УДК 519.7 DOI 10.17223/20710410/X/1

MATHEMATICAL PROBLEMS AND SOLUTIONS OF THE ELEVENTH INTERNATIONAL OLYMPIAD IN CRYPTOGRAPHY NSUCRYPTO[1]

N. N. Tokareva[1], I. S. Khilchuk[1], S. V. Bezzateev[2], O. S. Zaikin[3,1], E. A. Ishchukova[4,1], N. A. Kolomeec[1], A. V. Kutsenko[1], E. S. Malygina[5,1], S. P. Panasenko[6], A. A. Semenov[3]

[1] *Novosibirsk State University, Novosibirsk, Russia*
[2] *State University of Aerospace Instrumentation, Saint Petersburg, Russia*
[3] *Matrosov Institute for System Dynamics and Control Theory, Irkutsk, Russia*
[4] *Southern Federal University, Rostov-on-Don, Russia*
[5] *Immanuel Kant Baltic Federal University, Kaliningrad, Russia*
[6] *Aktiv company, Moscow, Russia*

E-mail: crypto1127@mail.ru

International Olympiad in Cryptography Non-Stop University CRYPTO (NSUCRYPTO) is a great annual event in the world of cryptographic research. The Olympiad offers mathematical problems for university and school students and, moreover, for specialists in cryptography and information security. Solutions of many real cryptographic tasks that appear in applications are based on math achievements. That is why we invite professionals and students to apply their knowledge to solving olimpiad tasks of NSUCRYPTO. In this paper we propose problems of NSUCRYPTO-2024 and consider their solutions. There were 14 problems related to distinct methods of cryptography, to hash functions and historical ciphers, cipher design and analysis, post-quantum schemes and signatures, SAT-solvers and steganography.

Keywords: *cryptography, ciphers, protocols, number theory, S-boxes, quantum circuits, matrices, hash functions, postquantum cryptosystems, signatures, Olympiad, NSUCRYPTO.*

МАТЕМАТИЧЕСКИЕ ПРОБЛЕМЫ И РЕШЕНИЯ ОДИННАДЦАТОЙ МЕЖДУНАРОДНОЙ ОЛИМПИАДЫ ПО КРИПТОГРАФИИ NSUCRYPTO

Н. Н. Токарева[1], И. С. Хильчук[1], С. В. Беззатеев[2], О. С. Заикин[3,1], Е. А. Ищукова[4,1], Н. А. Коломеец[1], А. В. Куценко[1], Е. С. Малыгина[5,1], С. П. Панасенко[6], А. А. Семенов[3]

[1] *Новосибирский государственный университет, г. Новосибирск, Россия*
[2] *Санкт-Петербургский государственный университет аэрокосмического приборостроения, г. Санкт-Петербург, Россия*
[3] *Институт динамики систем и теории управления имени В.М. Матросова СО РАН, г. Иркутск, Россия*
[4] *Южный федеральный университет, г. Ростов-на-Дону, Россия*

[1] The work of the first, second, fourth, fifth and sixth authors was supported by the Mathematical Center in Akademgorodok under the agreement No. 075-15-2025-349 with the Ministry of Science and Higher Education of the Russian Federation. The work is also supported by National Technology Center for Digital Cryptography, Novosibirsk State University, Southern Federal University, Kryptonite, Aktiv company and Kovalevskaya North-West Center of Mathematical Research under the agreement No. 075-02-2023-934 with the Ministry of Science and Higher Education of the Russian Federation.

[5]*Балтийский федеральный университет имени И. Канта, г. Калининград, Россия*
[6]*Компания «Актив», г. Москва, Россия*

Международная олимпиада по криптографии Non-Stop University CRYPTO (NSUCRYPTO) — ежегодное крупное мероприятие в области криптографии. На олимпиаде предлагаются математические задачи для студентов университетов и школ, а также для профессионалов в области криптографии и информационной безопасности. Решения многих реальных криптографических задач, которые появляются в приложениях, основаны на применении самых разных достижений математики. Именно поэтому мы приглашаем профессионалов и студентов к участию — здесь можно увидеть, как ваши знания находят реальные приложения. В данной статье мы рассматриваем задачи NSUCRYPTO-2024 и их решения. Приводятся 14 задач по различным методам криптографии, хэш-функциям и историческим шифрам, построению и анализу шифров, постквантовым схемам и подписям, SAT-решателям и стеганографии.

Ключевые слова: *криптография, шифры, протоколы, теория чисел, S-блоки, квантовые схемы, матрицы, хэш-функции, постквантовые криптосистемы, подписи, олимпиада, NSUCRYPTO.*

1. Introduction

Non-Stop University CRYPTO (**NSUCRYPTO**) is the unique international competition for professionals, school and university students, providing various problems on theoretical and practical aspects of modern cryptography, see [15]. The main goal of the olympiad is to draw attention of young researchers not only to competetive fascinating tasks, but also to sophisticated and tough scientific problems at the intersection of mathematics and cryptography. That is why each year there are several open problems in the list of tasks that require rigorous studying and deserve a separate publication in case of being solved. Since NSUCRYPTO holds via the Internet, everybody can easily take part in it. Rules of the Olympiad, the archive of problems, solutions and many more can be found at the official website [15].

The first Olympiad was held in 2014, since then several thousands students and specialists from more than 70 countries took part in it. The Program committee includes 31 members from cryptographic groups all over the world. Main organizers and partners are Cryptographic center (Novosibirsk), National Technology Center for Digital Cryptography, Novosibirsk State University, Kryptonite, Aktiv company, KU Leuven, Southern Federal University, Kovalevskaya North-West Center of Mathematical Research, Belarusian State University, Tomsk State University and Nsucrypto-lab.

In 2024 there were 1255 participants from 47 countries. This year 18 participants in the first round and 45 teams in the second round from 15 countries became the winners (see the list [16]). This year we proposed 14 problems to participants and 2 of them were entirely open or included some open questions.

