) <$$
$$< \rho_{m,\tau}(P_{r-\tau+1}) < \rho_{m,\tau}(P_{r-\tau}) = \rho_{m,\tau}(P_{r-\tau-1}) = \ldots = \rho_{m,\tau}(P_1) = \rho_{m,\tau}(P_0).$$

Characteristic $\rho_{m,\tau}$ does not distinguish permutations $P_0, P_1, \ldots, P_{r-\tau}$ for $\tau < r$ and this fact is a defect of $\rho_{m,\tau}$. But values $\rho_{m,\tau}(P_j) = \rho_{m,r}(P_j)$, $j = 0, 1, \ldots, r-2, r-1$ are completely different for $\tau = r$, and the smaller j the greater the value of this characteristic.

References

[1] Shannon C. Communication Theory of Security Systems. Bell Syst. Techn. J. V. 28. 1948. P. 656–715.

[2] Diffie W., Hellman M. Privacy and Authentication: An Introduction to Cryptography // Proc. IEEE. 1979. V. 67. $\mathcal{N}$ 3. P. 71–109.

[3] Massey J. L. An introduction to contemporary cryptology // Proc. IEEE. 1988. V. 76. $\mathcal{N}$ 5. P. 533–549.

[4] Malyshev F. M. Duality of differential and linear cryptanalysis in cryptography // Mat. Vopr. Kript. 2014. V. 5. $\mathcal{N}$ 3. P. 35–48 (in Russian).

[5] Daemen J., Rijmen V. The Design of Rijndael: AES - The Advanced Encryption Standard. Springer Verlag. 2002.

[6] Malyshev F. M., Trifonov D. I. The diffusion properties of XSLP-ciphers // Mat. Vopr. Kript. 2016. V. 7. $\mathcal{N}$ 3. P. 47–60 (in Russian).

[7] McWilliams F. J., Sloan N. J. A. The theory of error-correcting codes. Amsterdam, New York, Oxford: North-Holland Publishing Company, 1977.

[8] Hermelin M., Cho J. Y., Nyberg K. Multidimentional linear cryptanalysis of reduced round Serpent // Adv. Crypt. – Inf. Security and Privacy / Springer. 2008. LNCS 5107. P. 203–215.

[9] Malyshev F. M., Tarakanov V. E. On (v, k)-configurations // Matem. Sbornik. 2001. V. 192. $\mathcal{N}$ 9. P. 85–108 (in Russian).

[10] Panasenko S. P. The encryption algorithms. Special directory. BHV-Petersburg. 2009 (in Russian).

[11] Malyshev F. M., Tarakanov V. E. Generalized de Bruijn graphs // Mat. Zametki. 1997. V. 62. $\mathcal{N}$ 4. P. 540–548 (in Russian).

[12] Hall M. Combinatorial Theory. Toronto, London: Blaisdell Publishing Company, Waltham (Massachusetts), 1967.

On construction of correlation-immune functions via minimal functions

Evgeny Alekseev, Ekaterina Karelina, Oleg Logachev

Abstract

The use of correlation-immune functions in the structure of a cryptographic primitive can resist some statistical compromising key methods. Designing of the modern cryptographic primitives poses the challenge of constructing correlation-immune functions of a relatively large number of arguments. This paper proposes a method combining the two basic approaches to this problem - iterative and a direct-search method. This method is based on minimal correlation-immune functions, and the functions built with its help have no obvious structural characteristics that would distinguish them from a random function. Its first stage is an easily implemented iteration procedure, which allows to build many special functions that depend on the goal number of variables. The second stage is constructing by means of these set elements of the functions with the given cryptographic properties. The paper presents the description of reduction of the problem of constructing at the second stage of a resilient function with a preassigned order to the problem of solving a system of linear pseudo-Boolean equations. As well as how to apply some modification of the method described in order to improve the cryptographic parameters of the known "good" functions through small changes in their support. Examples of successful applications of the methods described are given.

This work was supported by The Russian Foundation for Basic Research, project 16-01-00470-a.

Keywords: boolean functions, correlation-immune functions

1 Introduction

A cryptographic property of the Boolean function is generally referred to as the one, the possession of which allows it to provide resistance relative to a method of analysis for a cryptographic primitive, designed with the help of this function. Most of these properties have been formulated as a result of the analysis of symmetric cryptographic primitives such as

stream ciphers, block ciphers and hash functions. Examples of the cryptographic properties of Boolean functions are non-linearity (resistance to a linear method [13] analysis), correlation immunity (resistance to various methods of correlation analysis, [10], [11]), nondegeneracy (resistance to a method of analysis based on algebraically degenerate approximations [3]) and algebraic immunity (resistance to algebraic methods of analysis [12]). A deep understanding of the specific properties, along with the relations among them, is especially important by solving problems of synthesis, since it allows you to build schemes optimal in terms of stability and efficiency. One of the most complete and profound descriptions of the status of research in this field is contained in the book [7].

There are a number of approaches to the solution of the problem of constructing a Boolean function with a given set of cryptographic properties. The simplest is the brute-force search method that checks all the functions of a sufficiently large suitable set. The advantage of this method is its simplicity and reliability. However, this method can be used in practice only with small dimensions (for example, the method is applicable to the selection of nodes replacement $V_4 \to V_4$ cipher Magma [8], but is much less effective when choosing $V_8 \to V_8$ cipher Kuznechik [8]). The sets from which the functions are selected are built in such a way that the selected functions a fortiori possessed a certain set of positive properties. The examples are the sets of the Maiorana-McFarland classes and $\mathcal{PS}$ [7]. The ability to exercise effective enumeration in sets of these classes is due to the regularity of their structure — elements of these sets are parameterized by some algebraic structures which in the aggregate can be informally called a "basis". For the class Maiorana-McFarland such a "basis" can thus be considered a direct product of a set of permutations on V_n and sets of Boolean functions of n variables (the cardinality of the set is respectively equal to $(2^n)! \cdot 2^{2^n}$). Regularity of the function structure of such classes can lead to the existence of the unremovable weaknesses. It has been shown in the paper of [6] that the use of the functions of Maiorana-McFarland class as a function of the complexity of the filter generator can lead to a considerable reduction in resistance to the threat of key compromise.

Another method is called iterative, which is about selecting some suit-

able function from a small number of variables (it can be found through a brute-force search method), after which on its basis the functions are built of an increasing number of variables with the help of some procedures. The procedures followed are defined in such a way that the resulting function with their use is guaranteed to have the necessary properties. The scheming should be continued until another constructed function depends on the required number of variables. One of the methods of this type had been proposed in the paper [9]. This method is to a greater extent not intended to build functions to be used in primitives, but to get results on the reachability of some theoretical upper bounds for different parameters. At each algorithm step associated with the increase in the number of variables leads to a certain structuredness into a function. The property which acquired may not always be positive from the perspective of cryptography. For example, thus a function $f^4_{10.2}$ constructed in the paper [9] is algebraically degenerate (that is shown in the paper of the [2]), which can lead to weakness described in the work [3].

In this paper a method for constructing function with the specified cryptographic properties based on a combination of the above approaches. With regard to the enumeration part, the guaranteed property is a correlation immunity of the k-th order, and the set whose elements can be efficiently searched through is a linear space of a special kind. That is to say, in this case, the analogue informal "basis" mentioned above is the basis of the space, consisting of k-minimal correlation-immune functions. A simple iterative method for their construction is proposed in the paper. Meanwhile the method of building up the number of variables the function depends upon suggests a significant randomization when choosing parameters at each step, which leads in the final class to the absence of those structural properties that would have distinguished the constructed functions from the random ones and could have lead to security weaknesses. With respect to construction of functions with given characteristics we describes reduction of the search problem of the resilient (correlation-immune and balanced) function with given order in the built-up space to the problem of solving a system of linear pseudo-Boolean equations.

The proposed approach to the building of functions can be modified

in order to study "neighbourhoods" of the already known functions. The subject of the research may be the presence of functions with the better cryptographic characteristics in the neighborhood under consideration. The value of the given cryptographic function f is slightly changed so that this change would guaranteedly not violate a certain part of its properties. Other characteristics are calculated directly. This change of function f is about replacing the minimal correlation-immune functions, to which it can be decomposed, to the minimal functions of the same weight from the function decomposition $f \oplus 1$.

The examples of successful application of the main method of construction functions with given properties and of the research method of the known functions are given in the paper.

2 Basic concepts and notations

Let $\mathbb{F}_2$ be the finite field of 2 elements. For any $n \in \mathbb{N}$ define $V_n = (\mathbb{F}_2 \times \ldots \times \mathbb{F}_2) = \mathbb{F}_2^n$ — vector space of a set of length n with the components from the field $\mathbb{F}_2$, $V_n^* = V_n \setminus \{0^n\}$, where $0^n = (0, \ldots, 0) \in V_n$. Boolean functions of n variables is a correspondence from V_n into $\mathbb{F}_2$. Constant Boolean functions are denoted as $\mathbf{1}$ and $\mathbf{0}$. The set of all Boolean functions is denoted as $\mathcal{F}_n$. The support $supp(f)$ of a Boolean function $f \in \mathcal{F}_n$ is a set $supp(f) = \{ x \in V_n \mid f(x) = 1 \}$. The weight $\mathrm{wt}(f)$ of a Boolean function $f \in \mathcal{F}_n$ is a cardinality of the support. The distance $\mathrm{dist}(f, g)$ between $f \in \mathcal{F}_n$ and $g \in \mathcal{F}_n$ is value of $\mathrm{wt}(f \oplus g)$. The ordinate vector $f \in \mathcal{F}_n$ is a string $(f(x_{2^n-1}), \ldots, f(x_0)) \in V_{2^n}$ (arguments lexicographically ordered from right to left). Algebraic degree $\deg(f)$ of a Boolean function $f \in \mathcal{F}_n$ of n variables is the number of variables in the longest term ANF (Zhegalkin polynomial). For $u \in V_n$ a Boolean function l_u denotes a linear Boolean function $l_u(x) = \langle u, x \rangle$, where $\langle u, x \rangle = \bigoplus_{i=1}^{n} u_i \cdot x_i$ is a scalar product of vectors u and x. The number of significant variables of function l_u is the weight of vector u. The set $\{l_u(x) \oplus b | u \in V_n, b \in \mathbb{F}_2\}$ of affine Boolean functions of n variables is denoted as $\mathcal{A}_n$. Nonlinearity $\mathrm{nl}(f)$ of a Boolean function $f \in \mathcal{F}_n$ is the Hamming distance to the set

of all affine functions $\mathcal{A}_n$: $\mathrm{nl}(f) = \mathrm{dist}(f, \mathcal{A}_n) = \min\limits_{l \in \mathcal{A}_n} \mathrm{dist}(f, l)$.

