ПРИКЛАДНАЯ ДИСКРЕТНАЯ МАТЕМАТИКА

2018 Математические методы криптографии № 40

UDC 519.7

MATHEMATICAL METHODS IN SOLUTIONS OF THE PROBLEMS PRESENTED AT THE THIRD INTERNATIONAL STUDENTS' OLYMPIAD IN CRYPTOGRAPHY[1]

N. Tokareva*,**, A. Gorodilova*,**, S. Agievich***, V. Idrisova*,**, N. Kolomeec*,**,
A. Kutsenko*, A. Oblaukhov*, G. Shushuev**

**Novosibirsk State University, Novosibirsk, Russia,*
***Sobolev Institute of Mathematics, Novosibirsk, Russia,*
****Belarusian State University, Minsk, Belarus*

The mathematical problems, presented at the Third International Students' Olympiad in Cryptography NSUCRYPTO'2016, and their solutions are considered. They are related to the construction of algebraic immune vectorial Boolean functions and big Fermat numbers, the secrete sharing schemes and pseudorandom binary sequences, biometric cryptosystems and the blockchain technology, etc. Two open problems in mathematical cryptography are also discussed and a solution for one of them proposed by a participant during the Olympiad is described. It was the first time in the Olympiad history. The problem is the following: construct $F : \mathbb{F}_2^5 \to \mathbb{F}_2^5$ with maximum possible component algebraic immunity 3 or prove that it does not exist. Alexey Udovenko from University of Luxembourg has found such a function.

Keywords: *cryptography, ciphers, Boolean functions, biometry, blockchain, Olympiad, NSUCRYPTO.*

DOI 10.17223/20710410/40/4

Introduction

The Third International Students' Olympiad in Cryptography — NSUCRYPTO'2016 was held on November 13–21, 2016. NSUCRYPTO is the unique cryptographic Olympiad containing scientific mathematical problems for students and professionals from any country. Its aim is to involve young researchers in solving curious and tough scientific problems of modern cryptography. From the very beginning, the concept of the Olympiad was not to focus on solving olympic tasks but on including unsolved research problems at the intersection of mathematics and cryptography.

The Olympiad consisted of two independent Internet rounds. The First round (duration 4 hours 30 minutes) was individual. It was divided into two sections: A and B. Theoretical problems in mathematics of cryptography were offered to participants. The Second round (duration 1 week) was devoted to research and programming problems of cryptography solved in teams. Anyone who wanted to try his/her hand in solving cryptographic problems were able to become a participant. During the registration every participant had to choose corresponding category: "School Student" (for junior researchers: pupils and school students), "University Student" (for participants who were currently studying at universities) or "Professional" (for participants who had already completed education, are PhD students, or just wanted to be in the restriction-free category). The winners

[1] The work was supported by Russian Ministry of Science and Education under the 5-100 Excellence Programme, RMC NSU, and by the Russian Foundation for Basic Research (projects no. 15-07-01328, 17-41-543364).

were awarded in each category separately. The language of the Olympiad is English. All information about organization and rules of the Olympiad can be found on the official website at `www.nsucrypto.nsu.ru`.

In 2016 the geography of participants has expanded significantly. There were 420 participants from 24 countries: Russia (Novosibirsk, Moscow, Saint Petersburg, Yekaterinburg, Kazan, Saratov, Taganrog, Krasnoyarsk, Petrozavodsk, Perm, Chelyabinsk, Zelenograd, Tomsk, Korolev, Omsk, Ramenskoye, Yaroslavl, Novokuznetsk), Belarus (Minsk), Ukraine (Kiev, Kharkov, Zaporozhye), Kazakhstan (Astana, Almaty, Kaskelen), Kirghizia (Bishkek), Great Britain (Bristol), Bulgaria (Sofia), Germany (Berlin, Munich, Bochum, Witten), France (Paris), Luxembourg (Luxembourg), Hungary (Szeged), Sweden (Gothenburg), Switzerland (Zurich, Bern), Italy (Padova), Czech Republic (Prague), Estonia (Tartu), Spain (Barcelona) Canada (Edmonton), Iran (Tehran), South Africa (Cape Town), China (Beijing), Vietnam (Ho Chi Minh City, Saigon), Indonesia (Bandung), India (Kollam, Haydebarad).

Organizers of the Olympiad are: Novosibirsk State University, Sobolev Institute of Mathematics (Novosibirsk), Tomsk State University, Belarusian State University and University of Leuven (KU Leuven, Belgium).

In section 1 we describe problem structure of the Olympiad according to sections and rounds. Section 2 is devoted to unsolved problems formulated in NSUCRYPTO for all years since 2014, with attention to solutions proposed for two of them. Section 3 contains the conditions of all 16 mathematical problems of NSUCRYPTO'2016. Among them, there are both some amusing tasks based on historical ciphers as well as hard mathematical problems. We consider mathematical problems related to the construction of algebraically immune vectorial Boolean functions and big Fermat numbers, problems about secrete sharing schemes and pseudorandom binary sequences, biometric cryptosystems and the blockchain technology, etc. Some unsolved problems are also discussed. In section 4 we present solutions of all the problems with paying attention to solutions proposed by the participants. Information about the winners is given in section 5.

Mathematical problems of the previous International Olympiads NSUCRYPTO'2015 and NSUCRYPTO'2016 can be found in [1] and [2] respectively.

1. Problem structure of the Olympiad

There were 16 problems on the Olympiad. Some of them were included in both rounds (Tables 1, 2). Thus, section A (school section) of the first round consisted of 6 problems, whereas section B (student section) contained 7 problems. Three problems were common for both sections. In the Table 1 you can see the highest scores one could get in case of solving the problem.

The second round was composed of 12 problems; they were common for all the participants. Two of the problems presented on the second round were marked as unsolved (awarded special prizes from the Program Committee).

Table 1

Problems of the first round

A — school section

N	Problem title	Maximum scores
1	Cipher from the pieces	4
2	Get an access	4
3	Find the key	4
4	Labyrinth	4
5	System of equations	4
6	Biometric pin-code	4

B — student section

N	Problem title	Maximum scores
1	Cipher from the pieces	4
2	Labyrinth	4
3	Quadratic functions	8
4	System of equations	4
5	Biometric key	6
6	Secret sharing	6
7	Protocol	6

Table 2

Problems of the second round

N	Problem title	Maximum scores
1	Algebraic immunity	Unsolved
2	Zerosum at AES	8
3	Latin square	6
4	Nsucoin	10
5	Metrical cryptosystem	6
6	Quadratic functions	8
7	Secret sharing	6
8	Biometric key	6
9	Protocol	6
10	Find the key	4
11	Labyrinth	4
12	Big Fermat numbers	Unsolved

2. Unsolved problems of NSUCRYPTO since the origin

In this section we shortly present all unsolved problems stated in the history of NSUCRYPTO (Table 3), where we mention also the current status of all the problems. The formulations of the problems can be found in [1] (year 2014), [2] (year 2015) as well as the current paper (year 2016).

Table 3

Unsolved Problems of NSUCRYPTO

N	Year	Problem title	Status
1	2014	Watermarking cipher	Unsolved
2	2014	APN permutation	Unsolved
3	2014	Super S-box	Unsolved
4	2015	A secret sharing	Partially SOLVED in [3]
5	2015	Hypothesis	Unsolved
6	2016	Algebraic immunity	SOLVED during the Olympiad
7	2016	Big Fermat numbers	Unsolved

The Olympiad NSUCRYPTO-2016 became particular since there was the first time when an unsolved problem stated was successfully solved by a participant during the Olympiad. Alexey Udovenko (University of Luxembourg) was able to find a solution to the problem “Algebraic immunity” (see section 4.15).

Moreover, we are also very pleased to say that the unsolved problem “A secret sharing” of NSUCRYPTO-2015 was also partially solved! In [3] Kristina Geut, Konstantin

Kirienko, Prokhor Kirienko, Roman Taskin, Sergey Titov (Ural State University of Railway Transport, Yekaterinburg) found a solution for the problem in the case of even dimension (the problem remains open for odd dimension).

The current status of unsolved problems from NSUCRYPTO of all years can be found at `http://nsucrypto.nsu.ru/unsolved-problems/`. Everyone is welcome to propose a solution to any problem stated. Please, send you ideas to `nsucrypto@nsu.ru`.

3. Problems

In this section we formulate all the problems of the Olympiad.

3.1. Problem "Cipher from the pieces"

Recover the original message, splitting the figure (Fig. 1) into equal pieces such that each color occurs once in every piece.

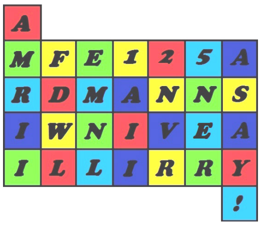


Fig. 1.

3.2. Problem "Get an access"

To get an access to the safe one should put 20 non-negative integers in the following cells (Fig. 2). The safe will be opened if and only if the sum of any two numbers is even number k, such that $4 \leqslant k \leqslant 8$, and each possible sum occurs at least once. Find the sum of all these numbers.

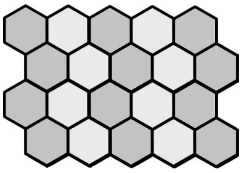


Fig. 2.

3.3. Problem "Find the key"

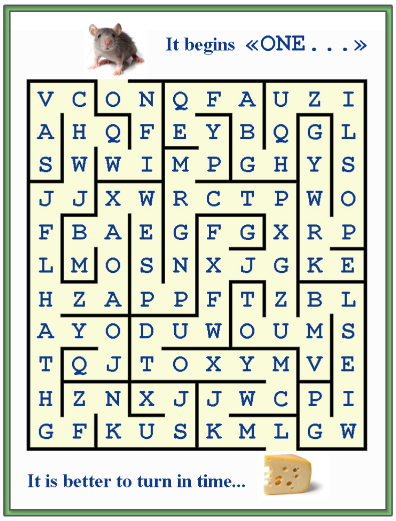
The key of a cipher is the set of positive integers a, b, c, d, e, f, g, such that the following relation holds:

$$a^3 + b^3 + c^3 + d^3 + e^3 + f^3 + g^3 = 2016^{2017}.$$

Find the key!

3.4. Problem "Labyrinth"

Read the message hidden in the labyrinth (Fig. 3)!