Following the results of each Olympiad we also publish scientific articles with detailed solutions and some analysis of the solutions proposed by the participants, including advances on unsolved ones, see [1, 2, 6–11, 13, 14].

2. An overview of open problems

One of the main characteristic of the Olympiad is that unsolved scientific problems are proposed to the participants in addition to problems with known solutions. All 38 open problems that were offered since the first NSUCRYPTO can be found here [15]. Some of these problems are of great interest to cryptographers and mathematicians for many years. These are such problems as “APN permutation” (2014), “Big Fermat numbers” (2016), “Boolean hidden shift and quantum computings” (2017), “Disjunct Matrices” (2018), and others.

We are proud that some researchers continue to work on solutions of unsolved problems even after the Olympiad was over. For example, authors of [12] proposed a complete solution for problem “Orthogonal arrays” (2018). Partial solutions for another open problem, “A secret sharing”, (2014) were presented in [4], [5], and a recursive algorithm for finding the solution was proposed in [3], etc.

3. Problem structure of the Olympiad

There were 14 problems stated during the Olympiad, some of them were included in both rounds (Tables 1, 2). Section A of the first round consisted of six problems, while Section B of the first round consisted of 8 problems. The second round was composed of 11 problems; 2 of them included unsolved questions (awarded special prizes).

Table 1

Problems of the first round

N	Problem title	Max score
1	Cryptographic Fish	4
2	Decrypt with a hint	4
3	Alice’s house	4
4	Weak key schedule for DES	8
5	A simple hash function	6
6	RSA signature	4

Section A

N	Problem title	Max score
1	Cryptographic Fish	4
2	Decrypt with a hint	4
3	Alice’s house	4
4	A nonlinear generator	8
5	Unsecure SP-network	8
6	Weak key schedule for DES	8
7	RSA signature	4
8	Unknown function	6

Section B

Table 2

Problems of the second round

N	Problem title	Max score
1	RSA signature	4
2	AntCipher 2.0	6
3	Steganography and codes	8
4	Weak key schedule for DES	8
5	Reverse engineering	6
6	Open competition: NSUCRYPTO lightweight cipher	10, open problem
7	A nonlinear generator	8
8	Unsecure SP-network	8
9	Post-quantum signature	10, open problem
10	Unknown function	6
11	A simple hash function	6

4. Problems and their solutions

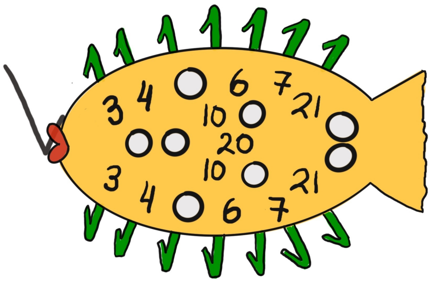
In this section, we formulate all the problems of 2024 year Olympiad and present their solutions, in some particular cases we also pay attention to solutions proposed by our participants.

4.1. Problem “Cryptographic Fish”

Formulation

There are several ciphers in modern world named after «fish». There are ciphers BlowFish, TwoFish, ThreeFish and even cryptocurrency CryptoFish.

But Alice has found a new fish-like crypto object, here it is.



Please, could you find a key from it? It is a sum of the missed circled numbers.

Solution

The numbers on the side of the fish, as well as 1's along the spine and belly (and even a hook in the form of «1») hint on famous Pascal's triangle.

$$\begin{array}{c}
1 \\
1 \quad 1 \\
1 \quad 2 \quad 1 \\
1 \quad 3 \quad 3 \quad 1 \\
1 \quad 4 \quad 6 \quad 4 \quad 1 \\
1 \quad 5 \quad 10 \quad 10 \quad 5 \quad 1 \\
1 \quad 6 \quad 15 \quad 20 \quad 15 \quad 6 \quad 1 \\
1 \quad 7 \quad 21 \quad 35 \quad 35 \quad 21 \quad 7 \quad 1
\end{array}$$

Рис. 1. The beginning of Pascal's triangle

So, the missing numbers are 2, 6, 5, 5, 15, 15, 35, 35, and their sum — number 118 — is the answer to this task.

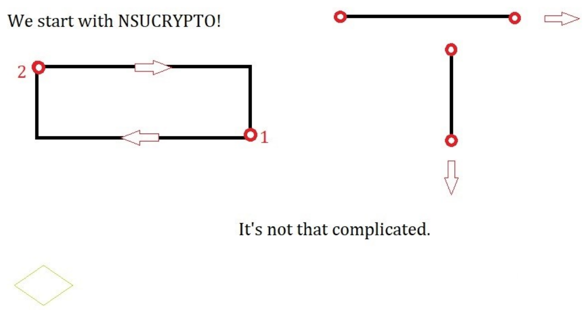
4.2. Problem “Decrypt with a hint”

Formulation

Alice has found an interesting message from Bob. Here it is:

LFUZGAEPPXLSOANA

And here is a hint how the text was encrypted. Could you decrypt the text?



We start with NSUCRYPTO!

It's not that complicated.

Solution

Diagrams and part of the portrait of lord Playfair indicate Playfair cipher, and to fill the initial 5×5 grid we need only a key. The hint «we start with NSUCRYPTO!» provides us with such key.