A Boolean function $f \in \mathcal{F}_n$ is correlation-immune of order m, $1 \leqslant m \leqslant n$ (further CI-function), if for any vector $u \in V_n$ such that $1 \leqslant \mathrm{wt}(u) \leqslant m$ the equality $\mathrm{wt}\left(f^{'}\right) = \frac{\mathrm{wt}(f)}{2^m}$ performs for any subfunction $f^{'}$ of $n - m$ variables ([9]). Another words a Boolean function f has the maximal distance from the set of affine functions $\mathcal{A}_n$, which are essentially depend on $1, 2, \ldots, m$ variables. Correlation-immune function of order m is the correlation-immune of any lower order, so we introduce the notation

$$cor(f) = \max\{m \in \mathbb{N} \mid f \text{ — correlation immune of order } m\} .$$

Further we use following notations: $\mathrm{CI}(n, k) = \{f \in \mathcal{F}_n | \mathrm{cor}(f) \geqslant k\}$ and $\mathrm{CI}(n) = \mathrm{CI}(n, 1)$. The balanced function $f \in \mathcal{F}_n$ is k-resilient, if $\mathrm{cor}(f) \geqslant k$.

For analysis of cryptographic properties of Boolean functions the Walsh-Hadamard transform is often used. The Walsh-Hadamard transform of a Boolean function $f \in \mathcal{F}_n$ is an integral function $W_f : V_n \to \mathbb{Z}$, which is defined by an equality $W_f(u) = \sum_{x \in V_n} (-1)^{f(x) \oplus \langle u, x \rangle}$. The value $W_f(u)$ is the Walsh-Hadamard coefficients (or Walsh coefficients). For example, there is the following criteria of correlation-immune function: a Boolean function $f \in \mathcal{F}_n$ is correlation-immune function of m order, $0 < m \leqslant n$, if and only if for any vector $u \in V_n$, such that $1 \leqslant \mathrm{wt}(u) \leqslant m$, the equality $W_f(u) = 0$ performs.

Let A be $(n \times k)$-matrix over $\mathbb{F}_2$, and $f \in \mathcal{F}_k$. Let f^A be denoted as a function from $\mathcal{F}_n$, defined as $f^A(x) = f(xA)$. The order of algebraic degeneracy $\mathrm{AD}(f)$ of a Boolean function $f \in \mathcal{F}_n$ is the maximum possible value of $n - k$, where the integer k, $0 \leqslant k \leqslant n$ such that a function $g \in \mathcal{F}_k$ and $(n \times k)$-matrix A over $\mathbb{F}_2$ exist that there is an equality $f = g^A$. Functions with $\mathrm{AD}(f) > 0$ are algebraically degenerate. The set of all degenerate algebraic functions of n variables is denoted as $\mathrm{DG}(n) = \{f \in \mathcal{F}_n \mid \mathrm{AD}(f) > 0\}$. Nondegeneracy of a function $f \in \mathcal{F}_n$ (refer to [2]) is the following value:

$$\rho(f) = \mathrm{dist}(f, \mathrm{DG}(n)) .$$

3 Minimal correlation-immune functions

This section explains the meaning of k-minimal correlation-immune functions and provides a short overview of known properties of this object.

Suppose we want to define a sufficiently large set of functions that are guaranteed to have a cryptographic property such as correlation immunity of k-th order, $k \geqslant 1$. One of the simplest algebraic structures that allows the effective construction of its elements is a linear space. It is easy to see (for example, [1]) that functions $f, g \in \mathrm{CI}(n,k)$ such that $f \cdot g = \mathbf{0}$ (further functions with disjoint supports will be called *orthogonal*) hold $f \oplus g \in \mathrm{CI}(n,k)$. So the linear space L with the basis consisting of orthogonal functions $f_1, \ldots, f_r \in \mathrm{CI}(n,k)$ is a subset of set $\mathrm{CI}(n,k)$. Could this subspace be embedded in the larger space with the same property? This basis can be extended by a function $g = f_1 \oplus \ldots \oplus f_r \oplus \mathbf{1}$ that belongs to a set $\mathrm{CI}(n,k)$ at the specified above conditions. Its support doesn't disjoint with supports of other functions from the basis. Therefore, only these bases, which satisfies the condition $f_1 \oplus \ldots \oplus f_r = \mathbf{1}$, will be considered further. The subspace which includes the subspace L and preserves the property of being the subset of set $\mathrm{CI}(n,k)$ can be constructed not only by adding functions but also due to the decomposition of existing functions f_i into a sum of orthogonal functions $f_i', f_i'' \in \mathrm{CI}(n,k)$. Then the construction of superspace for L function f_i is excluded from the basis of L and functions f_i', f_i'' are added. The basis of space L for which it is impossible to construct a superspace by this way consists of functions $f \in \mathrm{CI}(n,k)$, which could't be represented as a sum of functions $f', f'' \in \mathrm{CI}(n,k)$ such that $f' \cdot f'' = \mathbf{0}$ and $f' \oplus f'' = f$. Such functions will be called *k-minimal correlation-immune functions* (in this paper such functions will be called k-minimal for short).

The spaces with the basis consisting of mutually orthogonal functions also useful that the Walsh coefficient (except for the corresponding zero-argument) of the sum of any basis functions is equal to the sum of Walsh coefficients of these functions. In the general case it is necessary to calculate the convolution of Walsh coefficients of summand functions (see [7]). As mentioned above the apparatus of the Walsh coefficients is a powerful

tool for the study of cryptographic properties of Boolean functions. This property of the considering spaces is used in Section 4.2 by reducing the problem of constructing a resilient function of a given order to the problem of solving a system of linear pseudo boolean equations.

A special case of k-minimal function was first investigated in the work [1]. In this paper the notion of 1-minimal function was introduced in order to investigate the structure of set of correlation-immune functions in generally that is set $\mathrm{CI}(n)$. In this work the following inequalities have been proven for 1-minimal function f: $\mathrm{cor}(f) \leqslant 2$, where equality is achievable (constructed example of a function of 7 variables), and for $n \geqslant 4$ the inequality $\mathrm{wt}(f) < 2^{n-1}$ is true. Also the exact formula for the number of minimal functions of weight 4 has been proven. In the paper [4] the number of 1-minimal functions of 4 and 5 variables has been estimated (32 and 1240 respectively), and also their classification under the group Jevons have been composed (see [7]).

4 Construction of resilient functions

This section describes a method of construction the linear space of functions embedded in the set $\mathrm{CI}(n,k)$, and reduction of search problem a resilient function of a given order in this space.

4.1 Construction of 1-minimal functions with a given number of variables.

The obvious method of constructing the above-mentioned space $L \subset \mathrm{CI}(n,k)$ with basis consisting of k-minimal functions $f_1, \ldots, f_r$, $f_i \cdot f_j = \mathbf{0}$, $f_1 \oplus \ldots \oplus f_r = \mathbf{1}$ is decomposition of function $\mathbf{1}$ into a sum of such functions. Because of $\mathrm{cor}(\mathbf{1}) = n$ this decomposition exists for any k. Except in special cases of decomposition on 1-minimal functions with weight 2 and 4 (all basic functions are algebraically degenerate in this case (see [1])) the best known method of its construction is a brute force, when the subset V_n forming k-minimal functions are found. The complexity of brute force depends essentially on the capacity of the support of function

for which the decomposition is searched.

This section describes an iterative method of constructing a k-minimal functions of a given number of variables. The method is especially effective for the construction of a 1-minimal functions. But in some cases this method allows to construct k-minimal functions for $k > 1$, however, the more efficiency decreases, the more the value of k. The minimal functions constructing by this method allow to discard vectors, which are their supports, from the support of function $\mathbf{1}$. The weight of the function, which has been composed, is reduced, which leads to the fact that the brute force method becomes feasible in practice.

Since, as mentioned above, the proposed method is especially effective for $k = 1$, a description of it is given for this case.

In the first step of the proposed method 1-minimal functions of a small number of variables are constructed. These functions can be found using a simple brute force computing or taking from known sets (see, for example, [4]). In the second step the number of variables of 1-minimal functions constructed in the first step increases.

Now describe a method of increasing the number of variables 1-minimal functions.