V	C	O	N	Q	F	A	U	Z	I
A	H	Q	F	E	Y	B	Q	G	L
S	W	W	I	M	P	G	H	Y	S
J	J	X	W	R	C	T	P	W	O
F	B	A	E	G	F	G	X	R	P
L	M	O	S	N	X	J	G	K	E
H	Z	A	P	P	F	T	Z	B	L
A	Y	O	D	U	W	O	U	M	S
T	Q	J	T	O	X	Y	M	V	E
H	Z	N	X	J	J	W	C	P	I
G	F	K	U	S	K	M	L	G	W

Fig. 3.

3.5. Problem "System of equations"

Analyzing a cipher Caroline gets the following system of equations in binary variables $x_1, x_2, \ldots, x_{16} \in \{0,1\}$ that represent the unknown bits of the secrete key:

$$\begin{cases} x_1x_3 \oplus x_2x_4 = x_5 - x_6, \\ x_{14} \oplus x_{11} = x_{12} \oplus x_{13} \oplus x_{14} \oplus x_{15} \oplus x_{16}, \\ (x_8 + x_9 + x_7)^2 = 2(x_6 + x_{11} + x_{10}), \\ x_{13}x_{11} \oplus x_{12}x_{14} = -(x_{16} - x_{15}), \\ x_5x_1x_6 = x_4x_2x_3, \\ x_{11} \oplus x_8 \oplus x_7 = x_{10} \oplus x_6, \\ x_6x_{11}x_{10} \oplus x_7x_9x_8 = 0, \\ \left(\dfrac{x_{12} + x_{14} + x_{13}}{\sqrt{2}}\right)^2 - x_{15} = x_{16} + x_{11}, \\ x_1 \oplus x_6 = x_5 \oplus x_3 \oplus x_2, \\ x_6x_8 \oplus x_9x_7 = x_{10} - x_{11}, \\ 2(x_5 + x_1 + x_6) = (x_4 + x_3 + x_2)^2, \\ x_{11}x_{13}x_{12} = x_{15}x_{14}x_{16}. \end{cases}$$

Help Caroline to find the all possible keys!

Remark. If you do it in analytic way (without computer calculations) you get twice more scores.

3.6. Problem "Biometric pin-code"

Iris is one of the most reliable biometric characteristics of a human. While measuring let us take 16-bit vector from the biometric image of an iris. As in reality, we suppose that two 16-bit biometric images of *the same human* can differ not more than by 10–20 %, while biometric images of *different people* have differences at least 40–60 %.

Let a key k be an arbitrary 5-bit vector. We suppose that the key is a pin-code that should be used in order to get an access to the bank account of a client.

To avoid situation when malefactor can steal the key of a some client and then be able to get an access to his account, the bank decided to combine usage of the key with biometric authentication of a client by iris-code. The following scheme of covering the key with biometric data was proposed:

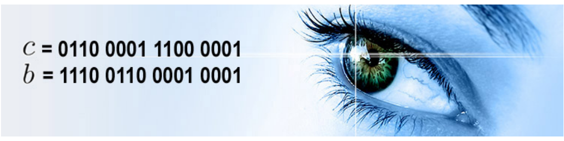
1) on registration of a client take 16-bit biometric image b_{template} of his iris;
2) extend 5-bit key k to 16-bit string s using Hadamard encoding, i.e. if $k = (k_1, \ldots, k_5)$, where $k_i \in \{0, 1\}$, then s is the vector of values of the Boolean function $f(x_1, \ldots, x_4) = k_1x_1 \oplus \ldots \oplus k_4x_4 \oplus k_5$;
3) save the vector $c = b_{\text{template}} \oplus s$ on the smart-card and give it to the client. A vector c is called *biometrically encrypted key.*

To get an access to his account a client should

1) take a new 16-bit biometric image b of his iris;
2) using information from the smart-card count 16-bit vector s' as $s' = b \oplus c$;
3) decode s' to the 5-bit vector k' using Hadamard decoding procedure.

Then the bank system checks: if $k' = k$ then the client is authenticated and the key is correct; hence bank provides an access to the account of this client. Otherwise, if $k' \neq k$ then bank signals about an attempt to get illegal access to the bank account.

The problem. Find the 5-bit k of Alice if you know her smart-card data c and a new biometric image b (both are given on the picture (Fig. 4)).



c = 0110 0001 1100 0001
b = 1110 0110 0001 0001

Fig. 4.

Remark. Vector of values of a Boolean function f in 4 variables is a binary vector $(f(x^0), f(x^1), \ldots, f(x^{15}))$ of length 16, where $x^0 = (0, 0, 0, 0)$, $x^1 = (0, 0, 0, 1)$, …, $x^{15} = (1, 1, 1, 1)$, ordered by lexicographical order; for example, vector of values of the function $f(x_1, x_2, x_3, x_4) = x_3 \oplus x_4 \oplus 1$ is equal to (1010101010101010).

3.7. Problem "Quadratic functions"

Alice and Bob are going to use the following pseudorandom binary sequence $u = \{u_i\}$, $u_i \in \mathbb{F}_2$:

— $u_1, \ldots, u_n$ are initial values;
— $u_{i+n} = f(u_i, u_{i+1}, \ldots, u_{i+n-1})$, where

$$f \in Q_n = \{a_0 \oplus \bigoplus_{i=1}^{n} a_ix_i \oplus \bigoplus_{1 \leqslant i < j \leqslant n} a_{ij}x_ix_j : a_0, a_i, a_{ij} \in \mathbb{F}_2\}.$$

Suppose that you have intercepted the elements $u_t, u_{t+1}, \ldots, u_{t+k-1}$ of a sequence for some t. Is it possible to uniquely reconstruct the elements $u_{t+k}, u_{t+k+1}, u_{t+k+2}, \ldots$ provided $k \leqslant cn$, where c is a constant independent on n?

3.8. Problem "Biometric key"

Iris is one of the most reliable biometric characteristics of a human. While measuring let us take 128-bit biometric image of an iris. As in reality, we suppose that two 128-bit biometric images of *the same human* can differ not more than by 10–20 %, while biometric images of *different people* have differences at least 40–60 %.

Let a key k be an arbitrary 8-bit vector. It can be represented in hexadecimal notation. For example, $\texttt{e2} = 11100010$. We suppose that the key is a pin-code that should be used in order to get access to the bank account of a client.

To avoid situation when malefactor can steal the key of a some client and then be able to get an access to his account, the bank decided to combine usage of the key with biometric authentication of a client by iris-code. The following scheme of covering the key with biometric data was proposed:

1) on registration of a client take 128-bit biometric image b_{template} of his iris;
2) extend 8-bit key k to 128-bit string s using Hadamard encoding, i.e. if $k = (k_1, \ldots, k_8)$, where $k_i \in \mathbb{F}_2$, then s is the vector of values of the Boolean function $f(x_1, \ldots, x_7) = k_1x_1 \oplus \ldots \oplus k_7x_7 \oplus k_8$;
3) save the vector $c = b_{\text{template}} \oplus s$ on the smart-card and give it to the client. A vector c is called *biometrically encrypted key.*

To get an access to his account a client should

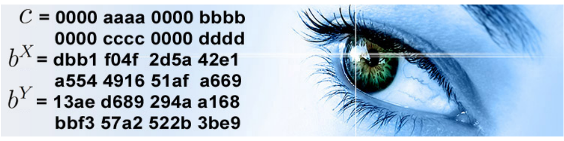
1) take a new 128-bit biometric image b of his iris;
2) using information from the smart-card count 128-bit vector s' as $s' = b \oplus c$;
3) decode s' to 8-bit vector k' using Hadamard decoding procedure.

Then the bank system checks: if $k' = k$ then the client is authenticated and the key is correct; hence bank provides an access to the account of this client. Otherwise, if $k' \neq k$ then bank signals about an attempt to get illegal access to the bank account.

The problem. One day a person, say X, came to the bank and tried to get an access to the bank account of Alice using the smart-card. This may be noticed that person X was in hurry and may be a little bit nervous. Suddenly, another person, say Y, appeared in the bank and declared loudly: “Please stop any operation! I am Alice! My smart-card was stolen.”

Bank clerk, say Claude, stopped all operations. In order to solve the situation he took new biometric images b^X and b^Y of persons X and Y respectively, and with smart-card containing vector c leaved his post for consultations with bank specialists.

When Claude came back, he already knew who was Alice. He wanted to stop the other person and call to police but that person has already disappeared. So, can you solve this problem too? Who was real Alice? Determine her 8-bit key k. You can use the data b^X, b^Y and c presented on the picture (Fig. 5). It is known also that the key of Alice contains odd number of ones.



C = 0000 aaaa 0000 bbbb
0000 cccc 0000 dddd
b^X = dbb1 f04f 2d5a 42e1
a554 4916 51af a669
b^Y = 13ae d689 294a a168
bbf3 57a2 522b 3be9

Fig. 5.

3.9. Problem “Secret sharing”

Alena, Boris and Sergey developed the following secret sharing scheme to share a password $P \in \mathbb{F}_2^{32}$ into three parts to collectively manage money through online banking.

— Vectors $v_i^a, v_i^b, v_i^s \in \mathbb{F}_2^{32}$ and values $c_i^a, c_i^b, c_i^s \in \mathbb{F}_2$ are randomly generated for all $i = 1, \ldots, 32$.
— Vectors v_i^a, v_i^b, v_i^s are known to all participants of the scheme.
— Values $c_i^a, c_i^b, c_i^s \in \mathbb{F}_2$ are known only to Alena, Boris and Sergey respectively.

— Then the secret password P is calculated by the rule

$$P = \bigoplus_{i=1}^{32} c_i^a v_i^a \oplus \bigoplus_{i=1}^{32} c_i^b v_i^b \oplus \bigoplus_{i=1}^{32} c_i^s v_i^s.$$

What is the probability that Alena and Boris together can not get any information about the password P? What is the probability that they are able without Sergey to get a guaranteed access to online banking using not more than 23 attempts?