So, the grid will look like this:

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

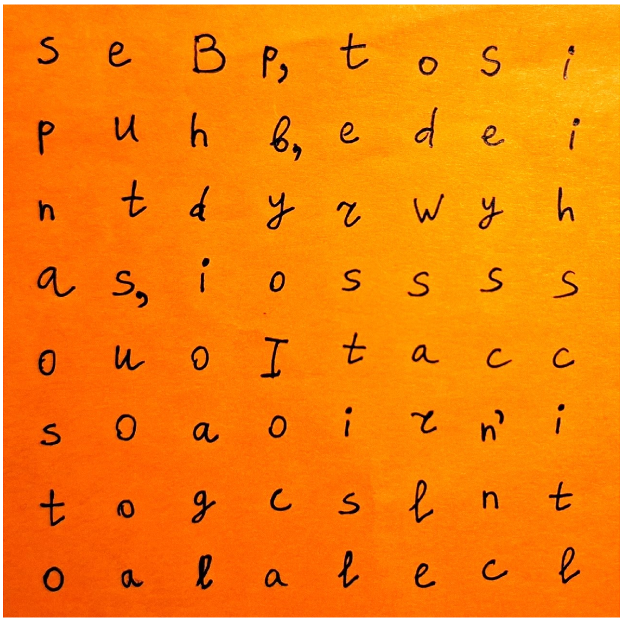
Рис. 2. Using the key for Playfair cipher

Then, applying the rules of the cipher, we get the plaintext FORWARDTOVICTORY.

4.3. Problem “Alice’s house”

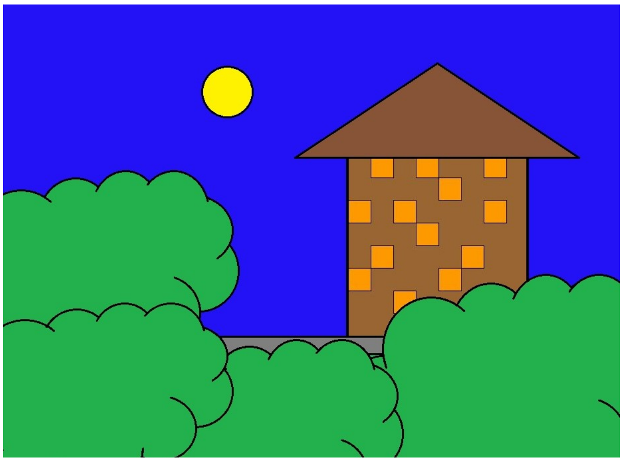
Formulation

When Bob was returning from Alice’s house, he found an encrypted letter on the road:



s	e	B	p,	t	o	s	i
p	u	h	b,	e	d	e	i
n	t	d	y	r	w	y	h
a	s,	i	o	s	s	s	s
o	u	o	I	t	a	c	c
s	o	a	o	i	r	n'	i
t	o	g	c	s	l	n	t
o	a	l	a	t	e	c	l

He guessed that it was a message from Alice. Bob looked around and immediately read it. What was the letter about?



Solution

The symbols in the letter are written in 8×8 matrix, and we can see that the windows on the side of the house are located in 8×8 grid, some of them are lit up, and some of them are dark and do not stand out. Moreover, a part of the house is hidden by a shrubs, so we cannot see whether the windows are dark or lit up in that part.

Turning grille cipher is what was used to encrypt the message: windows with light indicate the letter which is included in message. Applying the grille once we may recover the text "ep, send you a sig...", part of which is hidden behind shrubs. "ep," implies that it is not the beginning of the original message. Indeed, the grille has four possible positions: 1) original one yeilds the text "ep,sendyouasig", 2) turning 90 degrees to the right gives us "phrs,soIan'tsle" with some of the letters still missing, 3) turning 180 degrees provides us with "dyhistoricalc", 4) finally, turning 90 degrees to the left we obtain "Bob,it issocolt".

To recover hidden letters one must remember that grille must not indicate one letter more than one time upon turning. It leaves one single possibility:

	X		X			X	
				X			
X		X				X	
			X				
	X				X		
X				X			
		X				X	
			X				X

Таблица 1

Full turning grille

And the final message is «Bob, it is so cool to study historical ciphers, so I can't sleep, send you a signal».

4.4. Problem "Weak key schedule for DES"

Formulation

Alice is a novice cryptographer. She figured out how the DES encryption algorithm works and decided to implement it in order to exchange secret messages with Bob. She used the simplest ECB mode. But in her implementation, Alice made a mistake: inside the function F in addition of data with a round secret subkey, she forgot to change the index. So, in her implementation, in each round, the data is added modulo 2 with the first round key. Carol really wants to know what Alice and Bob are exchanging messages about. She even managed to get hold of a couple of files once. The `Book.txt` file [17] contains an open message, and the `Book_Cipher.txt` file [18] contains the corresponding encrypted text. Help Carol to find the secret encryption key and read the message she intercepted (the message is in hexadecimal format):

```
86991641D28259604412D6BA88A5C0A6471CA722
2C52482BF2D0E841D4343DFB877DC8E0147F3D5F
20FC18FF28CB5C4DA8A0F4694861AB5E98F37ADB
C2D69B35779D9001BB4B648518FE6EBC00B2AB10
```

Some explanations. Description of DES algorithm can be found in the web, see for inst. `https://csrc.nist.gov/files/pubs/fips/46/final/docs/nbs.fips.46.pdf` Consider an example. We are talking about the correct implementation of DES, where all 16 round subkeys are used correctly. For example, if we take the plaintext `8787878787878787`, and encrypt it with the DES key `0E329232EA6D0D73`, we end with the ciphertext `0000000000000000`. If the ciphertext is decrypted with the same secret DES key `0E329232EA6D0D73`, the result is the original plaintext `8787878787878787`. In the `Book.txt` file, each character corresponds to one byte of information according to the ASCII table. Since the DES algorithm processes 64 bits at a time, the first 8 characters of «Three Ri» will be used as the input message, which corresponds to the hexadecimal sequence `5468726565205269`. It should be noted that moving the carriage return to a new line in the file also takes two bytes `0D0A`.

Solution

Since Alice made a mistake, and in her implementation each round uses the same round key, then in this case the slide attack can be applied. To do this, it is necessary to compare two encryption processes as shown in Fig. 1 and try to find such pairs of texts that satisfy the given conditions.