The truth table of function $f \in \mathcal{F}_n$ is called the matrix T_f of order $\mathrm{wt}(f) \times n$, the rows of this matrix are vectors from $supp(f)$ lexicographically-ordered. For example, for function $f(x_1, x_2, x_3) = x_1x_2 \oplus x_3 \in \mathcal{F}_3$

$$T_f = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

Let be $\mathcal{F}_n^w = \{f \in \mathcal{F}_n | \mathrm{wt}(f) = w\}$. For any $w \in \{1, \ldots, 2^n\}$ define the map $\mathrm{AC}^{(w)}$:

$$\mathrm{AC}^{(w)} : \mathcal{F}_n^w \times V_w \times \{1, \ldots, n+1\} \mapsto \mathcal{F}_{n+1}^w.$$

The function $g = \mathrm{AC}_{v,i}^{(w)}(f) = \mathrm{AC}^{(w)}(f, v, i)$ is defined as follows. The matrix $\mathrm{wt}(f) \times (n+1)$ is formed by adding vector v of dimension w in the truth table T_f as i-th column. Whereas i-th and following columns T_f

are shifted to the right. The rows of formed matrix is support of function g. If $i = n + 1$, then the column is added to the end of the table.

The following statements, proof of which is given in the Appendix, are true.

Theorem 1. *Let* $f \in \mathrm{CI}(n)$ *and* $w = \mathrm{wt}(f)$. *Then for any* $v \in V_w$, *such that* $\mathrm{wt}(v) = w/2$, *and for any* $i \in \{1, \ldots, n+1\}$ *the following is true* $g = \mathrm{AC}_{v,i}^{(w)}(f) \in \mathrm{CI}(n+1)$.

Let $\mathrm{MCI}(n, k)$ be a set of k-minimal functions of n variables.

Theorem 2. *Let* $f \in \mathrm{MCI}(n, 1)$ *and* $w = \mathrm{wt}(f)$. *Then for any* $v \in V_w$, *such that* $\mathrm{wt}(v) = w/2$, *and for any* $i \in \{1, \ldots, n+1\}$ *the following is true* $g = \mathrm{AC}_{v,i}^{(w)}(f) \in \mathrm{MCI}(n+1, 1)$.

By Theorem 2 the map AC can be used for increasing the number of variables of 1-minimal functions. The presence of variable parameters v and i allow to get the set of minimal functions of a larger number of variables using the "started" function $f \in \mathrm{MCI}(n, 1)$.

For construction k-minimal functions, $k > 1$, using the map AC, the condition $\mathrm{wt}(v) = w/2$ is no longer sufficient. The problem of developing the effective methods for construction k-minimal functions for $k > 1$ is not solved at the moment.

In addition to use in the synthesis, Theorems 1 and 2 can be used for research of the structure of minimal functions. The results of such researches will be presented in the paper [5].

4.2 Search function with a given order of correlation immunity

This section describes the reduction of search problem of resilient function with a given order to the problem of solving a system of linear pseudo boolean equations in the space that are formed mutually orthogonal k-minimal functions.

Let $L \subset \mathrm{CI}(n, k)$ be a linear space with basis $f_1, \ldots, f_r \in \mathrm{CI}(n, k)$ and $f_i \cdot f_j = \mathbf{0}$ for any $i \neq j$. Suppose we want to find $k+m$*-resilient function* g in the space L, in other words this function must satisfy the conditions

$\mathrm{wt}\,(g) = 2^{n-1}$ and $\mathrm{cor}\,(g) \geqslant k+m$, $m \geqslant 1$, or prove that this function doesn't exist in the space L.

Because of the basis of the space L consists of mutually orthogonal functions for any u, $\mathrm{wt}\,(u) > 0$, and for any $g = b_1 \cdot f_1 \oplus \ldots \oplus b_r \cdot f_r$, $b_1, \ldots, b_r \in \mathbb{F}_2$, the following equation is true:

$$W_g(u) = b_1 \cdot W_{f_1}(u) + \ldots + b_r \cdot W_{f_r}(u).$$

In order to g be CI-function of $k+m$-th order it is necessary and sufficient that for any u, $1 \leqslant \mathrm{wt}\,(u) \leqslant k+m$, the equality $W_g(u) = 0$ is true. Since $f_i \in CI(n,k)$, so $W_{f_i}(u) = 0$ for any u such that $1 \leqslant \mathrm{wt}\,(u) \leqslant k$. Consequently similar equations hold for g.

In case to check whether the function g satisfies the condition $\mathrm{cor}\,(g) \geqslant k+m$, it is sufficient to verify correctness of $\binom{n}{k+1} + \ldots + \binom{n}{k+m}$ equations

$$b_1 \cdot W_{f_1}(u) + \ldots + b_r \cdot W_{f_r}(u) = 0$$

for all u, $k+1 \leqslant \mathrm{wt}\,(u) \leqslant k+m$. Further, since the supports of the functions of the basis do not intersect, so condition $\mathrm{wt}\,(g) = 2^{n-1}$ is true if the following equality is true

$$b_1 \cdot \mathrm{wt}\,(f_1) + \ldots + b_r \cdot \mathrm{wt}\,(f_r) = 2^{n-1}.$$

Set forth above shows that it is sufficient to find $0,1$-solutions $(b_1, \ldots, b_r)$ of system of $\binom{n}{k+1} + \ldots + \binom{n}{k+m} + 1$ linear equations

$$\begin{cases} b_1 \cdot W_{f_1}(u) + \ldots + b_r \cdot W_{f_r}(u) = 0, \text{ for } u, \text{ such that } k+1 \leqslant \mathrm{wt}\,(u) \leqslant k+m; \\ b_1 \cdot \mathrm{wt}\,(f_1) + \ldots + b_r \cdot \mathrm{wt}\,(f_r) = 2^{n-1}. \end{cases}$$

in order to find $k+m$-resilient function $g \in L$.

Since the target solutions are only $0,1$-solutions so this system is a pseudo boolean system. This reduction allows to use different methods of solving such system for construction a $k+m$-resilient Boolean function. In particular we can effectively check whether the building space L contains resilient functions of given order or not by checking the compatibility of this system of equations over the field of real numbers. The problem of estimating of probability of passing such checking by system without $0,1$-solutions is open.

5 Using minimal functions for analysis of neighbourhoods of known functions

The approach described above can be used not only for the construction of cryptographic functions "from scratch". Suppose that some function $f \in \mathcal{F}_n$, which depends on a given number of arguments, but satisfies only a part of the necessary requirements, is already known. Such case can arise due to the appearance of a previously unknown method of cryptanalysis, in relation to which the function f is not resistance.

The function f, with condition $\mathrm{cor}(f) \geqslant k$, can be a base for constructing a basis $\mathcal{M}$ consisting of k-minimal functions. This basis will be constructed in such a way that f contains in the space L, forming by functions from $\mathcal{M}$. For this purpose it's sufficient to create the basis $\mathcal{M}$ from orthogonal k-minimal functions, which are decomposition of functions f and $f \oplus \mathbf{1}$.

For searching of functions from L, which satisfy the specified requirements, the most of the functions are taken from decomposition f , and a few functions are taken from decomposition $f \oplus \mathbf{1}$, which complement already taken function to the desired weight. Thus the resulting functions are a relatively short distance from f, that is, they don't leave the determined "neighbourhood" of function f. At the same time the order of correlation-immunity of resulting functions isn't less than k.

The effectiveness of the above mentioned approach to construct the basis from k-minimal functions currently has no rigorous justification, that is, it is not known whether constructed spaces by this way are guaranteed to contain a lot of "good" functions. However the following section describes the results obtained by this method.

6 The results of applying the proposed methods

This section briefly lists the specific results of the approaches and methods that have been described above. The missing details, such as, for example, the vectors of values for considering functions are given in the Appendix.

Consider the function $f^4_{10,2} \in \mathcal{F}_{10}$ from the paper [9], which will be denoted by f_T. This function has the following parameters:

$\mathrm{wt}(f_T) = 512$	$\mathrm{cor}(f_T) = 6$	$\deg(f_T) = 3$	$\mathrm{nl}(f_T) = 384$	$\mathrm{nd}(f_T) = 0$

The trouble with this function in terms of the use of cryptographic schemes is algebraically degenerate of it ($\mathrm{nd}(f_T) = 0$). Using the method described in the Section 5 (functions f_T and $f_T \oplus \mathbf{1}$ were decomposed on 128 1-minimal functions with the weight 4), the new function g_T is constructed with the following parameters:

$\mathrm{wt}(g_T) = 512$	$\mathrm{cor}(g_T) = 2$	$\deg(g_T) = 7$	$\mathrm{nl}(g_T) = 360$	$\mathrm{nd}(g_T) = 8$

This function is not algebraically degenerate, and decrease of correlation-immunity $\mathrm{cor}(g_T)$ and nonlinearity $\mathrm{nl}(g_T)$ is offset by a significant increase of the parameter $\deg(g_T)$. At the same time $\mathrm{dist}(f_T, g_T) = 40$.

The similar method was applied to the filter function $f_c \oplus \mathbf{1}$, where f_c is used in stream cipher LILI128 [14]. This function of 10 variables has the following parameters:

$\mathrm{wt}(f_c) = 512$	$\mathrm{cor}(f_c) = 3$	$\deg(f_c) = 6$	$\mathrm{nl}(f_c) = 480$	$\mathrm{nd}(f_c) = 80$

After using the method described in the Section 5 the new function g_c is constructed with the following parameters:

$\mathrm{wt}(g_c) = 512$	$\mathrm{cor}(g_c) = 3$	$\deg(g_c) = 6$	$\mathrm{nl}(g_c) = 480$	$\mathrm{nd}(g_c) = 112$

The function g_c is not worse the function f_c on any parameter, and the value of algebraically degenerate $\mathrm{nd}(g_c)$ greatly exceeds $\mathrm{nd}(f_c)$. At the same time $\mathrm{dist}(f_c \oplus \mathbf{1}, g_c) = 288$. Thus with the help of the proposed approaches the function g_c is constructed and the using of it instead of the function f_c in the cipher LILI128 allow to improve its cryptographic properties.