3.10. Problem "Protocol"

Alena and Boris developed a new protocol for establishing session keys. It consists of the following three steps:

1) The system has a common prime modulus p and a generator g. Alena and Boris have their own private keys $\alpha_a \in \mathbb{Z}_{p-1}$, $\alpha_b \in \mathbb{Z}_{p-1}$ and corresponding public keys $P_a = g^{\alpha_a} \mod p$, $P_b = g^{\alpha_b} \mod p$.
2) To establish a session key Alena generates a random number $R_a \in \mathbb{Z}_{p-1}$, computes $X_a = (\alpha_a + R_a) \mod (p-1)$ and sends it to Boris. Then Boris generates his random number R_b, computes X_b in the same way as Alena and sends it back to her.
3) Alena computes the session key in the following way:

$$K_{a,b} = (g^{X_b} P_b^{-1})^{R_a} \mod p.$$

Bob computes the session key in the following way:

$$K_{b,a} = (g^{X_a} P_a^{-1})^{R_b} \mod p.$$

How can an attacker Evgeniy compute any future session key between Alena and Boris, if he steals the only one session key $K_{a,b}$?

3.11. Problem "Zerosum at AES"

Let $\texttt{AES}_0$ be a mapping that represents the algorithm AES-256 with the all-zero key. Let $X_1, \ldots, X_{128} \in \mathbb{F}_2^{128}$ be pairwise different vectors such that

$$\bigoplus_{i=1}^{128} X_i = \bigoplus_{i=1}^{128} \texttt{AES}_0(X_i).$$

1) Propose an effective algorithm to find an example of such vectors $X_1, \ldots, X_{128}$.
2) Provide an example of $X_1, \ldots, X_{128}$.

3.12. Problem "Latin square"

Alice has registered on Bob's server. During the registration Alice got the secret key that is represented as a latin square of order 10. A latin square is a 10×10 matrix filled with integers $0, 1, \ldots, 9$, each occurring exactly once in each row and exactly once in each column.

To get an access to Bob's resources Alice authenticates by the following algorithm:

1) Bob sends to Alice a decimal number $abcd$, where $a, b, c, d \in \{0, 1, \ldots, 9\}$ and $a \neq b$, $b \neq c$, $c \neq d$.
2) Alice performs three actions.
   — At first she finds the integer t_1 standing at the intersection of the row $(a+1)$ and the column $(b+1)$.
   — Then she finds t_2 standing at the intersection of the row $(t_1 + 1)$ and the column $(c+1)$.

   — Finally, Alice finds t_3 standing at the intersection of the row $(t_2 + 1)$ and the column $(d + 1)$.

3) Alice sends to Bob the integer t_3.
4) Bob performs the same actions and verifies Alice's answer.
5) Steps 1–4 are repeated several times. In case of success Bob recognizes that Alice knows the secret latin square.

Find Alice's secret key if you can get the answer t_3 for any your correct input request *abcd* here: `http://nsucrypto.nsu.ru/archive/2016r2/task3`.

3.13. Problem "*nsucoin*"

Alice, Bob, Caroline and Daniel are using a digital payment system ***nsucoin*** (Fig. 6) to buy from each other different sorts of flowers. Alice sells only chamomiles, Bob — only tulips, Caroline — only gerberas and Daniel — only roses. At the beginning each person has 5 flowers. The cost of each flower is 2 coins.

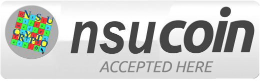


Fig. 6.

Transactions are used to make purchases by transferring coins in the system *nsucoin*. Each transaction involves two different users (the seller A and the buyer B) and distributes a certain amount of coins S between A and B, say $S = S_A + S_B$. The value S is equal to the sum of all the coins received by the buyer in the indicated k transactions, $1 \leqslant k \leqslant 2$. We will say that the current transaction is *based* on these k transactions. The value S_A is the amount of coins that the buyer pays the seller for his product, $S_A > 0$; the value S_B is the rest of available amount of coins S that returns to buyer (in further transactions B can spend these coins). At the same time, coins received by users in each transaction can not be distributed more than once in other transactions.

In order for transactions to be valid they must be verified. To do this **block chain** is used. Each block verifies from 1 to 4 transactions. Each transaction to be verified can be based on already verified transactions and transactions based on verified transactions.

There are 4 *special* transactions. Each of them brings 10 coins to one user. These transactions do not based on other transactions. The first block verifies all special transactions.

Define what bouquet Alice can make from the flowers she has if the last block in chain is the following string (hash of this block in 00004558):

```
height:2;prevHash:0000593b;ctxHash:8fef76cb;nonce:17052
```

Technical description of *nsucoin*

• **Transactions.** Transaction is given by the string `transaction` of the following format:

```
transaction = “txHash:{hashValue};{transactionInfo}”
hashValue = Hash({transactionInfo})
transactionInfo = “inputTx:{Tx};{sellerInfo};{buyerInfo}”
Tx = “{Tx1}” or “{Tx1,Tx2}”
sellerInfo = “value1:{V1};pubKey1:{PK1};sign1:{S1}”
buyerInfo = “value2:{V2};pubKey2:{PK2};sign2:{S2}”
```

Here `Tx1, Tx2` are values of the field `txHash` of transactions which the current transaction based on; `V`i is a non-negative integer that is equal to the amount of coins received by the user with public key `PK`i, $0 \leqslant$ `V`$i \leqslant 10$, `V1`$\neq 0$. Digital signature

$$\texttt{S}i = \texttt{DecToHexStr}(\texttt{Signature}(\texttt{Key2,StrToByteDec}(\texttt{Hash}(\texttt{Tx1+Tx2+PK}i)))),$$

where `+` is concatenation operation of strings. `Key2` is private key of buyer.

In the special transactions fields `inputTx`, `sign1` are empty and there is no `buyerInfo`. For example, one of the special transactions is the following:

```
txHash:1a497b59;inputTx:;value1:10;pubKey1:11;sign1:
```

- **Block chain.** Each block is given by the string `block` of the following format:

```
block = "height:{Height};prevHash:{PrHash};ctxHash:{CTxHash};nonce:{Nonce}"
```

Here `Height` is the block number in a chain, the first block has number 0. `PrHash` is hash of block with number `Height` -1. `CTxHash` is hash of concatenation of all the `TxHash` of transactions verified by this block. `Nonce` is the minimal number from 0 to 40000 such that block has hash of the form `0000####`.

Let `PrHash` = `00000000` for the first block.

- **Hash function.** `Hash` is calculated as reduced MD5: the result of hashing is the first 4 bytes of standard MD5 represented as a string. For example, `Hash("teststring")`= = "`d67c5cbf`", `Hash("1a497b5917")` = "`e0b9e4a8`".

- **Digital signature.** `Signature`$(key, message)$ is RSA digital signature with n of order 64 bits, $n = 9101050456842973679$. Public exponents `PK` of users are the following:

User	Alice	Bob	Caroline	Daniel
PK	11	17	199	5

For example,
`Signature`(2482104668331363539, 7291435795363422520) = 7538508415239841520.

- **Additional functions.** `StrToByteDec` decodes a string to bytes that are considered as a number. Given a number `DecToHexStr` returns a string that is equal to the hexadecimal representation of this number. For example, `StrToByteDec("e0b9e4a8")` = = 7291435795363422520 and `DecToHexStr`(7538508415239841520) = "`689e297682a9e6f0`". Strings are given in UTF-8.

Examples of a transaction and a block

- Suppose that Alice are buying from Bob 2 tulips. So, she must pay him 4 coins. The transaction of this operation, provided that Alice gets 10 coin in the transaction with hash `1a497b59`, is

```
txHash:98e93fd5;inputTx:1a497b59;value1:4;pubKey1:17;sign1:689e297682a9e6f0;
                    value2:6;pubKey2:11;sign2:fec9245898b829c
```

- The block on height 2 verifies transactions with hash values (values of `txHash`) `98e93fd5`, `c16d8b22`, `b782c145` and `e1e2c554`, provided that hash of the block on height 1 is `00003cc3`, is the following:

```
height:2;prevHash:00003cc3;ctxHash:9f8333d4;nonce:25181
```

Hash of this block is `0000642a`.

3.14. Problem "Metrical cryptosystem"

Alice and Bob exchange messages using the following cryptosystem. Let $\mathbb{F}_2^n$ be an n-dimensional vector space over the field $\mathbb{F}_2 = \{0, 1\}$. Alice has a set $A \subseteq \mathbb{F}_2^n$ and Bob has a set $B \subseteq \mathbb{F}_2^n$ such that both A and B are metrically regular sets and they are metrical complements of each other. Let d be the Hamming distance between A and B. To send some number a $(0 \leqslant a \leqslant d)$ Alice chooses some vector $x \in \mathbb{F}_2^n$ at distance a from the set A and sends this vector to Bob. To obtain the number that Alice has sent Bob calculates the distance b from x to the set B and concludes that the initial number a is equal to $d - b$.

Is this cryptosystem correct? In other words, does Bob correctly decrypt all sent messages, regardless of initial sets A, B satisfying given conditions and of the choice of vector x?

Remark 1. Recall several definitions and notions. The *Hamming distance* $\mathrm{d}(x, y)$ between vectors x and y is the number of coordinates in which these vectors differ. Distance from vector $y \in \mathbb{F}_2^n$ to the set $X \subseteq \mathbb{F}_2^n$ is defined as $\mathrm{d}(y, X) = \min\limits_{x \in X} \mathrm{d}(y, x)$. The *metrical complement* of a set $X \subseteq \mathbb{F}_2^n$ (denoted by $\widehat{X}$) is the set of all vectors $y \in \mathbb{F}_2^n$ at maximum possible distance from X (this maximum distance is also known as *covering radius* of a set). A set $X \subseteq \mathbb{F}_2^n$ is called *metrically regular*, if its second metrical complement $\widehat{\widehat{X}}$ coincides with X.

Remark 2. Let us consider several examples:

— Let X consist of a single vector $x \in \mathbb{F}_2^n$. It is easy to see that $\widehat{X} = \{x \oplus \mathbf{1}\}$, where $\mathbf{1}$ is the all-ones vector, and therefore $\widehat{\widehat{X}} = \{x \oplus \mathbf{1} \oplus \mathbf{1}\} = \{x\} = X$, so X is a metrically regular set; it is also easy to see that cryptosystem based on $A = \{x\}$, $B = \{x \oplus \mathbf{1}\}$ is correct.

— Let Y be a ball of radius $r > 0$ centered at x: $Y = B(r, x) = \{y \in \mathbb{F}_2^n : \mathrm{d}(x, y) \leqslant r\}$. You can verify that $\widehat{Y} = \{x \oplus \mathbf{1}\}$, but $\widehat{\widehat{Y}} = \{x\} \neq Y$, and Y is not metrically regular.