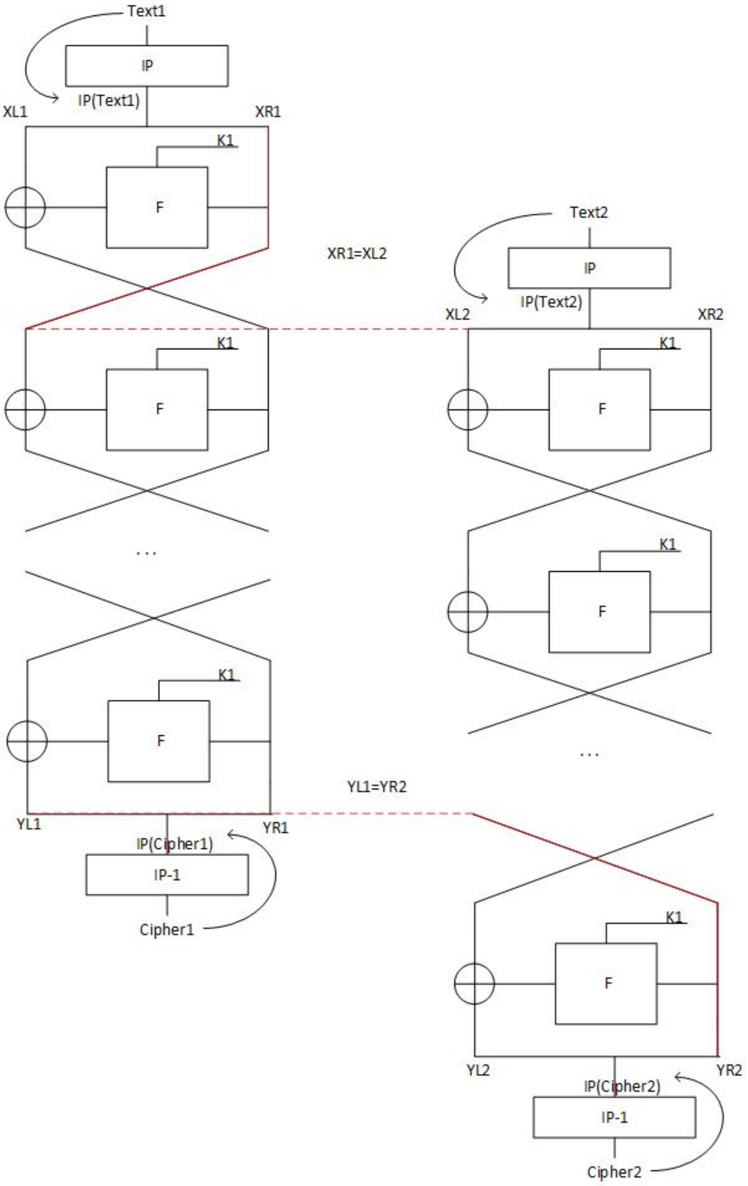
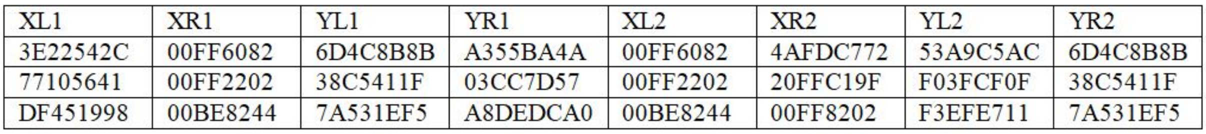


Рис. 3. Scheme

As a result of the data analysis of the plaintexts from the `Book.txt` file and the corresponding ciphertexts from the `Book_Cipher.txt` file, we will find at least three pairs of such texts. We will reflect them in Table 1. The results are presented for the values XL, XR, YL, YR. That is, after the IP permutations performed for the values Text1, Text2, Cipher1, Cipher2.



XL1	XR1	YL1	YR1	XL2	XR2	YL2	YR2
3E22542C	00FF6082	6D4C8B8B	A355BA4A	00FF6082	4AFDC772	53A9C5AC	6D4C8B8B
77105641	00FF2202	38C5411F	03CC7D57	00FF2202	20FFC19F	F03FCF0F	38C5411F
DF451998	00BE8244	7A531EF5	A8DEDCA0	00BE8244	00FF8202	F3EFE711	7A531EF5

Рис. 4

Having slide text pairs lets one to make a guess at the inputs and outputs of the first and the last rounds (Fig. 2). Analysing them and leaving only the keys one meets every time, it is possible to recover a secret key. If there are several candidates for secret key, one can find the right one by brute force search, since we have enough pairs of plaintext and ciphertext.

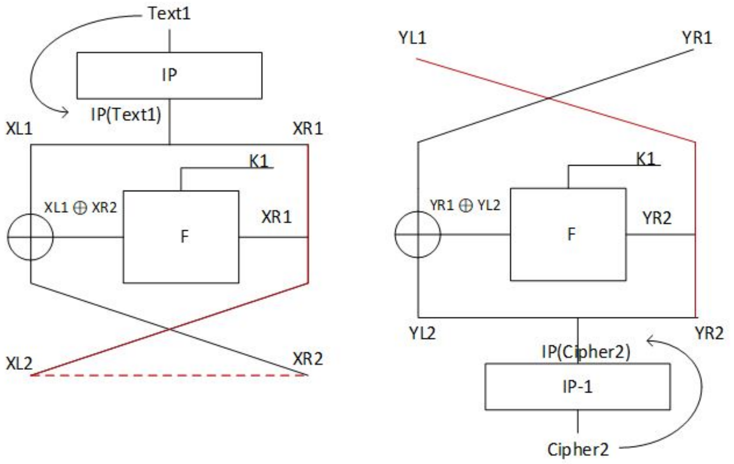


Рис. 5. Scheme

As a result we get a key $K = 0C74FA6A642A$. With it we can easily decipher the message: «`It is better to be in chains with friends, than to be in a garden with strangers.`»

4.5. Problem "A simple hash function"

Formulation

Carol invented a new hash function. The key $k = (k_1, \ldots, k_6)$ for this hash function is a binary vector of length 6. The input for the hash function is a sequence of digits. It should be divided into blocks of length 6. If the length of the sequence is not a multiple of 6 then it can be completed with 1, 2, 3, and so on up to the necessary length. For example, if an input is 7256 then it should be changed to 725612 first.