In conclusion it is given an example of constructing of a function without the use of known "good" functions. In the space consisting from 1-minimal functions of 10 variables with the weight 2 the 7-resilient function f_2 is found with the following parameters:

$\mathrm{wt}(f_2) = 512$	$\mathrm{cor}(f_2) = 7$	$\deg(f_2) = 2$	$\mathrm{nl}(f_2) = 256$	$\mathrm{nd}(f_2) = 0$

This function achieves the upper bound for nonlinearity for 7-resilient functions (see, for example, [9]). Functions with the same parameters can be constructed by the method described in the paper [9].

7 Open problems

The concept of a minimal correlation-immune function has been introduced recently, so nowadays there are a number of open issues related to the properties of these functions. The following some problems are the most important in the context of the main theme of this article in the authors' opinion.

1. Searching of efficient criteria for approving the k-minimality of this function.
2. Developing a method of increasing the number of variables k-minimal functions, $k > 1$, is as effective as for the $k = 1$.
3. Developing of efficient searching method of balanced functions with a given values of nonlinaruty/nondegeneracy/ algebraic immunity in the space generated by k- minimal functions.

8 Conclusion

In this paper the approach to the construction of Boolean functions with a given cryptographic parameters is given. This method bases on a combination of iterative and exhaustive search methods. On the basis of this approach a method of synthesis of a given order-resilient functions is developed, which uses minimal correlation-immune functions. Functions constructing by this method have not obvious structural features that distinguish them from a random function. In this paper the transformations are proposed, and it is proved that they can be used for realisation of the first step of the developing method. The reduction of search problem of resilient function with a given order on the second step of proposed method to the problem of solving a system of linear pseudo boolean equations is

described. Also the method for improving of method's characteristics with using known "good" cryptographic functions is proposed. Examples of the successful applying of this method are given. In particular the function is constructed, the use of which instead the filter function in the stream cipher LILI128 would improve its cryptographic properties.

The authors are grateful to Grigory Anatolyevich Karpunin for his valuable comments and constructive criticism.

References

[1] Alekseev E.K. "Some algebraic and combinatorial properties of correlation-immune Boolean functions" (in Russian), Discrete Math., 22:3 (2010), 110–126.

[2] Alekseev E.K. "Some measures of nonlinearity of Boolean functions" (in Russian), ADM, 2011, No.2, 5-16.

[3] Alekseev E.K., "Filtering generator attacks with function close to algebraically degenerate" (in Russian), Collection of papers of young scientists of CMC Faculty MSU, 2011, No.8, 19-32.

[4] Alekseev E.K., Karelina E.K. "Classification correlation-immune and minimal correlation-immune functions of 4 and 5 variables" (in Russian), Discrete Math, 27:1 (2015), 22–33.

[5] Karelina E.K. "The properties of minimal correlation-immune Boolean functions and methods of their construction" (in Russian) (be in preparation).

[6] Kuschinskaja L.A. "Research limits of applicability of some methods of cryptanalysis of stream ciphers that are constructed on the basis of shift registers", Diploma. MSU, CMC Faculty, 34 c., 2015.

[7] Logachev O.A., Salnikov A.A., Smyshlyaev S.V., Yashchenko V.V. "Boolean functions in coding theory and cryptology" (in Russian), URSS. ISBN 978-5-9710-0961-0, 576 c., 2015.

[8] GOST R 34.12–2015 "Information technology. Information security. Block ciphers". Federal Agency on Technical Regulating and Metrology, 2015.

[9] Tarannikov Yu.V. "Correlation-immune and resilient Boolean functions" (in Russian), Mathematical Problems of Cybernetics, 2002, T. 11, 91–148.

[10] Siegenthaler T. "Decrypting a Class of Stream Chipher Using Ciphertext Only", IEEE Trans. on Computers, Vol. C-34(1)., 81-85, 1985.

[11] Meier W., Staffelbach O. "Fast correlation attacks on certain stream ciphers", Journal of Cryptology, vol. 1, 1989, pp. 159-176.

[12] Courtois N., Meier W. "Algebraic attacks on stream ciphers with linear feedback". Proceedings of EUROCRYPT 2003, LNCS, vol 2656, pp. 346–359.

[13] Matsui M. "Linear Cryptanalysis Method for DES Cipher", Advanced in Cryptology – EUROCRYPT'93. Lect. Notes in Comp. Sci., Springer, 1994. V. 765. P. 386-397.

[14] Dawson E., Clark A., Golich J., Millan W., Penna L., Simpson L. "The LILI-128 keystream generator". Proc. of first NESSIE workshop, pp. 1-14, 2000.

9 Appendix

9.1 The vectors' values of function from Section 6

All vectors' values are represented in hexadecimal.

$f_T =$ c53a5ac33ca5a35ca35c3ca55ac3c53a3ac5a53cc35a5ca35ca3c35aa53c3ac5
a35c3ca55ac3c53a5ca3c35aa53c3ac5c53a5ac33ca5a35c3ac5a53cc35a5ca3
3ac5a53cc35a5ca3c53a5ac33ca5a35c5ca3c35aa53c3ac5a35c3ca55ac3c53a
5ca3c35aa53c3ac53ac5a53cc35a5ca3a35c3ca55ac3c53ac53a5ac33ca5a35c

$g_T =$ 3ac596c33ca5a35ca35c3ca55ac3c53a3ac5a53cc35a5ca35ca3c35aa53c3ac5
a35c3ca55ac3c53a5ca3c35aa53c3ac5c53a5ac33ca5a35c3ac5a53cc35a5ca3
3ac5a53cc35a5ca3c53a5ac33ca5a35c5ca3c35aa53c3ac5a35c3ca55ac3c53a
5ca3c35aa53c3ac53ac5a53cc35a5ca3a35c3ca55ac3c53ac53a5ac33c965ca3

$f_c =$ 3cc3c33c3cc3c33c3cc3c33cc33c3cc35aa5a55a5aa5a55a5aa5a55aa55a5aa5
669999666699996669699696699669960ff0f00ff00f0ff033cccc33cc3333cc
3c3cc3c3c3c33c3c3cc33cc3c33cc33c55aaaa55aa5555aa5a5aa5a5a5a55a5a
5aa55aa5a55aa55a666699999999666666996699996699666969696996969696

$g_c =$ 6969696996969696669966999966996666669999999966665aa55aa5a55aa55a
5a5aa5a5996666999696696996699669f00f0ff00ff0f00fcc3333cc33cccc33
c3c33c3c3c3cc3c3c33cc33c3cc33cc3aa5555aa55aaaa55a5a55a5a66999966
a55a5aa55aa5a55a5aa5a55a5aa5a55ac33c3cc33cc3c33c3cc3c33c3cc3c33c

$f_2 =$ 9669699669969669966969966996696699696699669699669969966996696996
9669699669969669699696699669696699696699669969966996696996696996
6996966996696996696699669969669969966996696699669969966996699669
6996966996696996696699669969669969966996696699669969966996699669

9.2 The proof of Theorem 1

Proof. To prove the theorem it suffices to show that $\mathrm{cor}\,(g) \geqslant 1$. For this it is necessary and sufficient that $W_g(u) = 0$ for all u of weight 1. Let e_i be a vector from V_n of weight 1, in which the unit is on i-th position. As for any i, $1 \leqslant i \leqslant n$, and for any $f \in \mathcal{F}_n$ it is true

$$W_f(e_i) = \sum_{x \in V_n} (-1)^{f(x) \oplus x_i} = \sum_{x \in V_n} (-1)^{x_i} - 2 \cdot \sum_{x \in supp(f)} (-1)^{x_i},$$

so the condition $W_f(e_i) = 0$ for all i is equivalent to that each column of the truth table is balanced. Therefore the adding a more balanced column will not lead to a breach of this condition, ie, the resulting function will be correlation-immune of the first order. □

9.3 The proof of Theorem 2

Proof. From Theorem 1 it follows that $g \in \mathrm{CI}(n+1)$. It remains to prove that g is 1-minimal function.

Assume that this is not true. Then from the support of function g the subset can be selected, which will be a support of function h with $\mathrm{cor}\,(h) \geqslant 1$. Then the i-th column is deleted from the table T_h (it corresponded to the adding column v to the table T_f). The result is a matrix with non-repeated rows, because these rows correspond to certain rows of the matrix T_f. Arrange these rows lexicographically the new table truth of some function h', where $supp(h') \subset supp(f)$, is obtained. Further we use the arguments from the proof of Theorem 1. The columns $T_{h'}$ are balanced because they are columns of the table T_h, and $\mathrm{cor}\,(h) \geqslant 1$. So $\mathrm{cor}\,(h') \geqslant 1$, and this contradicts the minimality of the original function f. □

On the construction of generalized approximations for one filter generator key recovery method

Evgeny Alekseev, Liudmila Kushchinskaya

Abstract

This paper presents the findings of a study into the possibility of building generalized approximations that can be used to recover the key of a filter generator. The study assesses the characteristics of the most general method for building such approximations for general type filter generators.

The study was carried out with the support of the Russian Foundation for Basic Research; project 16-01-00470 A.

Keywords: boolean function, filter generator

1 Introduction

One approach often used in cryptographic analysis is the approximation of Boolean functions (mappings) with special form functions. Prominent examples based on this approach are the correlation method proposed by Siegenthaler [1] and the linear method proposed by M. Matsui [2]. Another example of this approach is the method proposed in [3] based on using algebraically degenerate functions. However, this method has a number of limitations: for example, it cannot be used when the filter generator key has a length that is a prime number.

An extension of the approach based on approximating algebraically degenerate functions is the generalized approximation method proposed in [5]. In this type of approximation the idea is to find a sufficient number of planes that a specific value of the function is predominant on. This type of approximation makes it possible to recover the key of the filter generator with complexity that in some cases may reach the square root of the power of the set of keys.