— Let X be an arbitrary subset of $\mathbb{F}_2^n$. Then, if we denote $X_0 := X$, $X_{k+1} = \widehat{X_k}$ for $k \geqslant 0$, there exists a number M such that X_m is a metrically regular set for all $m > M$. You can prove this fact as a small exercise, or simply use it in your solution.

3.15. Problem "Algebraic immunity" (Unsolved)

A mapping F from $\mathbb{F}_2^n$ to $\mathbb{F}_2^m$ is called a *vectorial Boolean function.* If $m = 1$ then F is a *Boolean function* in n variables. A *component function* F_v of F is a Boolean function defined by a vector $v \in \mathbb{F}_2^m$ as follows $F_v = \langle v, F \rangle = v_1 f_1 \oplus \ldots \oplus v_m f_m$, where $f_1, \ldots, f_m$ are coordinate functions of F. A function F has its unique *algebraic normal form* (ANF)

$$F(x) = \bigoplus_{I \in \mathcal{P}(N)} a_I \prod_{i \in I} x_i,$$

where $\mathcal{P}(N)$ is the power set of $N = \{1, \ldots, n\}$ and a_I belongs to $\mathbb{F}_2^m$. Here $\oplus$ denotes the coordinate-wise sum of vectors modulo 2. The *algebraic degree* of F is the degree of its ANF: $\deg(F) = \max\{|I| : \ a_I \neq (0, \ldots, 0), \ I \in \mathcal{P}(N)\}$.

Algebraic immunity $\mathrm{AI}(f)$ of a Boolean function f is the minimal algebraic degree of a Boolean function g, $g \not\equiv 0$, such that $fg \equiv 0$ or $(f \oplus 1)g \equiv 0$. The notion was introduced by W. Meier, E. Pasalic, C. Carlet in 2004.

It is wellknown that $\mathrm{AI}(f) \leqslant \lceil n/2 \rceil$, where $\lceil x \rceil$ is the ceiling function of number x. There exist functions with $AI(f) = \lceil n/2 \rceil$ for any n.

Component algebraic immunity $\mathrm{AI}_{\mathrm{comp}}(F)$ of a function from $\mathbb{F}_2^n$ to $\mathbb{F}_2^m$ is defined as the minimal algebraic immunity of its component functions F_v, $v \neq (0, \ldots, 0)$. Component algebraic immunity was considered by C. Carlet in 2009. It is easy to see that $\mathrm{AI}_{\mathrm{comp}}(F)$ is also upper bounded by $\lceil n/2 \rceil$.

The problem. What is the tight upper bound of component algebraic immunity? For all possible combination of n and m, $m \leqslant n \leqslant 4$, vectorial Boolean functions with $\mathrm{AI}_{\mathrm{comp}}(F) = \lceil n/2 \rceil$ exist.

Construct $F : \mathbb{F}_2^5 \to \mathbb{F}_2^5$ with maximum possible component algebraic immunity 3 or prove that it does not exist.

3.16. Problem "Big Fermat numbers" (Unsolved)

It is known that constructing big prime numbers is very actual and complicated problem interesting for cryptographic applications. One of the popular way to find them is... to guess! For example to guess them between numbers of some special form. For checking there are Mersenne numbers $2^k - 1$, Fermat numbers $F_k = 2^{2^k} + 1$ for nonnegative integer k, etc.

Let us concentrate our attention on Fermat's numbers.

It is known that Fermat numbers $F_0 = 3$, $F_1 = 5$, $F_2 = 17$, $F_3 = 257$, $F_4 = 65537$ are prime. But the number $F_5 = 4\,284\,967\,297 = 641 \cdot 6\,700\,417$ is already composite as was proven by L. Euler in XVIII.

For now it is known that all Fermat numbers, where $k = 5, \ldots, 32$, are composite and there is the hypothesis that every Fermat number F_k, where $k \geqslant 5$ is composite.

Could you prove that for any big number N there exists a composite Fermat number F_k such that $F_k > N$?

4. Solutions of the problems

In this section we present solutions of the problems with paying attention to solutions proposed by the participants (right/wrong and beautiful).

4.1. Problem "Cipher from the pieces"

Solution. The only way to split this figure is the following (Fig. 7).

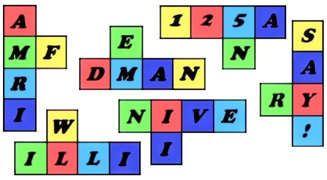


Fig. 7.

Then we need to arrange these pieces and read letters horizontally. In this way we can obtain the text «`WILLI AMFRI EDMAN 125AN NIVER SARY!`». So, the answer is «`WILLIAM FRIEDMAN 125 ANNIVERSARY!`».

The problem was devoted to the anniversary of a mathematician and cryptographer William Friedman known as «The Father of American Cryptology» (September 24, 1891 — November 12, 1969).

This problem was solved completely by 68 participants from all categories. They all acted in the same way. We could mentioned the best solutions of school students Alexander Grebennikov and Alexander Dorokhin (Presidential Physics and Mathematics Lyceum 239, Saint Petersburg), Arina Prostakova (Gymnasium 94, Yekaterinburg).

4.2. Problem "Get an access"

Solution. Here we want to describe a very compact and full solution by Vladimir Schavelev (Presidential Physics and Mathematics Lyceum 239, Saint Petersburg).

Let us prove that the sum can be any even number from 44 up to 76. It is obvious that the whole sum is even number since sums of any two numbers is even number (4, 6 or 8). Let us consider the set where the sum takes its minimal possible value. Due to condition of the problem we have two numbers a and b with the sum equal to 8, so split all the numbers in pairs, such that one of these pairs is a, b. So, the minimal sum of such pairs is $8 + 9 \cdot 4 = 44$. In the same way we obtain that the maximal possible sum is $4 + 9 \cdot 8 = 76$.

If we fill cells with two 2 and the rest of the numbers are 4, we obtain the sum $4 \cdot 18 + 4 = = 76$, and this filling satisfies the condition. If we substitute any "4" by "2" we obtain the sum 74. We can continue this process, obtaining all possible variants of the sum, up to minimal, that is 44.

This problem was solved by 9 school students in the first round.

4.3. Problem "Find the key"

Solution. The answer for this problem is any set of positive integers a, b, c, d, e, f, g such that the following relation holds:

$$a^3 + b^3 + c^3 + d^3 + e^3 + f^3 + g^3 = 2016^{2017}.$$

For example, such a set can be found in the following way:

$$a^3 + b^3 + c^3 + d^3 + e^3 + f^3 + g^3 = 2016^{2017} = 2016 \cdot 2016^{2016} = 2016 \cdot (2016^{672})^3.$$

Let us divide both sides on 2016^{2016} and assume that there exist a', b', c', d', e', f', g' such that $a'^3 + b'^3 + c'^3 + d'^3 + e'^3 + f'^3 + g'^3 = 2016$. Then we can find easily these numbers, for instance, one of such sets is 3, 4, 5, 6, 7, 8, 9.

Then the original solution has the form $x = x' \cdot 2016^{672}$. So, we have

$$(3 \cdot 2016^{672})^3 + (4 \cdot 2016^{672})^3 + (5 \cdot 2016^{672})^3 + (6 \cdot 2016^{672})^3 + (7 \cdot 2016^{672})^3 + \\ +(8 \cdot 2016^{672})^3 + (9 \cdot 2016^{672})^3 = 2016^{2017}.$$

The most elegant and beautiful solution was send by Alexey Udovenko (University of Luxembourg). There also were great solutions from Alexandr Grebennikov (Presidential Physics and Mathematics Lyceum 239, Saint Petersburg), Vadzim Marchuk, Anna Gusakova, and Yuliya Yarashenia team (Research Institute for Applied Problems of Mathematics and Informatics, Institut of Mathematics, Belarusian State University), that contains a lot of such keys and even the solutions by Alexey Ripinen, Oleg Smirnov, and Peter Razumovsky team (Saratov State University), Henning Seidler and Katja Stumpp team (Technical University of Berlin) that describes all possible keys. These solutions were awarded by additional scores.

4.4. Problem "Labyrinth"

Solution. Since there is a labyrinth, one can conjecture that the ciphertext is hidden in the right path from a mouse to cheese. Such a path is unique and contains the following string: `ONFIWQHWJJFLHZAOAXWESPPNGRCTPXGJXFWUDTOXYMCWJKML`.

The first hint given tells us that the secret message begins with "`ONE...`". By comparing the first three letters "`ONF`" of the ciphertext with "`ONE`" one can suppose that a polyalphabetic cipher is used. The hint about turns gives us an idea that each simple cipher is a substitution Ceasar cipher with some shift, where each turn in the labyrinth increase shift in the alphabet (Table 4).

Table 4

Path	O	N	F	I	W	Q	H	W	J	J	F	L	H	Z	A	O	A	X	W	E	S	P	P	N
Shift	0	0	1	1	2	3	4	5	5	6	7	7	7	8	8	9	9	9	10	11	11	11	12	13
Message	O	N	E	H	U	N	D	R	E	D	Y	E	A	R	S	F	R	O	M	T	H	E	D	A

Path	G	R	C	T	P	X	G	J	X	F	W	U	D	T	O	X	Y	M	C	W	J	K	M	L
Shift	13	13	14	14	14	15	15	16	16	17	17	18	18	19	20	20	20	20	21	22	22	23	24	24
Message	T	E	O	F	B	I	R	T	H	O	F	C	L	A	U	D	E	S	H	A	N	N	O	N

Thus, the secrete message can be read as

«One hundred years from the date of birth of Claude Shannon».

It was devoted to the anniversary of an outstanding mathematician, electrical engineer, and cryptographer Claude Elwood Shannon (April 30, 1916 — February 24, 2001).

The problem was completely solved by 46 participants of all categories in both rounds. The most beautiful solutions were presented by Arina Prostakova (Gymnasium 94, Yekaterinburg), Maria Tarabarina (Lomonosov Moscow State University), Dragos Alin Rotaru, Marco Martinoli, and Tim Wood team (University of Bristol, United Kingdom), Nguyen Duc, Bui Minh Tien Dat, and Quan Doan team (University of Information Technology, Vietnam), Henning Seidler and Katja Stumpp team (Technical University of Berlin).