Then each block of input, say $(p_1, \ldots, p_6)$ should be transformed into the number by the rule: $(-1)^{k_1} \cdot p_1 + \ldots + (-1)^{k_6} \cdot p_6$. Here $(-1)^0 = 1$ and $(-1)^1 = -1$. Results of such calculations for blocks, say $n_1, n_2, n_3, n_4, \ldots$, then form a resulting hash value as $H = n_1 - n_2 + n_3 - n_4 + \ldots$.

For example, if the key is (001101) then hash for the sequence 134875 512293 is $H = (1 + 3 - 4 - 8 + 7 - 5) - (5 + 1 - 2 - 2 + 9 - 3) = -6 + 8 = 2$.

Carol applied her hash function in the system for logging at the bank website. Every user enters his password, say P, then system counts hash value $H(P, K)$ and if it coincides with the hash value from the data base, then user is logged.

But after some time Carol realized that the system is not secure. Malefactors can construct collisions and enter the system illegally. How do they do it? Propose a simplest algorithm how to get a collision of the first order for any known input sequence P if the key K is unknown.

By the way, find the shortest collision for the sequence from the example, $P = 134875\,512293$. Since K is unknown, the hash value $H(P, K)$ is still unknown to you.

Solution

The trivial collison can be built by concatenation of the text with zero symbol but the participants also proposed more complicated ways of obtaining short collisions for the cases of different block structures and messages as well as messages with multiple blocks. In the best solution participants described different ways of obtaining collisions based on the proposed equivalence relation and considered some classes. The shortest collision for the given message was also found by many participants.

4.6. Problem “RSA signature”

Formulation

We want to sign the message M using the RSA-signature. As usually, let $N = p \cdot q$ be the RSA-modulus, where p and q are two big primes. Let e be the RSA-public exponent and d be the RSA-secret exponent satisfying that $e \cdot d = 1 \mod (p-1)(q-1)$. The desired signature is given by

$$S = M^d \mod N.$$

Suppose that the attacker knows the value

$$M_p := M^{d_p} \mod p,$$

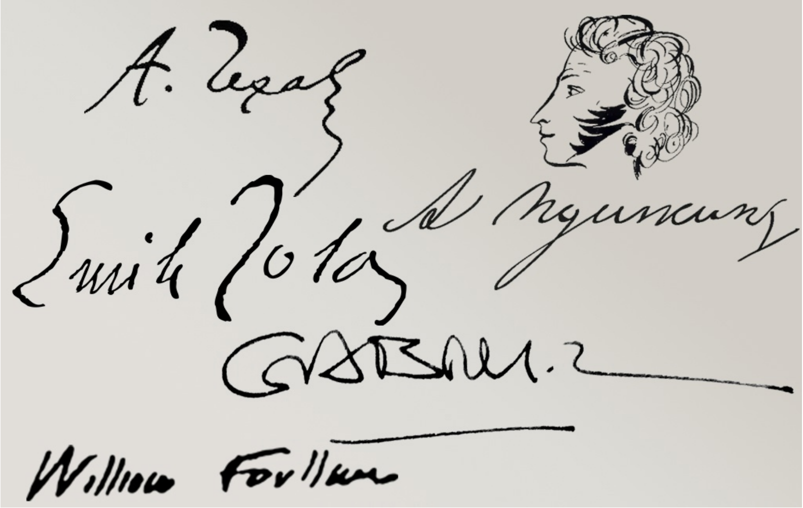
but he doesn’t know the value

$$M_q := M^{d_q} \mod q,$$

where

$$d_p := d \mod (p-1), \quad d_q := d \mod (q-1).$$

If the attacker knows the modulus N (but not p and q), the public exponent e (but not d), and the original message M, what secret signature parameters can he calculate? Justify the answer.



Solution

We will use the Chinese remainder theorem. Then

$$S = M^d \mod N = (\alpha \cdot M_p + \beta \cdot M_q) \mod N,$$

where

$$\begin{cases} \alpha \equiv 1 \mod p, \\ \alpha \equiv 0 \mod q, \end{cases} \qquad \begin{cases} \beta \equiv 0 \mod p, \\ \beta \equiv 1 \mod q. \end{cases}$$

Obviously, if the attacker knows M_p and doesn’t know M_q, then the desired signature will be incorrect, because

$$S = M^d \mod N \neq (\alpha \cdot M_p + \beta \cdot \tilde{M}_q) \mod N = \tilde{S}$$

and

$$S^e = M \neq \tilde{S}^e.$$

But

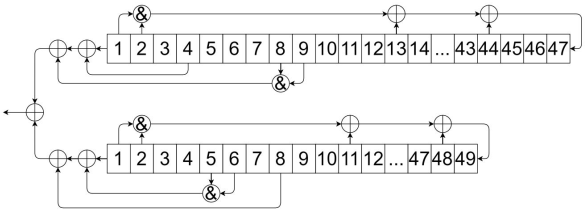
$$\tilde{S} \mod p = M_p = (M \mod p)^{d \mod (p-1)},$$

consequently, $\tilde{S}^e = M \mod p$, so $p|(\tilde{S} - M)$. Therefore, the attacker only needs to calculate $\mathrm{GCD}(N, \tilde{S}^e - M)$ to get p.