This article considers the possibility of building a generalized approximation. It presents a brief description of the method, assesses its key characteristics, and looks at possible use cases. The article presents the following finding. We assess the characteristics of the general method for building generalized approximations derived from a model with random sets for general form generators. The article presents numerical experiments to verify the accuracy of the proposed model. The findings presented in this method make it possible to assess when the method proposed in [5] can be applied.

2 Basic concepts

Let $\mathbb{F}_2$ be a field of 2 elements. Let $V_n = \mathbb{F}_2^n$ be an affine space of vectors of length n with components from $\mathbb{F}_2$. *The Boolean function* f of n variables will then be mapping $f : V_n \to \mathbb{F}_2$.

Then, we can define a *filter generator* as a function built on the basis of linear mapping $A : V_n \to V_n$ and the Boolean function $f \in \mathcal{F}_n$. Some open text $x = x_0, x_1, \ldots, x_n, \ldots$ then gets encrypted on key $u^* \in V_n$ with a stream cipher built on the basis of the filter generator with the encryption happening in the following manner. Every bit of the encrypted text $c = c_0, c_1, \ldots, c_n, \ldots$ satisfies the ratio $c_i = x_i \oplus z_i, i \geqslant 0$, where $z_i = f(A^i u^*)$ — the output bit of the filter generator at the i-th step.

For a filter generator, *the trajectory* will refer to the three values $\text{Traj} =< m, \mathbb{L}, \mathbb{T} >$, where $m \in \mathbb{N}$ is the length of the trajectory, $\mathbb{L} = \{L_i - \text{is a plane in } V_n | i = \overline{1,m}\}$, $\mathbb{T} = \{t_i | t_i \in \mathbb{N}, t_i > t_{i-1}, i = \overline{1,m};\ t_1 = 0\}$, such that

$$L_i = A^{t_i - t_{i-1}}(L_{i-1}),\ \ t_i, t_{i-1} \in \mathbb{T}, i = \overline{2,m}.$$

The characteristic of trajectory $\text{Traj} =< m, \mathbb{L}, \mathbb{T} >$ is a pair of sets $(\mathbb{P}, C)$, where $\mathbb{P} = \{p_i | p_i \in \left(\frac{1}{2}; 1\right], i = \overline{1,m}\}$, $C = \{c_i | c_i \in \mathbb{F}_2, i = \overline{1,m}\}$, p_i is the probability that the value of the filter function f is the same as constant c_i in plane $L_i, i = \overline{1,m}$ provided that vector $v \in L_i$ is picked randomly with each value having the same probability of being selected.

The set of all the trajectories $\{\text{Traj}^{(i)}\}$ will then be referred to as *the generalized approximation* of filter function f in the generator with linear

mapping A.

The starting set $\mathbb{L}_{start}$ of the generalized approximation is the collection of sets $\{L_1^{(i)}\}$ from each trajectory.

Definition 2.1. *[4] Let's assume we have a sample of n n elements without a return from the final collection of size N, and let assume that D of these elements have a given property. We can then say that random value x has a hyper-geometric distribution with parameters* N, D, n *(*$x \sim HG(D, N, n)$*) if the following equation holds true*

$$Pr[x = k] = \frac{\binom{D}{k}\binom{N-D}{n-k}}{\binom{N}{n}}.$$

3 The key recovering method and its properties

3.1 Description of the method

Let's assume that for a given generator a generalized approximation for the filter function has been built $\{\text{Traj}^{(i)}, i = \overline{1, s}\}$, and it comprises s different trajectories (figure 1).

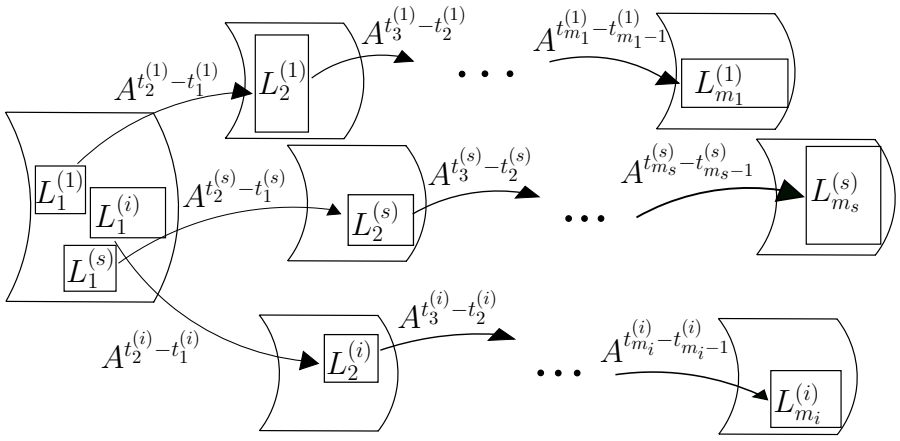

Figure 1: General case

From now on we're going to omit the upper indexes when talking about one specific trajectory. Let's assume we've selected some $L \in \mathbb{L}_{start}$. Let's fix the parameters corresponding to this trajectory: m, $\mathbb{L} = \{L_i | L_i -$

plane in $V_n, i = \overline{1,m}\}$, $\mathbb{P} = \{p_i | p_i \in \left(\frac{1}{2}; 1\right], i = \overline{1,m}\}$, $C = \{c_i | c_i \in \mathbb{F}_2, i = \overline{1,m}\}$; $\mathbb{T} = \{t_i | t_i \in \mathbb{N}, t_i > t_{i-1}, i = \overline{1,m};\ t_1 = 0\}$ the numbers of the steps of the generator during which the constant value predominates in the corresponding plane $L_i, i = \overline{1,m}$. Let's build a vector

$$w = (c_1 \oplus \widetilde{z}_1, \ldots, c_m \oplus \widetilde{z}_m),\ \ \widetilde{z}_i = z_{t_i},\ \ i = \overline{1,m}.$$

If the original key is $u^* \notin L$, then the weight of the vector w is close to $m/2$. However, if $u^* \in L$, then the vector weight w is different from $m/2$, and the degree to which it deviates from $m/2$ depends on the quality of the approximation, i.e. on the characteristics of the trajectory. Let's assume that in order to differentiate between these two cases there is some deciding rule of the form $F(L) \geqslant 0$ (maximum likelihood type), that allows us to accept or reject the trajectory Traj (plane $L \in \mathbb{L}_{start}$). The deciding rule F is constructed in accordance with the principle of maximum plausibility.

Let $\hat{L} = \mathbb{L}_{start}$, $M = V_n \setminus \left(\bigcup_{L \in \mathbb{L}_{start}} L\right)$.

Description of the algorithm:

1. Stage one (selecting the «correct» generalized approximation trajectories). $\widetilde{L} := \emptyset$.

   1.a) If $\hat{L} = \emptyset$, then go to stage two. Otherwise select a random element L_i from the set $\hat{L}$; $\hat{L} := \hat{L} \setminus \{L_i\}$.

   1.b) Build vector $w \in V_{m_i}$, as was demonstrated above. If the inequality $F(L_i) \geqslant 0$, holds then assume $\widetilde{L} = \widetilde{L} \cup \{L_i\}$. Go to step 1.a).

2. Stage two (thorough testing of the «correct» trajectories).

   2.a) If $\widetilde{L} = \emptyset$, then go to stage three, else select Y from set $\widetilde{L}$; $\widetilde{L} := \widetilde{L} \setminus \{Y\}$.

   2.b) If $Y = \emptyset$, then go to step 2.a). Otherwise select $v \in Y$; $Y := Y \setminus \{v\}$.

   2.c) If $f(A^i v) = z_i$ for any $i = \overline{0, N-1}$, then return v and stop, otherwise go to step 2.b).

3. Stage three (viewing set M).

3.a) If $M = \emptyset$, then quit without returning anything, otherwise select $u \in M$; $M := M \setminus \{u\}$.

3.b) If $f(A^i u) = z_i$ for any $i = \overline{0, N-1}$, then return u as the result and exit, otherwise go to 3.a).

3.2 Characteristics of the method

The general characteristics of the method are proved in [5]. We also estimate complexity here for the following simple case.

Let's assume that a generalized approximation has been built for a generator and that it has the following characteristics. Planes from $\mathbb{L}_{start}$ do not intersect with each other and $\bigcup_{L \in \mathbb{L}_{start}} L = V_n$, while $dim(L) = k, \forall L \in L_{start}$. Thus, $s = 2^{n-k}$ и $\{m_i = m, i = \overline{1, s}\}$, $\{p_j^{(i)} \geqslant p, i = \overline{1, s}, j = \overline{1, m}\}$, $\{|L_j^{(i)}| = 2^k, i = \overline{1, s}, j = \overline{1, m}\}$.

Let α, β be the type I and type II errors for the trajectory. Meaning that α is the probability that a false trajectory will be selected, the starting set of which does not contain the key; β is the probability that the true trajectory will be rejected.

Then the complexity of recovering the key is

$$D = s + s \cdot |L| \cdot \alpha + 2^{-n} \cdot s \cdot |L|^2 \cdot (1 - \alpha - \beta) = 2^{n-k} + 2^n \cdot \alpha + 2^k \cdot (1 - \alpha - \beta).$$

In the most common case $\alpha \ll \beta$, so α could be neglected in the third addendum.

Examples of applying the method to the LILI-128 [10] cipher can be found in [5].

4 A search algorithm for constructing generalised approximations

In this section, we describe an exhaustive search method for constructing a generalised approximation of a filter function and offer a model with random sets to estimate its characteristics. We also offer an experimental proof for the adequacy of the proposed model.

Let's assume we need to construct a generator generalised approximation such that the key recovery method based on it has a complexity Q and reliability π_0 while requiring the minimal possible amount of the generator's output sequence to run. We will construct approximations for which none of the planes in L_{start} mutually intersect and have the same capacity that significantly exceeds the capacity of set M.