4.5. Problem "System of equations"

Solution. One can notice these equations can be grouped under the three subsystems:

$$1)\begin{cases} x_1 \oplus x_2 \oplus x_3 \oplus x_5 \oplus x_6 = 0, \\ x_1x_3 \oplus x_2x_4 = x_5 - x_6, \\ (x_2 + x_3 + x_4)^2 = 2(x_1 + x_5 + x_6), \\ x_2x_3x_4 \oplus x_1x_5x_6 = 0; \end{cases} \qquad 2)\begin{cases} x_6 \oplus x_7 \oplus x_8 \oplus x_{10} \oplus x_{11} = 0, \\ x_6x_8 \oplus x_7x_9 = x_{10} - x_{11}, \\ (x_7 + x_8 + x_9)^2 = 2(x_6 + x_{10} + x_{11}), \\ x_7x_8x_9 \oplus x_6x_{10}x_{11} = 0; \end{cases}$$

$$3)\begin{cases} x_{11} \oplus x_{12} \oplus x_{13} \oplus x_{15} \oplus x_{16} = 0, \\ x_{11}x_{13} \oplus x_{12}x_{14} = x_{15} - x_{16}, \\ (x_{12} + x_{13} + x_{14})^2 = 2(x_{11} + x_{15} + x_{16}), \\ x_{11}x_{12}x_{13} \oplus x_{14}x_{15}x_{16} = 0. \end{cases}$$

Note that the first and the second subsystems have a common variable x_6, just like the second and the third ones both involve x_{11}.

The first two are the subsystems having the template

$$\begin{cases} y_1 \oplus y_2 \oplus y_3 \oplus y_5 \oplus y_6 = 0, \\ y_1y_3 \oplus y_2y_4 = y_5 - y_6, \\ (y_2 + y_3 + y_4)^2 = 2(y_1 + y_5 + y_6), \\ y_2y_3y_4 \oplus y_1y_5y_6 = 0, \end{cases}$$

but the third subsystem has the following one

$$\begin{cases} y_1 \oplus y_2 \oplus y_3 \oplus y_5 \oplus y_6 = 0, \\ y_1y_3 \oplus y_2y_4 = y_5 - y_6, \\ (y_2 + y_3 + y_4)^2 = 2(y_1 + y_5 + y_6), \\ y_1y_2y_3 \oplus y_4y_5y_6 = 0. \end{cases}$$

In both of them variables are $(y_1, y_2, y_3, y_4, y_5, y_6) = \mathbf{y}$, where $y_i \in \{0, 1\}$, $i = 1, 2, \ldots, 6$. Obviously, one of the solutions of both templates is equal to $\mathbf{y} = (0, 0, 0, 0, 0, 0)$; we denote it by $\mathbf{y}^1$.

Let us consider the first template and find all its solutions.

In the case $\underline{y_5 = y_6} = y \in \{0, 1\}$ we have

$$\begin{cases} y_1 \oplus y_2 \oplus y_3 = 0, \\ y_1 y_3 \oplus y_2 y_4 = 0, \\ (y_2 + y_3 + y_4)^2 = 2(y_1 + 2y), \\ y_2 y_3 y_4 \oplus y_1 y = 0. \end{cases}$$

Evidently, the third equation holds if

a) $y_2 + y_3 + y_4 = 0$ and $y_1 + 2y = 0$; or b) $y_2 + y_3 + y_4 = 2$ and $y_1 + 2y = 2$.

From a we obtain $\mathbf{y}^1$. Using the case b and the equation $y_1 \oplus y_2 \oplus y_3 = 0$ we receive $y = 1$, $y_1 = 0$, $y_2 \oplus y_3 = 0$ (i.e. $y_2 = y_3 = y' \in \{0, 1\}$) and $2y' + y_4 = 2$. Thus, we have $\mathbf{y} = (0, 1, 1, 0, 1, 1)$, which evidently satisfies the other equations of the template. So, the second solution is $(0, 1, 1, 0, 1, 1)$; we denote it by $\mathbf{y}^2$.

In the case $\underline{y_5 \neq y_6}$ (i.e. $y_5 = 1$, $y_6 = 0$ since it must hold $y_5 \geqslant y_6$ by virtue of the second equation) we have

$$\begin{cases} y_1 \oplus y_2 \oplus y_3 \oplus 1 = 0, \\ y_1 y_3 \oplus y_2 y_4 = 1, \\ (y_2 + y_3 + y_4)^2 = 2(y_1 + 1), \\ y_2 y_3 y_4 = 0. \end{cases}$$

Again, the third equation holds if

a) $y_2 + y_3 + y_4 = 0$ and $y_1 + 1 = 0$; or b) $y_2 + y_3 + y_4 = 2$ and $y_1 + 1 = 2$.

The case a is impossible for binary y_1. Using the case b and the equation $y_1 \oplus y_2 \oplus y_3 = 1$ we receive $y_1 = 1$, $y_2 \oplus y_3 = 0$ (i.e. $y_2 = y_3 = y'' \in \{0, 1\}$) and $2y'' + y_4 = 2$. In this way we have $\mathbf{y} = (1, 1, 1, 0, 1, 0)$, which evidently satisfies the other equations of the template. So, the third solution is $(1, 1, 1, 0, 1, 0)$; we denote it by $\mathbf{y}^3$.

Let us consider the second template. It only differs from the first template in the fourth equation. But at the same time, this equation has not been used while obtaining the solutions besides checking. So, it is enough to check whether these solutions satisfy the equation $y_1 y_2 y_3 \oplus y_4 y_5 y_6 = 0$. Obviously, $\mathbf{y}^1$ and $\mathbf{y}^2$ are suitable, but $\mathbf{y}^3$ does not satisfy it.

Thus, after considering the links between corresponding subsystems that involve common variables, we get that all solutions of the initial system are the following:

1. (00000**0**0000**0**00000); 3. (01101**1**1101**0**00000); 5. (11101**0**0000**0**00000);
2. (00000**0**0000**0**11011); 4. (01101**1**1101**0**11011); 6. (11101**0**0000**0**11011).

The problem was completely solved by 26 participants, 12 of whom used computer calculations. Many participants have noticed it is possible to solve the system separately by taking into account the structure of the whole system. But at the same time, some of them have not considered the difference between the templates, in such cases extra solutions have been included in the problem's answer. The right solutions with good explanation were made by Alexandr Grebennikov (Presidential Physics and Mathematics Lyceum 239, Saint Petersburg), Dmitry Morozov (Novosibirsk State University), Anna Gusakova (Institute of Mathematics of National Academy of Sciences of Belarus).

4.6. Problem “Biometric pin-code”

Solution. First we compute $s' = b \oplus c$, as described in the algorithm, and obtain $s' = (1000\,0111\,1101\,0000)$. Note that, since a new biometric image b of Alice can be different from the image that was taken during creation of her biometrically encrypted key (by not more than 10–20 %), s' can also be different from Hadamard code codeword s corresponding to real Alice's key k, but by 10–20 % at most.

So, in order to find Alice's key, we need to find a codeword of Hadamard code of length 16, which differs from s' in not more than 20 % of bits. Since all codewords of Hadamard code differ from each other in at least 50 % of bits, such a codeword is unique (in case it exists). Thus, we can simply search for the codeword closest to s'.

Since Hadamard code of length 16 has $2^5 = 32$ codewords, we can easily (with or without use of computer) find the closest codeword: it is $s = (0000111111110000)$ and it corresponds to the key $k = (11000)$. We see that s and s' differ in only 3 bits, which is 18.75 % of all 16 bits. So the key k that we have found fits all conditions. Therefore, it is the real key of Alice.

Unfortunately, there were no complete solutions of the problem by school students.

4.7. Problem “Quadratic functions”

Solution. Here we would like to describe the solution by George Beloshapko, Stepan Gatilov, and Anna Taranenko team (Novosibirsk State University, Ledas, Sobolev Institute of Mathematics) as the simplest.

Let us consider the sequences $\{u_i\}$ and $\{v_i\}$ generated by the functions $f_u(x_1,\ldots,x_n) = x_1x_2$ and $f_v(x_1,\ldots,x_n) = x_2x_n \oplus x_1 \oplus x_n$ respectively with the initial value $\underbrace{1\ldots10}_{n}$. Let us describe the first $n^2 - n + 1$ elements of the sequences:

$$\begin{aligned}
\{u_i\} &= \underbrace{1\ldots1}_{n-1}\underbrace{0}_{1}\underbrace{1\ldots1}_{n-2}\underbrace{00}_{2}\ldots\underbrace{11}_{2}\underbrace{0\ldots0}_{n-2}\underbrace{1}_{1}\underbrace{0\ldots0}_{n-1}0\ldots\\
\{v_i\} &= \underbrace{1\ldots1}_{n-1}\underbrace{0}_{1}\underbrace{1\ldots1}_{n-2}\underbrace{00}_{2}\ldots\underbrace{11}_{2}\underbrace{0\ldots0}_{n-2}\underbrace{1}_{1}\underbrace{0\ldots0}_{n-1}1\ldots
\end{aligned}$$

The first $n^2 - n$ elements of the sequences are the same, but the next elements differ. So, it is impossible to uniquely reconstruct the sequence by the segment of length cn, where c is a constant.

There were no complete solutions of the problem in the first round. In the second round, in addition to the given solution, the problem was completely solved by Alexey Udovenko (University of Luxembourg), Alexey Ripinen, Oleg Smirnov, and Peter Razumovsky team (Saratov State University), Maxim Plushkin, Ivan Lozinskiy, and Alexey Solovev team (Lomonosov Moscow State University).

4.8. Problem “Biometric key”

Solution. To solve the problem, we compute values s' for both persons X and Y and check if any of them is close enough to some Hadamard code codeword, corresponding to the key with odd number of 1's. Here “close enough” means difference in not more than 20 % bits, since two biometric images of the same person can not be more different from each other.

Let us denote these values s^X and s^Y for person X and Y respectively. Now, with the use of a computer program, we can go through all 256 codewords of Hadamard code of length 128, and find those which are closest to s^X and s^Y.

For person Y, the closest Hadamard codeword differs from s^Y by 49 bits (around 38.28 %). This means that there exists no key k, using which person Y could make himself biometrically encrypted key c.

For person X, the closest Hadamard codeword is obtained from (11011010) and it differs from s^X by 25 bits (around 19.53 %). This means that person X is Alice and her 8-bit key is $k = (11011010)$ (it also has an odd number of 1's, which solidifies our confidence).