4.7. Problem "A nonlinear generator"

Formulation

Alice invented a keystream generator presented at the figure:



It consists of two shift registers of lengths 47 and 49 with non-linear feedback functions. The contents of the cells of a specific register at any time moment $t = 1, 2, \ldots$ form the state number t of this register. At time t, each register first generates keystream bit number t and then transitions to the next state number $t+1$. The states of the registers at moment t are denoted as

$$A(t) = (a_1(t), \ldots, a_{47}(t)) \text{ and } B(t) = (b_1(t), \ldots, b_{49}(t)) \text{ respectively.}$$

Both registers are shifted synchronously. For instance,

$$A(t+1) = (a_2(t), \ldots, a_{47}(t), (a_1(t)\&a_2(t)) \oplus a_{13}(t) \oplus a_{44}(t)).$$

The keystream Γ of length 8192, created by this generator, is given and can be found in `keystream.txt` [19]. Also, the states $A(8192)$ and $B(8192)$ are known:

$A(8192) = (\mathtt{00101001110001001110111001100001010100000101110})$,
$B(8192) = (\mathtt{0000010000101001000011000001010111001110000100101})$.

Could you find the initial states $A(1)$ and $B(1)$ of these registers?

Solution

The answer is [0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0] and also [1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0].

4.8. Problem "Unsecure SP-network"

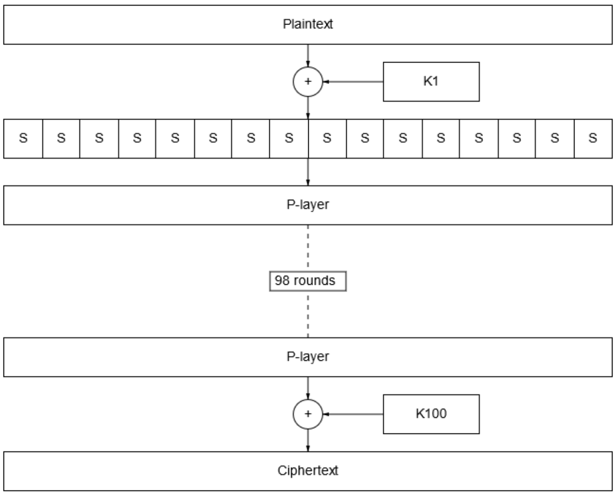
Formulation

Bob heard about the SP-network, and decided to make his own cipher on this base, so that Carol would not be able to read his correspondence with Alice. The block size was chosen 32 bits. He made S-boxes of size 2×2 and P-layer used the part of secret key. Recall that P is an arbitrary linear transformation $P : \mathbb{F}_2^{32} \to \mathbb{F}_2^{32}$, i. e. $P(x \oplus y) = P(x) \oplus P(y)$ for any $x, y \in \mathbb{F}_2^{32}$.

In addition to this, there is secret key $K \in \mathbb{F}_2^{128}$. Using it Bob determined

$$K^i = (K_{32(i \mod 4)+1}, \ldots, K_{32(i \mod 4)+32})$$

of length 32 for all $i \in \{1, \ldots, 100\}$. Here is the scheme of the cipher:



The i-th round of the cipher is as follows:

$$r_i(x) = P(S(x_1 \oplus K_1^i, x_2 \oplus K_2^i), S(x_3 \oplus K_3^i, x_4 \oplus K_4^i), \ldots, S(x_{31} \oplus K_{31}^i, x_{32} \oplus K_{32}^i)), x \in \mathbb{F}_2^{32}$$

The encrypted message (ciphertext) is

$$c = K^{100} \oplus r_{99}(r_{98}(\ldots r_1(m))),$$

where m is the initial message (plaintext), $m = (m_1, \ldots, m_{32})$, where $m_i \in \mathbb{F}_2$.

However, he soon discovered that Carol could read his correspondence with Alice without any problems if she had known some 100 random pairs of plaintext and ciphertext. How is this possible?

Solution

Let us denote the encryption mapping by E_K, where K is the secret key. The key observation here is that the function E_K is affine.

Indeed, any invertible mapping of the form $\mathbb{F}_2^2 \to \mathbb{F}_2^2$ is affine (it cannot be quadratic since the maximum degree 2 for such functions implies that some of its coordinate functions is of odd Hamming weight, i.e. not balanced). Thus, all S-boxes of E_K are affine. Since they are the only nonlinear components of E_K, hence E_K is affine as well.

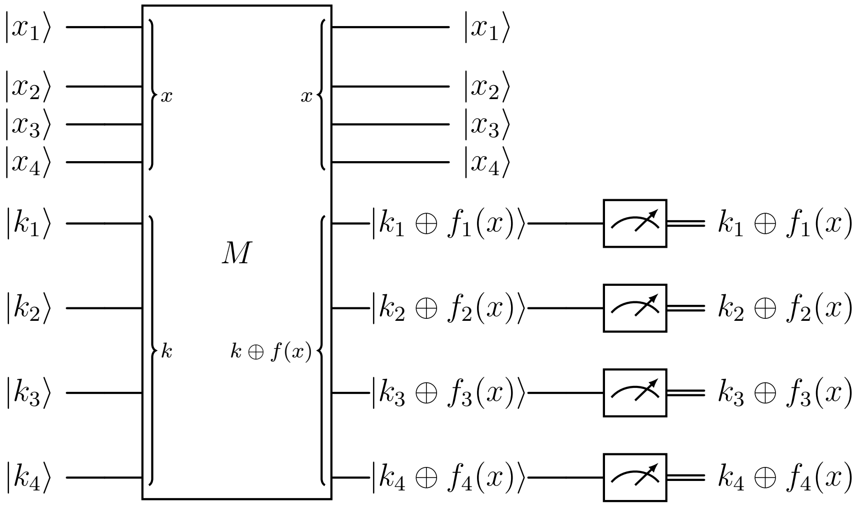
Overall, E_K maps $\mathbb{F}_2^{32}$ to itself, it is affine and invertible. Thus, we need to know only 32 linearly independent pairs plaintext/ciphertext (plaintexts linearly independent if and only if ciphertexts are linearly independent since E_K is affine and invertible) plus any additional pair to restore both E_K and E_K^{-1} without knowledge of K. Since Carol knows 100 random pairs, there are 32 linearly independent ones among them with probability very close to 1.