4.1 Description of the method

When constructing our approximation, we assume that for all the planes that make it up, the following parameters are the same.

- Parameter $k \in \{1, 2, \ldots, n-1\}$ is the number of dimensions for the plane in the trajectory with the capacity of the plane being designated as $N = 2^k$.
- Parameter $\delta \in \{1, 2, \ldots, N\}$ will be responsible for the minimal predominance of one constant or another in the plan, then $T_0 = \frac{N}{2} - \frac{\delta}{2}$ is the boundary for the number of zero values and $T_1 = \frac{N}{2} + \frac{\delta}{2}$ is the boundary for ones. In other words, if we use S_N to designate the weight of function f in a plane that is part of our approximation, then either $S_N < T_0$ (the plane has enough zeroes of the function), or $S_N > T_1$ (the plane has enough ones of the function).

The search method for constructing the trajectory for a generalised approximation that we consider here can be described as follows. We can then select a random plane L_0 with k dimensions, i.e. a plane with a capacity of $N = 2^k$. Then for each $i = 0, 1, 2, \ldots$ we follow the following algorithm:

- If in plane L_i filter function f equals 1 a certain number of times different than $N/2$ by a large enough value then we add it to the trajectory we're constructing.
- $L_{i+1} := A(L_i)$.
- We then repeat the steps above until we've achieved the desired length of the trajectory.

4.2 Mathematical model

We suggest using the following model to study the characteristics of the method we just described. Let's assume that plane L_0 is a random sent while the image of random set L_i (resulting from linear transformation A) is selected randomly and independently from L_i. Then the process of constructing one trajectory can be modelled as a selection of random sets with the same capacity 2^k from V_n and calculating the weight of function f in those sets.

Let $p(\delta, k)$ be the probability that a random plane gets selected for the trajectory. Let N_1 be the length of the trajectory, $N_2 = \frac{N_1}{p(\delta,k)}$ be the number of steps in the algorithm that must be completed to construct trajectory with length N_1. It should be noted that N_2 equals the volume of the output sequence of the generator that is needed to recover the key with the given parameters and that we need to minimise.

Since for any plane in the trajectory some value of the function can be observed in more than $(N/2 + \delta/2)$ points, then to simplify our calculations we can assume that the predominance in accuracy equals $1/2 + \delta/2N$. In this case the values N_1 and N_2 will be the same for each trajectory.

It should be noted that if parameter k approaches zero it means that the capacity of N planes in the trajectory diminishes, which means that the method approaches an exhaustive search of all the vectors in the key space in the first stage of the algorithm. At the same time, if $k \to n$, then an increase in the capacity of a plane in the trajectory will mean that the processing time of the second stage of the key recovery algorithm is going to approach the processing time of an exhaustive search.

It should also be noted that for a fixed k with $\delta \to 0$ the probability of accepting a random plane increases, but at the same time there is an increase in value N_1 needed to achieve the target reliability of the method. If $\delta \to N$, the length of trajectory N_1 decreases but so does the probability of accepting the plane because the expected predominance in the plane goes up.

Thus, we end up with the following task on our hands: find such values for the parameters (δ, k) that minimise value N_2 for a given complexity and reliability of the method.

4.3 Characteristics of the method

Let L be a random set of vectors from V_n with a capacity of $N = 2^k$. Selecting a set like this can be modelled as a random sample of capacity N taken without replacement from a finite population consisting of 2^n elements.

Let's consider random value $S_N = \sum\limits_{v \in L} f(v)$ — the number of points in set L for which the function equals one. The value S_N has hypergeometric distribution $HG(2^{n-1}, 2^n, 2^k)$. Since $HG(D, S, m) \approx Bin(m, D/S)$ when $S \to \infty$, we can assume that $S_N \sim Bin(N, \frac{1}{2})$.

Since $Bin(n, p) \approx N(np, npq)$ for large values of n where $N(np, npq)$ is a normal distribution with mathematical expectation np and dispersion npq, then $Bin(N, \frac{1}{2}) \approx N(\frac{N}{2}, \frac{N}{4})$, $p = q = \frac{1}{2}$.

Let's find the probability of first and second type errors arising in the key recovery for the trajectories. Since the parameters k and δ are the same for all trajectories, the error probabilities will be the same.

- $\beta = 1 - \pi_0$ is the probability of second type errors.
- Since $Q = 2^{n-k} + \alpha \cdot 2^{n-k} \cdot 2^k + (1 - \beta) \cdot 2^k$, then the probability of first type error is $\alpha = 2^{-n} \cdot \left(Q - 2^{n-k} - \pi_0 \cdot 2^k\right)$.

As per method, for each trajectory the following vector must be constructed

$$w = (c_1 \oplus \widetilde{z}_1, \ldots, c_{N_1} \oplus \widetilde{z}_{N_1}), \;\; \widetilde{z}_i = z_{t_i}, \;\; i = \overline{1, N_1}.$$

Meanwhile, if the key is in the starting set of this trajectory then $Pr[w_i = 0] > q_1 = \frac{1}{2} + \frac{\delta}{2N}$, and otherwise $Pr[w_i = 0] = q_0 = \frac{1}{2}$. For statistical differentiation between two Bernoulli distributions with success probabilities of q_0 and q_1 and first and second type error probabilities of α and β respectively, we're going to need a trajectory length of [6]

$$N_1 \approx \frac{(u_\alpha \sqrt{q_0(1 - q_0)} + u_\beta \sqrt{q_1(1 - q_1)})^2}{(q_1 - q_0)^2},$$

where u_α, u_β are the quantiles of a standard normal distribution.

Let's derive an expression for the probability that a random set will be included in the trajectory. The following expression holds

$$Pr\left[T_0 \le S_N \le T_1\right] = Pr\left[\frac{T_0 - Np}{\sqrt{Npq}} \le \frac{S_N - Np}{\sqrt{Npq}} \le \frac{T_1 - Np}{\sqrt{Npq}}\right] =$$

$$= Pr\left[-\frac{\delta}{\sqrt{N}} \le \frac{S_N - Np}{\sqrt{Npq}} \le \frac{\delta}{\sqrt{N}}\right] = \Phi\left(\frac{\delta}{\sqrt{N}}\right) - \Phi\left(-\frac{\delta}{\sqrt{N}}\right) =$$

$$= 2\Phi\left(\frac{\delta}{\sqrt{N}}\right) - 1,$$

where $\Phi(y) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{y} e^{-\frac{x^2}{2}} dx$ is a distribution function for a normally distributed random value. From this, we can derive the following:

$$p(\delta, k) = 1 - Pr[T_0 \le S_N \le T_1] = 2\left(1 - \Phi\left(\frac{\delta}{\sqrt{N}}\right)\right).$$

Then the volume of the output sequence of the generator that we need to construct the trajectory is:

$$N_2 = \frac{N_1}{p(\delta, k)} \approx \frac{\left(u_\alpha + u_\beta \cdot \sqrt{1 - \left(\frac{\delta}{N}\right)^2}\right)^2 \cdot \left(\frac{N}{\delta}\right)^2}{2\left(1 - \Phi\left(\frac{\delta}{\sqrt{N}}\right)\right)}.$$

Let's now move to continuous variable $t = \delta/\sqrt{N}, t \in (0; \sqrt{N}]$. Then

$$N_2 \approx \frac{N}{2} \cdot \left(u_\alpha + u_\beta \sqrt{1 - \frac{t^2}{N}}\right)^2 \cdot \frac{1}{t^2(1 - \Phi(t))}.$$

Let's consider the most interesting case for the task at hand, namely the case where $Q \ll 2^n$, and π_0 takes values from a range natural for practical uses (for instance, $\pi_0 = 1/2$ or $\pi_0 = 1/10$). Assuming that $\alpha \ll \beta$ and, consequently, $u_\alpha \gg u_\beta$. we have then to assume that $N_2 \approx \frac{N}{2} \cdot u_\alpha^2 \cdot \frac{1}{t^2(1-\Phi(t))}$.

Let's find the values of variable t for which maximum is achieved for $f(t) = t^2(1 - \Phi(t)), t \in (0; \sqrt{N}]$. By setting the derivative $f'(t)$ to zero, we get the equation

$$2(1 - \Phi(t)) = t \cdot \frac{1}{\sqrt{2\pi}} \cdot e^{-\frac{t^2}{2}},$$

which is equivalent to

$$\frac{t}{2} = \frac{1 - \Phi(t)}{\varphi(t)},$$

where $\varphi(t) = \frac{1}{\sqrt{2\pi}} e^{-t^2/2}$.

The expression $R(t) = \frac{1-\Phi(t)}{\varphi(t)}$ is known as the Mills ratio [7]. Function $R(t)$ monotonously decreases [8], which means that the equation has just one solution. Getting a formula for the root of this equation is impossible, but we can estimate it for any given degree of precision. Here we're going to calculate the root to a precision of 0.00001: $t_0 = 1.19061$.

Thus, $\delta \approx \lceil t_0 \cdot \sqrt{N} \rceil$. Value N_2 in the minimum:

$$N_2 \approx \frac{N}{2} \cdot u_\alpha^2 \cdot C_\Phi,$$

where $C_\Phi = \frac{1}{t_0^2(1-\Phi(t_0))} \approx 6.03442$.