Note that there are many interesting approaches for combining biometrics with cryptography (for example, see [4]).

The problem was completely solved by Irina Slonkina (Novosibirsk State University of Economics and Management) and George Beloshapko (Novosibirsk State University) in the first round, and by 10 teams in the second round, Evgeniya Ishchukova, Ekaterina Maro, and Dmitry Alekseev team (Southern Federal University, Taganrog) was among them.

4.9. Problem "Secret sharing"

Solution. Without Sergey, Alena and Boris know the following vector:

$$P' = P \oplus \bigoplus_{i=1}^{32} c_i^s v_i^s.$$

So, they do not know any information about P if and only if the dimension of the linear span $\langle v_1^s, \ldots, v_{32}^s\rangle$ is equal to 32, i.e. $v_1^s, \ldots, v_{32}^s$ form basis of $\mathbb{F}_2^{32}$. Otherwise, $\langle v_1^s, \ldots, v_{32}^s\rangle \subset \mathbb{F}_2^{32}$ and $P \in P' \oplus \langle v_1^s, \ldots, v_{32}^s\rangle$, it means that there are not more than $2^{\mathrm{rk}\langle v_1^s,\ldots,v_{32}^s\rangle}$ ways for Alena and Boris to get P.

Let V be a 32×32 matrix with columns $v_1^s, \ldots, v_{32}^s$. Note that $\mathrm{rk}V = \mathrm{rk}\langle v_1^s, \ldots, v_{32}^s\rangle$. Since all bits of $v_1^s, \ldots, v_{32}^s$ are randomly generated (and independent of each other), all elements of V are randomly generated too.

Hence, the probability p_1 that Alena and Boris can not get any information about P is equal to $N_1/2^{32\cdot 32}$, where N_1 is the number of matrices V of rank 32:

$$p_1 = \frac{(2^{32} - 2^0)(2^{32} - 2^1) \cdot \ldots \cdot (2^{32} - 2^{31})}{2^{32\cdot 32}} \approx 0.288788.$$

In order for Alena and Boris to get a guaranteed access to online banking without Sergey using not more than 23 attempts, it should hold $2^{\mathrm{rk}\langle v_1^s,\ldots,v_{32}^s\rangle} \leqslant 23$, i.e. $\mathrm{rk}\langle v_1^s, \ldots, v_{32}^s\rangle \leqslant 4$. So, the probability p_2 of that is equal to $N_2/2^{32\cdot 32}$, where N_2 is the number of matrices V of rank not more than 4. Note that the number of $n \times n$ matrices of rank k is equal to

$$R_n^k = \frac{(2^n - 2^0)^2(2^n - 2^1)^2 \cdot \ldots \cdot (2^n - 2^{k-1})^2}{(2^k - 2^0)(2^k - 2^1) \cdot \ldots \cdot (2^k - 2^{k-1})}.$$

Therefore,

$$p_2 = \frac{R_{32}^0 + R_{32}^1 + R_{32}^2 + R_{32}^3 + R_{32}^4}{2^{32\cdot 32}} \approx 1.625 \cdot 2^{-783}.$$

In the first round, the problem was completely solved by Alexey Udovenko (University of Luxembourg) and Igor Fedorov (Novosibirsk State University); almost completely solved by Mohammadjavad Hajialikhani (Sharif University of Technology, Iran), Pavel Hvoryh (Omsk State Technical University), George Beloshapko (Novosibirsk State University) and Ekaterina Kulikova (Munich, Germany). In the second round (in addition to the first round) the problem was completely solved by Aliaksei Ivanin, Oleg Volodko, and Konstantin Pavlov team (Belarusian State University).

4.10. Problem "Protocol"

Solution. Here we would like to describe the solution proposed by Alexey Udovenko (University of Luxembourg), that is similar to the author's one, but is more elegant and compact.

The session key is equal to

$$K_{a,b} = g^{R_a R_b} = g^{(X_a-\alpha_a)(X_b-\alpha_b)} = g^{X_a X_b - \alpha_a X_b - \alpha_b X_a + \alpha_a \alpha_b} \bmod p.$$

Evgeniy observes X_a and X_b and he also knows g, $P_a = g^{\alpha_a} \bmod p$ and $P_b = g^{\alpha_b} \bmod p$. From g, P_a, P_b and one exposured key he can compute

$$s = g^{\alpha_a \alpha_b} = K_{a,b}/(g^{X_a X_b} P_a^{-X_b} P_b^{-X_a}) \bmod p.$$

Then for new sessions he can intercept new X_a and X_b and easily compute new

$$K'_{a,b} = g^{X_a X_b} P_a^{-X_b} P_b^{-X_a} s \mod p.$$

It worth noting that recovering of the next keys would be impossible if the protocol has a property of so-called *forward secrecy*. More details can be found in [5].

This problem was completely solved by 7 participants in the first round, Robert Spencer (University of Cape Town, South Africa) was among them, and by 15 teams in the second round, Roman Lebedev, Ilia Koriakin, and Vlad Kuzin team (Novosibirsk State University) was among them. All solutions proposed were made in the similar way. Also there were solutions with reduced score, containing some inexactness, but they were still very close to complete ones.

4.11. Problem "Zerosum at AES"

Solution. Let us describe one of the simplest approach to get the solution proposed and implemented by Alexey Udovenko (University of Luxembourg).

Denote by $Y_i = \texttt{AES}_0(X_i) \oplus X_i$. The equation can be rewritten:

$$\bigoplus_{i=1}^{128}(\texttt{AES}_0(X_i) \oplus X_i) = \bigoplus_{i=1}^{128} Y_i = 0.$$

First we encrypt random 256 plaintexts and obtain $Y_1, \ldots, Y_{256}$. Then the problem is to find distinct indices $i_1, \ldots, i_{128}$ such that $Y_{i_1} \oplus \ldots \oplus Y_{i_{128}} = 0$.

Let M be the 128×256 binary matrix with columns $Y_1, \ldots, Y_{256}$. Let us consider the solutions of the linear equation $Mz = 0$: for any solution z it holds

$$\bigoplus_{i:z_i=1} Y_i = 0.$$

Moreover, the Hamming weight of a random solution vector z will be close to 128 and with high probability equal to 128. Next, we try to find a solution vector of weight 128.

All information about the block cipher AES can be found in the book [6] of AES authors J. Daemen and V. Rijmen.

The problem was completely solved (proposed an algorithm and the solution) by three teams: Alexey Udovenko (University of Luxembourg), George Beloshapko, Stepan Gatilov, and Anna Taranenko team (Novosibirsk State University, Ledas, Sobolev Institute of Mathematics) and Maxim Plushkin, Ivan Lozinskiy, and Alexey Solovev team (Lomonosov Moscow State University).

4.12. Problem "Latin square"

Solution. A $n \times n$ latin square can be given as the set of its columns permutation σ_i, $i = 1, \ldots, n$. Then, the answer t_3 on the request *abcd* can be calculated as $\sigma_d \sigma_c \sigma_b(a)$ (here $\sigma = \sigma_d \sigma_c \sigma_b$ denotes the permutations composition). Choosing all possible $a \neq b$, one is able to reconstruct σ. It is needed 9 request-answer pairs. The inverse permutations can also be found, $\sigma^{-1} = \sigma_b^{-1} \sigma_c^{-1} \sigma_d^{-1}$.

Moreover, for any distinct $i, j \in \{0, 1, \ldots, 9\}$ the following permutation is recovered using 18 request-answer pairs:

$$\sigma_j^{-1} \sigma_i = (\sigma_j^{-1} \sigma_c^{-1} \sigma_d^{-1})(\sigma_d \sigma_c \sigma_i).$$

Here c, d are arbitrary numbers from the set $\{0, 1, \ldots, 9\} \setminus \{i, j\}$.

If one knows $\sigma_j^{-1} \sigma_i$, then he also knows 10 preimage–image pairs x, y: $y = \sigma_j^{-1} \sigma_i(x)$. Each of such a pair means an equality $\sigma_i(x) = \sigma_j(y)$, from which σ_j can be expressed by σ_i.

For example, let us express σ_j, $j = 1, 2, \ldots, 9$, by σ_0. To do this we need $18 \cdot 9 = 162$ pairs of request–answer. Then we will check all possible variants of σ_0 (there are 10! such variants). For each variant $\hat{\sigma}_0$ we find $\hat{\sigma}_j$, build the corresponding latin square $\hat{L}$ and check whether answers on $\hat{L}$ is equal to answers on the secret latin square. Using $162 \gg \log_{10} 10!$ answers Alice's secret key will be uniquely recovered with high probability.

The answer is the following 10×10 latin square (Fig. 8).

3	9	2	1	7	8	5	0	6	4
4	3	7	9	1	5	2	6	8	0
1	8	6	7	0	9	4	3	5	2
2	6	5	3	9	7	1	4	0	8
6	0	9	4	5	1	8	2	3	7
0	7	3	6	2	4	9	8	1	5
8	4	1	0	3	2	6	5	7	9
5	1	4	8	6	0	7	9	2	3
7	2	8	5	4	3	0	1	9	6
9	5	0	2	8	6	3	7	4	1

Fig. 8.

The problem was completely solved by 26 teams. The best solution was proposed by Sergey Titov (Ural State University of Railway Transport, Yekaterinburg).

4.13. Problem "*nsucoin*"

Solution. The payment method used is an example of blockchain based money working on the proof-of-work principal. The first such a system *bitcoin* was proposed in [7].

To solve this problem one need to restore a history of transactions that leads to the block on height 2 from condition of the problem:

```
height:2; prevHash:0000593b; ctxHash:8fef76cb; nonce:17052
```

Given such a history of transactions one can find how many flowers of different kinds Alice has at the end of trading.

Solution plan consists of the following steps:

1. One need to find each user's private key (using private keys one is able to make a sign of each user and as a result generate transactions).

2. One need to find all special transactions that give to each user 10 coins.

3. One need to find all possible blocks on height 0.

4. Looking through all possible transactions, one need to find blocks on height 1 for each block on height 0 so that hash of each block found will be equal to value of prevHash field of the given block, i.e. `0000593b`.

5. It is known that there is at least one block among the blocks found on the previous step such that there exist transactions such that hash of these transactions concatenation is equal to `8fef76cb`. One need to find these transactions.