Surprisingly, there were only two completely correct solutions in the first round (Alexey Chilikov and Vladimir Schavelev). Arseny Lebedev, Irina Slonkina and Nilabha Saha gave almost correct solutions. In the second round, many teams proposed completely corrects solutions.

4.9. Problem “Unknown function”

Formulation

Bob works in a field of quantum mechanics, he invented a quantum machine M that encrypts 4-bit words by using 4-bit secret key $k = (k_1, k_2, k_3, k_4)$ according to the following quantum circuit:



This machine operates with 4-bit plaintext $x = (x_1, x_2, x_3, x_4)$ that is initially encoded to the corresponding 4-qubit «plainstate» $|x_1, x_2, x_3, x_4\rangle$ that is both with the «keystate» $|k_1, k_2, k_3, k_4\rangle$ is operated as

$$|x\rangle\,|k\rangle \xrightarrow{M} |x\rangle\,|k \oplus f(x)\rangle\,,$$

where $f(x) = (f_1(x), f_2(x), f_3(x), f_4(x))$ is an invertible vectorial Boolean function in 4 variables. The «cipherstate» $|k_1 \oplus f_1(x), k_2 \oplus f_2(x), k_3 \oplus f_3(x), k_4 \oplus f_4(x)\rangle$ is further measured and the ciphertext is obtained.

The problem is could Alice find the secret key k if the function f is unknown to her? Assume she has oracle access to the quantum machine with the fixed key k and she provided additional information $f(0,0,0,0) \oplus f(1,1,1,1) \oplus f(1,0,0,0) = c \in \mathbb{F}_2^4$ with known c.

Remark. Recall some key points of quantum circuits. A qubit is a two-level quantum mechanical system whose state $|\psi\rangle$ is the superposition of basis quantum states $|0\rangle$ and $|1\rangle$. The superposition is written as $|\psi\rangle = \alpha_0\,|0\rangle + \alpha_1\,|1\rangle$, where α_0 and α_1 are complex numbers, called amplitudes, that satisfy $|\alpha_0|^2 + |\alpha_1|^2 = 1$. The amplitudes α_0 and α_1 have the following physical meaning: after the measurement of a qubit with the state $|\psi\rangle$ in a basis $\{|0\rangle\,, |1\rangle\}$, it will be found in the state $|0\rangle$ with probability $|\alpha_0|^2$ and in the state $|1\rangle$ with probability $|\alpha_1|^2$. After the computation, the state of the qubit is measured and the result is observed.

Solution

The idea of solution is to encrypt the «plainstates» $|0000\rangle$, $|1111\rangle$, $|1000\rangle$ and consider the sum modulo 2 of the vectors from the corresponding «cipherstates». It yields

$$|0000\rangle\,|k\rangle \xrightarrow{M} |0000\rangle\,|k \oplus f(0000)\rangle\,,$$
$$|1111\rangle\,|k\rangle \xrightarrow{M} |1111\rangle\,|k \oplus f(1111)\rangle\,,$$
$$|1000\rangle\,|k\rangle \xrightarrow{M} |1000\rangle\,|k \oplus f(1000)\rangle\,.$$

For every encryption one can measure the second group of 4 qubits and sum up modulo 2 all results, that yields

$$k \oplus f(0000) \oplus k \oplus f(1111) \oplus k \oplus f(1000) = k \oplus f(0000) \oplus f(1111) \oplus f(1000).$$

The given condition $f(0000) \oplus f(1111) \oplus f(1000) = c$ with known vector $c \in \mathbb{F}_2^4$ allows to recover the unknown key.

It is clear that for the case when length of the input message is divisible by 6 the trivial collision can be built by concatenation of the message with symbol 0. If length is not divisible by 6, one can complete it in by the described procedure and consider as a new message. Many participants also proposed more complicated ways of obtaining short collisions for messages with multiple blocks and different block structures. In the best solution different ways of obtaining collisions based on the proposed equivalence relation were described and deep analysis of the classes within this equivalence was provided.

The shortest collision for the given message was also found by many participants, it has length 7 and, for example, can be obtained from the following messages: 034927(4), 134927(5), 234927(6), 334927(7), 434927(8), 534927(9).

4.10. Problem “AntCipher 2.0”

Formulation

Sam studies microelectronics, while his hobbies are biology and cryptography. He united all these areas in a research project aimed at constructing a tiny GPS tracker for an ant to monitor its movements. When coordinates are determined, they are encrypted and transmitted to a Sam’s computer, where they are automatically decrypted. Sam developed a symmetric cipher AntCipher for this purpose, but it was quite weak. That is why Sam developed a new symmetric stream cipher called AntCipher 2.0.

Once a minute, the tracker determines its GPS coordinates using satellites. Then the latitude as an IEEE 754 single-precision floating-point value is converted into a 32-bit binary sequence, while the same is done with the longitude. These two sequences are concatenated (latitude || longitude) to form a 64-bit plaintext. The plaintext is bitwise XORed with a keystream produced by the cipher thus forming a 64-bit ciphertext which is transmitted to the computer.