Since for small α $u_\alpha \approx \sqrt{-ln(2\pi\alpha^2)}$ will hold, u_α will vary by no more than n for valid k. At the same time, N increases exponentially over k. Thus N_2 reaches the minimum at the minimal possible k. Valid $k \in \{1, 2, \ldots, n-1\}$ are those for which $\alpha = 2^{-n} \cdot \left(Q - 2^{n-k} - \pi_0 \cdot 2^k\right) > 0$. The minimal possible k can be defined as follows

$$k = \left\lceil log_2 \left(\frac{Q - \sqrt{Q^2 - \pi_0 2^{n+2}}}{2\pi_0} \right) \right\rceil.$$

The values of the functions in the minimum are as follows:

- $N_1 = (u_\alpha)^2 \cdot \left(\frac{N}{\delta}\right)^2 = \left(\frac{u_\alpha}{t_0}\right)^2 \cdot N$,
- $N_2 = N \cdot u_\alpha^2 \cdot \frac{C_\Phi}{2}$;

The table with parameters k, δ and values of N_1, N_2 for $n = 128$, $\pi_0 = 1/2$ and for different values of complexity is provided below.

Method characteristics

	k	δ	N_1	N_2
$Q = 2^{70}$	59	2^{30}	2^{65}	2^{67}
$Q = 2^{80}$	49	2^{25}	2^{54}	2^{56}
$Q = 2^{90}$	39	2^{20}	2^{25}	2^{27}

4.4 Experimental verification of the model's relevance

To verify he relevance of the mathematical model introduced in Section 4.2 the following experiments were carried out.

Let $n = 32$ be the dimension of key space V_n, let A linear feedback shift register which characteristic polynomial is equal to primitive $p(x) = x^{32} + x^7 + x^6 + x^2 + 1$. We consider four Boolean funtions of 32 variables described below as the filter functions.

Let we need to construct such generalised approximation that key recovery method based on it has a complexity $Q = 2^{24}$ and reliability $\pi_0 = 1/2$. As per relations of Section 4.3 we receive the following values:

$$k = 9, \delta = 27, N_1 = 3007, N_2 = 12861.$$

For each of the four generators we conduct an experiment in construction of one trajectory of N_1 length. This experiment was repeated many times (100 attempts) for different random start planes to calculate average value of N_2.

Let's describe used filter functions. Let $\pi_i : V_4 \to V_4$ be permutations defined in [9]. Let $\Psi : V_{32} \to V_{32}$ be mapping $\Psi(x) = \Psi(x_0||\dots||x_7) = \pi_0(x_0)||\dots||\pi_7(x_7)$, where $x = x_0||\dots||x_7 \in V_{32}$, $x_i \in V_4, i = 0, 1, \dots, 7$. Also let $\mathcal{S}(x) : V_{32} \to V_{32}$ be circular right shift by 11, let $\mathcal{X}(x) : V_{32} \to \mathbb{F}_2$ be sum modulo 2 of the vector x bits. The experiment was conducted for the following functions:

- $f_1(x) = \mathcal{X}(\Psi(\mathcal{S}(\Psi(\mathcal{S}(\Psi(x))))))$;
- $f_2(x) = \mathcal{X}(\Psi(x))$;
- $f_3(x) = \mathcal{X}(x_0||\dots||x_3||\pi_4(x_4)||\dots||\pi_7(x_7))$;
- $f_4(x) = \mathcal{X}(\pi_0(x_0)||\pi_1(x_1)||x_2||\dots||x_7)$.

As a result of experiments the following average values of N_2 were obtained for functions f_i. The average value of N_2 for f_1 is equal to 12910, for f_2 — 14107, f_3 — 14036, f_4 — 17215.

It's easy to see that the results obtained by introduced model are closest to the values obtained experimentally for the function f_1. Difference between predicted and real values of the parameter N_2 growing together with

increasing «structuring» of functions. Informally, this can be explained by the following fact. Proposed model assumes a relatively random distribution of the function values on V_n. The functions f_3 and f_4 depend on a lot of its variables linearly and function f_2 can be decomposed in the sum of 8 functions of 4 variables. The study and evaluation of this dependence is an open question and the subject of further research.

In general the experimental results confirm the relevance of the proposed model.

5 Conclusion

In this paper we consider the most simple exhaustive search method for constructing a generalised approximation of a filter function of some generator. The property of the key recovery method based on such approximation is that bits which the decision on key assignment to some plane is based on can be substantially separated by a generator's output sequence. This property produces the task of finding such values for the parameters of approximation that minimise the length of the output sequence needed key recovery method to achieve given complexity and reliability. This paper presents a solution to this problem. Modeling a plane belonging to an approximation by a random set we obtained the formula for calculating the optimal values of the planes dimension and the minimum of the number of prevailing function values on these planes.

This result allows to evaluate the limitations of the method, which can in some cases reduce the filter generator resistance to the threat of key recovery to the square root of the key space power.

References

[1] T. Siegenthaler. Decrypting a Class of Stream Cipher Using Ciphertext Only. IEEE Trans. on Computers, 1985, Vol. C-34(1), 81-85.

[2] M. Matsui. Linear cryptanalysis method for DES cipher, Advances in Cryptology – EUROCRYPT '93, Vol. 765 of Lecture Notes in Computer Science, pp. 386-397, Springer-Verlag, 1994.

[3] E. K. Alekseev. Attacking filter generators with combining functions close to algebraically degenerate functions. Collection of Articles by Young Researchers of the CMC Department of Moscow State University, 2011, issue 8, pages 114-123. (In Russian)

[4] V. Feller. An introduction to Probability Theory and its Applications, vol. 2, Wiley, New York, 1971.

[5] E. K. Alekseev, L. A. Kushchinskaya. Generalizing one method for recovering the key of a filter generator. Discrete Mathematics and Applications, 2017, forthcoming. (In Russian)

[6] O. A. Logachev, A. A. Salnikov, S. V. Smyshlyaev, V. V. Yashchenko. Boolean functions in coding theory and cryptology. URSS. ISBN 978-5-9710-0961-0, pp. 576, 2015. (In Russian)

[7] J. P. Mills. Table of the ratio: area to bounding ordinate, for any portion of normal curve. Biometrika 18, 1926, pp.395-400.

[8] G. Armengol, F. Utzet. Approximating Mills ratio. J. of Math. Anal. Appl. 420 (2014), pp. 1832-1853.

[9] GOST R 34.12-2015. Information technology. Information security. Block ciphers. Federal Agency on Technical Regulating and Metrology, 2015.

[10] E. Dawson, A. Clark, J. Golic, W. Millan, L. Penna, L. Simpson. The LILI-128 Keystream Generator. Proc. of first NESSIE workshop, Leuven, November 2000, http://www.cryponessie.org.

On upper bounds for periods of LCG sequences over Galois rings

Dmitry Ermilov

Abstract

Let $R = GR(q^n, p^n)$ be a Galois ring of cardinality q^n and characteristic p^n. The *linear congruential generator* (LCG) over R is a machine (see [3]) with the states sequence $\{x_i\}$ of elements defined by relation $x_{i+1} = ax_i + b$, where a, b and $x_0 \in R$. It is obvious that sequence $\{x_i\}$ is purely periodic with some least period $t \leq q^n$. In this paper we present an upper bound for the period of the LCG sequence. Some examples are given where the bound is achievable.

Keywords: Galois ring, LCG, sequence period.

1 Introduction

A very popular tool for pseudo-random sequence generation is provided by linear congruential generator. It is known (see [1]), when LCG sequence over $\mathbb{Z}_m$ achieves the largest period m. In paper [2] we proved that there is no full cycle polynomial transformation over Galois ring $GR(q^n, p^n)$, $q \neq p$ and $n > 1$. The aim of this paper is to prove an achievable upper bound for the period of LCG sequence over the Galois ring.

2 The upper bound of the cycle length in the graph $G_{ax+b,R}$

Let $R = GR(q^n, p^n)$ be a Galois ring of cardinality q^n and characteristic p^n, where $q = p^m$, $m > 1$, and let $G_{f,R}$ be the graph of bijective transformation of the ring R assigned by polynomial $f(x) \in R[x]$. By definition $J = pR$ and $R_i = R/J^i$, $i \in \overline{1, n}$.

Define the *polynomial congruential generator* (PCG) over the ring R as a machine with the states sequence $\{x_i\}$, $i = 0, 1, \ldots$. The elements x_i, $i = 0, 1, \ldots$ are defined by conditions:

$$x_{i+1} = f(x_i), \tag{1}$$

where $x_0 \in R$, and $f(x) \in R[x]$.

The maps

$$\phi_i : R \to R_i, i \in \overline{1, n}$$

are rings epimorphisms. There is the induced epimorphism of polynomials rings

$$\widehat{\phi}_i : R[x] \to R_i[x].$$

Let $t_n(C)$ be the length of the cycle C of the graph $G_{f,R}$. Denote by $t_s(C)$ the length of the cycle $\phi_s(C)$ of the graph G_{f_s,R_s}. Sometimes we will write t_s instead of $t_s(C)$ if the cycle C is known.

By definition, put

$$f^{[t]} = f \circ f \circ \cdots \circ f \ (t \text{ times}),$$

where $\circ$ – *composition* polynomials. We say that $f^{[t]}$ is t-*composition power* of polynomial f.

It follows from definition that $t_s(C)$ is a such minimum $T \in \mathbb{N}$ that $f^{[T]}(a) \equiv a \pmod{J^s}$ for all $a \in C$.

Fix the cycle C of the graph $G_{f,R}$. By definition, put $d_s = \frac{t_{s+1}}{t_s}$, $s = 1, 2, \ldots, n-1$, and let $F'(x)$ be standard *derivative* of the polynomial $F(x) = f^{[t_1]}(x)$ (see [4]), and $\alpha_C = F'(a) \in R$, $a \in C$. It was shown in [2], that value α_C depend only on the cycle C.

By $\bar{a}$ denote the image $\phi_1(a)$ of element a.

Let $\operatorname{ord} \bar{\alpha}_C$ be the order of the element $\bar{\alpha}_C$ from the multiplicative group $\bar{R}^*$ and

$$\delta_C = \begin{cases} p, \text{ if } \bar{\alpha}_C = \bar{e}, \\ \operatorname{ord} \bar{\alpha}_C \ - \ \text{otherwise}, \end{cases} \tag{2}$$

where e is the unit of the ring R.