6. Thus, all transactions history leading to the given block on height 2 is found. It remains to track the movement of flowers and give an answer.

While searching blocks we need to remember that `nonce` does not exceed 40000. Despite the fact that there are 24 transpositions of special transactions, only 6 of them can be verified by a block. Also, it is useful to remember that each participant can sell only 5 flowers, each of them costs 2 coins, and a participant can not buy anything if he does not have coins.

It is easy to factor a given small module of RSA:

$$n = p \cdot q = 2250339337 \cdot 4044301367 = 9101050456842973679,$$
$$\varphi(n) = (p-1)(q-1) = 9101050450548332976.$$

Then, one can find private keys as inverse numbers to public keys module $\varphi(n)$ (Table 5).

Table 5

User	PubKey	PrivKey
Alice	11	2482104668331363539
Bob	17	3747491361990490049
Caroline	199	9009582606824229127
Daniel	5	7280840360438666381

Despite all the limitations of searching, it turns out to be quite time-consuming. There were laid some hints in the condition of the problem allowing one to get a solution. This two hints were published on the Olympiad website:

18 November. A tip to reduce exhaustive search: The transaction from the example has been verified by the block with height 1.

20 November. All hashes of transactions that are verified by the block from the example correspond to hashes of transactions verified by the block with height 1 into the sought-for blockchain.

The first participant who found out the right answer was Alexey Udovenko (University of Luxembourg). He based only on the first hint and guessed that all four transactions hashes from the example are contained in the block with height 1. By the other words, he guessed the second hint and found the answer we had conceived.

In fact, we knew only one answer but we hoped that someone was able to find another one, and it happened! There were two teams that found two answers based only on the first hint: George Beloshapko, Stepan Gatilov, and Anna Taranenko team (Novosibirsk State University, Ledas, Institute of Mathematics) and the team of Alexey Ripinen, Oleg Smirnov, and Peter Razumovsky team (Saratov State University).

The final flowers distribution and transactions history of two solutions are presented in Tables 6, 7.

There were 9 teams received scores for this problem, 2 teams found two answers, 1 found an answer based only on the first hint, 2 found answer based on both hints, 3 teams found blocks with height 0 and 1 but did not found transactions verified by block with height 2, and 1 team found only a block with height 0. Probably, someone were not able to find

Table 6

Flowers distribution of two solutions

User	Camomiles		Tulips		Gerberas		Roses	
	1st sol.	2d sol.	1st sol.	2d sol.	1st sol.	2d sol.	1st sol.	2d sol.
Alice	1	1	2	3	3	0	1	0
Bob	4	2	0	2	2	2	0	1
Caroline	0	2	0	0	0	1	1	0
Daniel	0	0	3	0	0	2	3	4

answer due to the following fact: each transaction to be verified can be based not only on already verified transactions but also on transactions based on verified transactions. It is transactions of the second type that are in blocks. Also, the difficulties for the decisive could create transactions based on two other transactions. There is one such transaction verified by the block with height 2 in the first answer and four such transactions in the second answer.

Special thanks to the team of George Beloshapko, Stepan Gatilov, and Anna Taranenko for illustrating the solutions (Fig. 9).

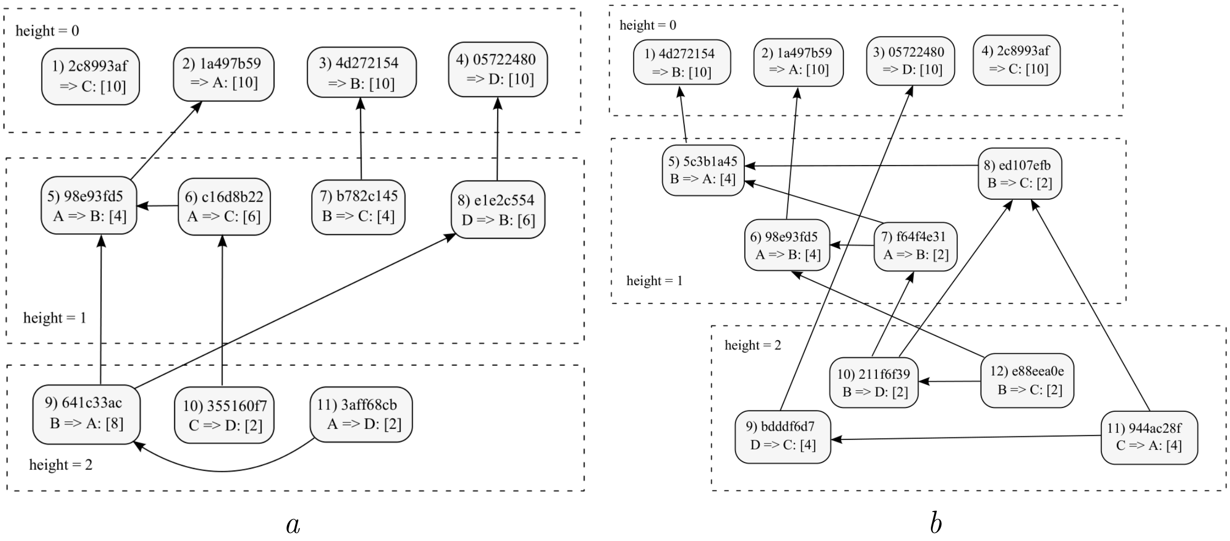


Fig. 9. Illustrations of transactions history of two solutions.

4.14. Problem "Metrical cryptosystem"

Solution. First of all, let us reformulate the problem: the statement "The cryptosystem presented is correct" is equivalent to the statement "It holds $\mathrm{d}(x, A) + \mathrm{d}(x, B) = d$ for any pair of metrically regular sets $A, B \subseteq \mathbb{F}_2^n$ at the Hamming distance d from each other and any vector $x \in \mathbb{F}_2^n$".

After considering pairs of metrically regular sets in spaces of small dimension, one may conclude that the cryptosystem is correct, and try to prove it. However, it is not correct. Here we will present the solution of George Beloshapko, Stepan Gatilov, and Anna Taranenko team (Novosibirsk State University, Ledas, Sobolev Institute of Mathematics) as the most elegant.

Table 7

Transactions history of two solutions

First solution:

```
Tx1: txHash:2c8993af;inputTx:;value1:10;pubKey1:199;sign1:
Tx2: txHash:1a497b59;inputTx:;value1:10;pubKey1:11;sign1:
Tx3: txHash:4d272154;inputTx:;value1:10;pubKey1:17;sign1:
Tx4: txHash:05722480;inputTx:;value1:10;pubKey1:5;sign1:
Block 1: height:0;prevHash:00000000;ctxHash:bde430dd;nonce:21095 with hash 00003cc0
Tx5: txHash:98e93fd5;inputTx:1a497b59;value1:4;pubKey1:17;sign1:689e297682a9e6f0;value2:6;pubKey2:11;sign2:fec9245898b829c
Tx6: txHash:c16d8b22;inputTx:98e93fd5;value1:6;pubKey1:199;sign1:611cddad83c8e352;value2:0;pubKey2:11;sign2:6ab37bb9f5f4249d
Tx7: txHash:b782c145;inputTx:4d272154;value1:4;pubKey1:199;sign1:5f50adf4a51899e5;value2:6;pubKey2:17;sign2:78a4f7d6d0d578d7
Tx8: txHash:e1e2c554;inputTx:05722480;value1:6;pubKey1:17;sign1:b91ab453e4bcb53;value2:4;pubKey2:5;sign2:3803907748416c12
Block 2: height:1;prevHash:00003cc0;ctxHash:9f8333d4;nonce:21438 with hash 0000593b
Tx9: txHash:641c33ac;inputTx:98e93fd5,e1e2c554;value1:8;pubKey1:11;sign1:110a7c6e1dfd937;value2:2;pubKey2:17;sign2:14ad50a36ef5540d
Tx10: txHash:3aff68cb;inputTx:641c33ac;value1:2;pubKey1:5;sign1:20281a4d62b20cb7;value2:6;pubKey2:11;sign2:7b662b128b1200bf
Tx11: txHash:3aff68cb;inputTx:641c33ac;value1:2;pubKey1:5;sign1:20281a4d62b20cb7;value2:6;pubKey2:11;sign2:7b662b128b1200bf
Block 3: height:2;prevHash:0000593b;ctxHash:8fef76cb;nonce:17052 with hash 000023d4
```

Second solution:

```
Tx1: txHash:4d272154;inputTx:;value1:10;pubKey1:17;sign1:
Tx2: txHash:1a497b59;inputTx:;value1:10;pubKey1:11;sign1:
Tx3: txHash:05722480;inputTx:;value1:10;pubKey1:5;sign1:
Tx4: txHash:2c8993af;inputTx:;value1:10;pubKey1:199;sign1:
Block 1: height:0;prevHash:00000000;ctxHash:4ddc0244;nonce:20670 with hash 0000857a
Tx5: txHash:txHash:5c3b1a45;inputTx:4d272154;value1:4;pubKey1:11;sign1:5866152e5bf782f1;value2:6;pubKey2:17;sign2:78a4f7d6d0d578d7
Tx6: txHash:98e93fd5;inputTx:1a497b59;value1:4;pubKey1:17;sign1:689e297682a9e6f0;value2:6;pubKey2:11;sign2:fec9245898b829c
Tx7: txHash:f64f4e31;inputTx:98e93fd5,5c3b1a45;value1:2;pubKey1:17;sign1:6a9369e096c2cd05;value2:8;pubKey2:11;sign2:5021efb4fb05e703
Tx8: txHash:ed107efb;inputTx:5c3b1a45;value1:2;pubKey1:199;sign1:7e12b526fd676d32;value2:4;pubKey2:17;sign2:4d9942d6d31e6392
Block 2: height:1;prevHash:0000857a;ctxHash:00f229f7;nonce:19574 with hash 0000593b
Tx9: txHash:bdddf6d7;inputTx:05722480;value1:4;pubKey1:199;sign1:6b33ced4b96ed36f;value2:6;pubKey2:5;sign2:3803907748416c12
Tx10: txHash:211f6f39;inputTx:ed107efb,f64f4e31;value1:2;pubKey1:5;sign1:6b12e0f356e951ea;value2:4;pubKey2:17;sign2:88b48219f607775
Tx11: txHash:944ac28f;inputTx:ed107efb,bdddf6d7;value1:4;pubKey1:11;sign1:395867768ee9f790;value2:2;pubKey2:199;sign2:555727e07c7e0c97
Tx12: txHash:e88eea0e;inputTx:211f6f39,98e93fd5;value1:2;pubKey1:199;sign1:76a91f809d7468b7;value2:6;pubKey2:17;sign2:1c6905596113f9a6
Block 3: height:2;prevHash:0000593b;ctxHash:8fef76cb;nonce:17052 with hash 000023d4
```

Consider the Hadamard code of length 16 as A:

$$\begin{aligned}A = \{&(0000000000000000),\ (0000000011111111), (0000111100001111),\ (0000111111110000),\\ &(0011001100110011),\ (0011001111001100), (0011110000111100),\ (0011110011000011),\\ &(0101010101010101),\ (0101010110101010), (0101101001011010),\ (0101101010100101),\\ &(0110011001100110),\ (0110011010011001), (0110100101101001),\ (0110100110010110)\}.\end{aligned}$$

In other words, A is the set of values vectors of all linear functions in 4 variables. It is easy to check (for example, using computer program), that metrical complement B of the set A is the set of values vectors of all affine function (excluding linear) in 4 variables and vice versa, and the distance between A and B is equal to 8:

$$\begin{aligned}B = \{&(1001011001101001),\ (1001011010010110), (1001100101100110),\ (1001100110011001),\\ &(1010010101011010),\ (1010010110100101), (1010101001010101),\ (1010101010101010),\\ &(1100001100111100),\ (1100001111000011), (1100110000110011),\ (1100110011001100),\\ &(1111000000001111),\ (1111000011110000), (1111111100000000),\ (1111111111111111)\}.\end{aligned}$$

Now consider vector $x = (0000000000010111)$. It is at distance 4 from set A and at distance 6 from set B, therefore

$$\mathrm{d}(x, A) + \mathrm{d}(x, B) = 10 > 8,$$

which shows us that the cryptosystem is incorrect. Other solutions include straightforward computer search for metrically regular sets and vectors, for which $\mathrm{d}(x, A) + \mathrm{d}(x, B) \neq d$. The smallest dimension for which example is found is equal to 7. Despite the fact that the cryptosystem is not correct, many participants, who tried to prove that the cryptosystem is correct, received 1 or 2 points for creative ideas.

4.15. Problem "Algebraic immunity"
(Unsolved, Special Prize)

Solution. The problem was completely solved! It was the first time in the Olympiad history. We would like to describe part of the solution proposed by Alexey Udovenko (University of Luxembourg). At first we mentioned that the notion of component algebraic immunity was proposed by C. Carlet in [8].

"It is natural to try simple constructions to build a vectorial Boolean function from a 1-bit Boolean function. Let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be some function such that $\mathrm{AI}(f) = \lceil n/2 \rceil$. Let $F_{f,m} : \mathbb{F}_2^n \to \mathbb{F}_2^m$ be defined as $F_{f,m}(x) = f(x)||f(L(x))||f(L^2(x))||\ldots||f(L^{m-1}(x))$, where L is some linear mapping, for example rotation of the input vector left by one position. We generated random Boolean functions f and for those which had $\mathrm{AI}(f) = \lceil n/2 \rceil$ we found the largest m such that the corresponding $F_{f,m}$ had high algebraic immunity too, i.e. such that $\mathrm{AI}_{\mathrm{comp}}(F_{f,m}) = \lceil n/2 \rceil$. Note that this approach can produce functions only with $m \leqslant n$ if L is the rotation mapping. Here are our results (L is always rotation left by one):

- $n = 3$: Let $f(x_0, x_1, x_2) = x_0 + x_1 + x_1x_2$. Then $\mathrm{AI}_{\mathrm{comp}}(F_{f,3}) = 2$.
- $n = 4$: Let $f(x_0, \ldots, x_3) = x_0x_1x_2 + x_0x_1 + x_3$. Then $\mathrm{AI}_{\mathrm{comp}}(F_{f,4}) = 2$.
- $n = 5$: Let $f(x_0, \ldots, x_4) = x_0x_1x_2x_3 + x_0x_1x_2 + x_0x_1x_3 + x_0 + x_1x_3 + x_2x_4 + x_4$. Then $\mathrm{AI}_{\mathrm{comp}}(F_{f,5}) = 3$. Interestingly, $F_{f,5}$ happens to be a permutation. Its lookup table is as follows: (0, 24, 12, 20, 6, 29, 10, 1, 3, 23, 30, 26, 5, 22, 16, 19, 17, 9, 27, 2, 15, 21, 13, 7, 18, 4, 11, 14, 8, 28, 25, 31)."

Alexey Udovenko also found a function with optimum component algebraic immunity $\lceil n/2 \rceil$ by a rotational symmetries construction for the following values (n, m): (6,6), (7,3), (8,8), (9,2), (10,10). Moreover, he found function with the maximum component algebraic immunity by random search for the $(n, m) \in \{(4, 5), (6, 8)\}$.

Additionally, he has found F' which is also a permutation of $\mathbb{F}_2^5$ such that $\mathrm{AI}_{\mathrm{comp}}(F') = 3$ but which is differentially 2-uniform (is APN) and its nonlinearity is equal to 12. Therefore, it is more suitable for cryptography. However, the algebraic degree of F' is equal to 3. The lookup table of F' is as follows: $F' =$ (0, 12, 6, 11, 3, 25, 21, 4, 17, 7, 28, 9, 26, 10, 2, 27, 24, 22, 19, 8, 14, 18, 20, 23, 13, 16, 5, 15, 1, 30, 29, 31).

Alexey Udovenko was the only person who completely solved this problem. We also presented several ideas for finding constructions of vectorial Boolean function with optimum component algebraic immunity, but unfortunately it was not completed.

4.16. Problem “Big Fermat numbers”
(Unsolved, Special Prize)

Solution. There was no complete solution of this problem. The best solution was proposed by Alisa Pankova (University of Tartu, Estonia). The main idea was to show how to use a prime Fermat number $2^{2^k} + 1$ to construct a larger composite Fermat number $2^{2^n} + 1$ for a certain $n > k$. In this way, if there is an arbitrarily large prime Fermat number, then there exists even larger composite Fermat number, so there is no point after which all Fermat numbers become prime. Unfortunately, there was a mistake in the solution.

The team of Vadzim Marchuk, Anna Gusakova, and Yuliya Yarashenia (Belarusian State University) found a very nice heuristic bound but the statement was not proven with probability 1. Several interesting attempts were also proposed by the team of Alexey Udovenko (University of Luxembourg), George Beloshapko, Stepan Gatilov, and Anna Taranenko team (Novosibirsk State University, Ledas, Sobolev Institute of Mathematics), Roman Ginyatullin, Anatoli Makeyev, and Victoriya Vlasova team (Moscow Engineering Physics Institute), Nikolay Altukhov, Vladimir Bushuev, and Roman Chistiakov team (Bauman Moscow State Technical University), Evgeniy Manaka, Aleksandr Sosenko, and Pavel Ivannikov team (Bauman Moscow State Technical University).

5. Winners of the Olympiad

At all, 10 school students, 19 university students and 5 professionals were awarded by diplomas in the first round, 16 university students teams and 11 professionals teams were awarded in the second round of NSUCRYPTO-2016.

The first places in different categories were won by the following participants:

- *First round, Section A, school student*: **Alexander Grebennikov** (Russia, Saint Petersburg, Presidential PML 239);
- *First round, Section B, university student*: **Robert Spencer** (South Africa, Cape Town, University of Cape Town) and **Henning Seidler** (Germany, Berlin, Technische Universität Berlin);
- *First round, Section B, professional*: **Alexey Udovenko** (Luxembourg, Luxembourg, University of Luxembourg);
- *Second round, university student*: **Maxim Plushkin, Ivan Lozinskiy,** and **Alexey Solovev** team (Russia, Moscow, Lomonosov Moscow State University);
- *Second round, professional*: **Alexey Udovenko** (Luxembourg, Luxembourg, University of Luxembourg). Note that Alexey also got a special prize for solving an unsolved problem “Algebraic immunity”.

All information about the winners of NSUCRYPTO-2016 can be found on the official website at `http://nsucrypto.nsu.ru/archive/2016/`

Acknowledgements

We would like to thank Sergei Kiazhin for his interesting ideas of the problems. We thank Novosibirsk State University for the financial support of the Olympiad and invite you to take part in the next NSUCRYPTO that will be held in October, 2018. Your ideas on the mentioned unsolved problems are also very welcome and can be sent to `nsucrypto@nsu.ru`.

REFERENCES

1. *Agievich S., Gorodilova A., Kolomeec N., et al.* Problems, solutions and experience of the first international student's Olympiad in cryptography. Prikladnaya Diskretnaya Matematika, 2015, no. 3, pp. 41–62.
2. *Agievich S., Gorodilova A., Idrisova V., et al.* Mathematical problems of the second international student's Olympiad in cryptography. Cryptologia, 2017, vol. 41, iss. 6, pp. 534–565.
3. *Geut K., Kirienko K., Sadkov P., et al.* On explicit constructions for solving the problem "A secret sharing". Prikladnaya Diskretnaya Matematika. Prilozhenie, 2017, no. 10, pp. 68–70. (in Russian)
4. *Rathgeb C. and Uhl C.* A survey on biometric cryptosystems and cancelable biometrics. EURASIP J. Inform. Security, 2011, vol. 2011:3. `https://doi.org/10.1186/1687-417X-2011-3`
5. *Diffie W., Van Oorschot P. C., and Wiener M. J.* Authentication and authenticated key exchanges. Designs, Codes and Cryptography, 1992, vol. 2, iss. 2, pp. 107–125.
6. *Daemen J. and Rijmen V.* The Design of Rijndael: AES — the Advanced Encryption Standard. Springer Verlag, 2002.
7. *Nakamoto S.* Bitcoin: a peer-to-peer electronic cash system. 2009. Available at `https://bitcoin.org/bitcoin.pdf`
8. *Carlet C.* On the algebraic immunities and higher order nonlinearities of vectorial Boolean Functions. Proc. NATO Advanced Research Workshop ACPTECC, Veliko Tarnovo, Bulgaria, October 6–9, 2008, Amsterdam, IOS Press, 2009, pp. 104–116.