We will need some results from the previous paper [2].

Theorem 1. [2] *If $f(x) \in R[x]$ is a bijective polynomial, $p > 2$ and $\delta_C = p$, then the sequence $d_1,\ d_2,\ \ldots,\ d_{n-1}$ has the form* a), b) *or* c):

a) $1, 1, \ldots, 1$;

b) $1, 1, \ldots, 1, p$;

c) $1, 1, \ldots, 1, p, \ldots, p$.

The first series of units may be missing.

We give an analogue of theorem 1 for $p = 2$.

Theorem 2. [2] *If $f(x) \in R[x]$ is a bijective polynomial, $p = 2$ and $\delta_C = p$, then the sequence $d_1,\ d_2,\ \ldots,\ d_{n-1}$ has the form* a), b), c) *or* d)*:*

a) $1, 1, \ldots, 1$;

b) $1, 1, \ldots, 2, \ldots, 2$;

c) $1, 1, \ldots, 2, 1, \ldots, 1$;

d) $1, 1, \ldots, 2, 1, \ldots, 1, 2, \ldots, 2$.

The first series of units may be missing. The cases c) *and* d) *are possible iff $\alpha_C \equiv 3e \pmod{J^2}$.*

The maps

$$\varphi_i : R_{s+1} \to R_s, s \in \overline{1, n-1}$$

are the rings epimorphisms.

Theorem 3. [2] *Let $f(x)$ be polynomial from $R[x]$, C be the cycle of the graph G_{f_s,R_s}, and t be length of the cycle C. Then the set of elements $\varphi_s^{-1}(C) \subset R_{s+1}$ generates the following cycles of the graph $G_{f_{s+1},R_{s+1}}$:*

a) *one cycle with length t and $\frac{q-1}{\delta_C}$ cycles with length $\delta_C t$;*

b) *$\frac{q}{p}$ cycles with length pt;*

c) *q cycles with length t,*

where δ_C defined in 2.

Consider a *linear congruential generator* (LCG) assigned by the polynomial $ax + b \in R[x]$.

We have got the upper bound of length cycle of the graph $G_{ax+b,R}$ in the follow statement.

Statement 4. *The length of the cycle of the graph $G_{ax+b,R}$, $ax+b \in R[x]$ is at most $(q-1)p^{n-1}$.*

Proof. Let C be the cycle of the graph $G_{ax+b,R}$, with a length greater than one. We claim that the parameter δ_C is equal to p. Indeed, since $f(x) = ax + b$, it follows that $k-$composition power of polynomial $f(x)$ satisfies the equation:

$$f^{[k]}(x) = a^k x + b(1 + \cdots + a^{k-1}).$$

We have

$$f^{[t_1]}(x) \equiv a^{t_1} x + b(1 + \cdots + a^{t_1-1}) \equiv$$

$$\equiv a^{t_1} x + b(\frac{a^{t_1} - e}{a - e}) \equiv x \,(\mathrm{mod}\, J), \qquad (3)$$

for any $x \in C$.

Hence $a^{t_1} \equiv e \,(\mathrm{mod}\, J)$, then using (3) we obtain that $\widehat{\phi}_1(f^{[t_1]}(x)) = x$.

Then $\alpha_C \equiv (f^{[t_1]}(x))' \equiv e \,(\mathrm{mod}\, J)$. From definition (2) value δ_C it follows that $\delta_C = p$.

From theorems 1 and 2 it follows that the length of the cycle C of the graph $G_{ax+b,R}$ is at most value $t_1 p^{n-1}$ when $p \neq 2$ and does not exceed value $t_1 2^{n-1}$ when $p = 2$. In both cases $t_1 \leq q - 1$, and this completes the proof. $\triangleright$

Suppose that $p \neq 2$.

Theorem 5. *There is a cycle with length $(q-1)p^{n-1}$ in the graph $G_{ax+b,R}$, $p \neq 2$ iff*

1. *$a \equiv a_0 \pmod J$, where a_0 is a primitive element of the field R_1;*
2. *$a \in R \setminus \Gamma(R)$, where $\Gamma(R) = \{r \in R |\, r^q = r\}$.*

Proof. We obviously have

$$f^{[k]}(x_0) \equiv a^k x_0 + b\frac{a^k - 1}{a - 1} \equiv 0 \pmod{J} \Leftrightarrow$$

$$\Leftrightarrow x_0 = \frac{b}{a-1} \text{ or } k = ord\,\bar{a}.$$

Using condition 1 we obtain $ord\,a = q - 1$ and the cyclic structure of the graph G_{ax+b,R_1} equals $[(q-1)^1, 1^1]$ and we have $t_1 = q - 1$ for the larger cycle.

Using condition 2 we get

$$f^{[q-1]}(x_0) \equiv a^{q-1}x_0 + b\frac{a^{q-1} - 1}{a - 1} \not\equiv x_0 \pmod{J^2} \Leftrightarrow$$

$$\Leftrightarrow a^{q-1} \not\equiv e \pmod{J^2} \Leftrightarrow a \in R \setminus \Gamma(R).\triangleright$$

Corollary 5.1. *Suppose there is a cycle of the length $(q-1)p^{n-1}$ in the graph $G_{ax+b,R}$ then there are $(\frac{q}{p})^{n-1}$ cycles with the length $(q-1)p^{n-1}$ in the graph $G_{ax+b,R}$.*

Proof. Combining theorems 1 and 3 we get

$$(d_1, d_2, \ldots, d_{n-1}) = (p, p, \ldots, p)$$

and the cyclic structure of the graph $G_{ax+b,R}$ contains $(\frac{q}{p})^{n-1}$ cycles with the length$(q-1)p^{n-1}$.$\triangleright$

Now suppose that $p = 2$. The follow theorem and corollary are proved similarly to theorem 5 and corollary 5.1.

Theorem 6. *There is a cycle with length $(2^m - 1)2^{n-1}$ in the graph $G_{ax+b,R}$, $p = 2$ iff*

1. *$a \equiv a_0 \pmod{J}$, where a_0 – primitive element of the field R_1;*
2. *$a \in R \setminus \Gamma(R)$;*
3. *$\alpha_C \not\equiv 3e \pmod{J^2}$ for a cycle C of the graph $G_{ax+b,R}$.*

Corollary 6.1. *Suppose there is a cycle of length* $(2^m-1)2^{n-1}$ *in the graph* $G_{ax+b,R}$ *then there are* $2^{(m-1)(n-1)}$ *cycles with the length* $(2^m-1)2^{n-1}$ *in the graph* $G_{ax+b,R}$.

The experiments showed, that the bound from theorem 4 is achievable.

Here is an example for the case $p \neq 2$.

Example 1. Consider the Galois ring $R = GR(5^{3*2}, 5^3) = \mathbb{Z}_{125}[y]_{/y^2+y+1}$. Denote by $(a_1, a_0) \in \mathbb{Z}_{125}^2$ the element $[a_1 y + a_0]_{y^2+y+1} \in R$

For example, the element

$$[2y+1]_{y^2+y+1} \in R$$

is denoted by $(2,1)$. Consider the polynomial $f(x) \in R[x]$:

$$f(x) = (6,2)x + (3,4).$$

The cyclic structure of the graph $G_{f,R}$ is equal to $[600^{25}, 120^5, 24^1, 1^1]$ and the bound from the theorem 4 is achieved.

Here is an example for the case $p = 2$.

Example 2. Consider the Galois ring $R = GR(2^{4*3}, 2^4) = \mathbb{Z}_{16}[y]_{/y^3+y+1}$. Denote by $(a_2, a_1, a_0) \in \mathbb{Z}_{16}^3$ the element $[a_2 y^2 + a_1 y + a_0]_{y^3+y+1} \in R$.

For example, the element

$$[y^2+2y+1]_{y^3+y+1} \in R$$

is denoted by $(1,2,1)$. Consider the polynomial $f(x) \in R[x]$:

$$f(x) = (4,3,1)x + (0,1,1).$$

The cyclic structure of the graph $G_{f,R}$ is equal to $[56^{16}, 28^8, 14^4, 7^1, 1^1]$ and the bound from the theorem 4 is achieved.

Now we show that the condition 3 of the theorem 6 is essential.

Example 3. Consider the Galois ring $R = GR(2^{3*3}, 2^3) = \mathbb{Z}_8[y]_{/y^3+y+1}$ from the example 2.

Consider the polynomial $f(x) \in R[x]$:

$$f(x) = (4,3,3)x + (1,1,2).$$

The cyclic structure of $G_{f,R}$ is equals $[14^{32}, 7^8, 1^1]$ and the bound from the theorem 6 is not achieved.

3 Conclusion

In this paper we get the upper bound for the length of cycle in the graph of affine transformation under Galois rings. The result of this paper is the first step to obtaining a cyclic structure of the graph of affine transformation under Galois rings. Studying a cyclic structure of graphs of quadratic and inverse transformations under Galois rings are very interesting and is an area ripe for further study.

References

[1] D.E. Knuth. Seminumerical Algorithms. The Art of Computer Programming // Addison-Wesley. Vol 2, 1969.

[2] D.M. Ermilov, O.A. Kozlitin. Cyclic structure of a polinomial genger-ator over the Galois ring // Mathmatical Aspects of Cryptography, 2013 vol. 4, no 1, pp. 27-57. (In Russian)

[3] A. Gill. Introduction to the theory of finite-state machines // New York, San Frsncisco, Toronto, London: McGraw-Hill Book Company, 1962, 218 p.

[4] S. Leng. Algebra. // Springer, 2002, 914 p.