The cipher works as follows. At the initialization stage, a 64-bit secret key is written to a 64-bit register R. In iteration number $i, i \geqslant 1$, a 64-bit sequence (keystream) K_i is produced taking a value of R as an input. The keystream is also used to update the register: at the end of the iteration, K_i is written to R. Consider the following CNF C, where CNF is a conjunction of disjunctions of literals, yet literal is a Boolean variable or its negation:

$C = (x_1 \vee x_2 \vee \neg x_5) \wedge (\neg x_1 \vee \neg x_2 \vee x_5) \wedge (x_1 \vee x_3 \vee \neg x_5) \wedge (\neg x_1 \vee \neg x_3 \vee x_5) \wedge (x_2 \vee x_3 \vee \neg x_5) \wedge (\neg x_2 \vee \neg x_3 \vee x_5) \wedge (x_1 \vee x_2 \vee \neg x_6) \wedge (\neg x_1 \vee \neg x_2 \vee x_6) \wedge (x_1 \vee x_4 \vee \neg x_6) \wedge (\neg x_1 \vee \neg x_4 \vee x_6) \wedge (x_2 \vee x_4 \vee \neg x_6) \wedge (\neg x_2 \vee \neg x_4 \vee x_6) \wedge (x_1 \vee x_3 \vee \neg x_7) \wedge (\neg x_1 \vee \neg x_3 \vee x_7) \wedge (x_1 \vee x_4 \vee \neg x_7) \wedge (\neg x_1 \vee \neg x_4 \vee x_7) \wedge (x_3 \vee x_4 \vee \neg x_7) \wedge (\neg x_3 \vee \neg x_4 \vee x_7) \wedge (x_2 \vee x_3 \vee \neg x_8) \wedge (\neg x_2 \vee \neg x_3 \vee x_8) \wedge (x_2 \vee x_4 \vee \neg x_8) \wedge (\neg x_2 \vee \neg x_4 \vee x_8) \wedge (x_3 \vee x_4 \vee \neg x_8) \wedge (\neg x_3 \vee \neg x_4 \vee x_8)$.

The equation $C = 1$ represents a nonlinear function F_C that takes a 4-bit input x_1, x_2, x_3, x_4 and produces a 4-bit output x_5, x_6, x_7, x_8. In the i-th iteration of the cipher, a 64-bit value of R is divided into 16 4-bit sequences, which are given to F_C as inputs. Then 16 4-bit outputs are produced and concatenated thus forming a 64-bit K_i that is written to R and is used as a keystream.

On the computer, the cipher is initialized by the same secret key, so the same keystream is produced as on the tracker. When a 64-bit ciphertext is transmitted from the tracker, the corresponding keystream is produced and bitwise XORed with the ciphertext thus obtaining the plaintext. The first 1 704 ciphertexts were transmitted with no problem and the coordinates were automatically decrypted. Then a hard disk drive failure happened on the computer and as a result the secret key, as well as almost all 64-bit ciphertexts and keystreams were lost. Sam could recover only the last 1704-th ciphertext: 1001 1000 0011 1101 0110 0011 1101 0101 1011 0011 1011 0111 0000 0000 1000 0011. Also, the keystreams generated in iterations 1702 and 1703 were partially recovered ('X' stands for an unknown bit value):

- K_{1702} = 0101 1001 1111 0011 00X1 X111 1X00 00X0 111X X000 XXXX XXXX XXXX XXXX XXXX XXXX;
- K_{1703} = XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX X111 000X X010 01X1 0X10 0101 0000 1111.

Please, help Sam to find the plaintext in iteration 1704 to find the ant.

Solution

The problem can be solved in both analytical and numerical ways.

Analytical solution

In the CNF C, the majority function is represented four times: $x_5 = maj(x_1, x_2, x_3)$; $x_6 = maj(x_1, x_2, x_4)$; $x_7 = maj(x_1, x_3, x_4)$; $x_8 = maj(x_2, x_3, x_4)$. It is clear that having 16 possible input's values, there are only 8 possible outputs of F_C: 0000, 0011, 1100, 0101, 1010, 0110, 1001, 1111. Therefore, there is even number of 0s in all 4-bit outputs. Taking this feature into account, 6 previously unknown bits in K_{1702} are determined: 0101 1001 1111 0011 00**11** **1**111 1**1**00 00**0**0 111**1** **0**000 XXXX XXXX XXXX XXXX XXXX XXXX. The same action for K_{1703} determines 5 more bits: XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX **1**111 000**0** **1**010 01**0**1 0**1**10 0101 0000 1111.

The second feature of F_C in the context of the cipher is that starting from the second iteration, there are only 8 possible inputs, and 6 of them are mapped onto themselves. The only special words are 0110 and 1001: they are mapped onto 1001 and 0110, respectively. Thus keystreams in all odd iterations are equal to each other, while keystreams in all even iterations are equal to each other as well.

There is one word 0110 in the last 24 bits K_{1703} and no words 1001, so these 24 bits (along with changing 0110 to 1001) can be added to the first 40 bits of K_{1702} because of the second feature. Thus K_{1702} is 0101 1001 1111 0011 0011 1111 1100 0000 1111 0000 1010 0101 1001 0101 0000 1111. Recall that $K_{1704} = K_{1702}$. By XORing K_{1704} with the ciphertext, the plain text encrypted in iteration 1704 is obtained: 1100 0001 1100 1110 0101

1100 0001 0101 0100 0011 0001 0010 1001 0101 1000 1100. To verify the correctness, divide the plaintext into two 32-bit sequences:

1) 11000001110011100101110000010101;
2) 01000011000100101001010110001100.

Using an online converter from binary to float[2], numbers -25.794962 and 146.58417 are obtained, which correspond to the GPS coordinates of the Meat ant park in the Queensland state, Australia. The ant with the tracker lives in this very park, while Sam lives nearby.

Numerical solution

There are 8 Boolean variables and 24 clauses in the CNF C. To encode the first iteration of the cipher, $8 \times 16 = 128$ Boolean variables and $24 \times 16 = 384$ clauses are required. To add the second iteration, only 64 new Boolean variables and 384 clauses are required since the output of the first iteration is input of the second iteration. The same holds for all further iterations. It means that to encode 1704 iterations, $128 + 1703 \times 64 = 109\ 120$ Boolean variables and $1704 \times 384 = 654336$ clauses are required. This CNF should be written to a file in the DIMACS format. Then all known keystream bits in K_{1702} and K_{1703} should be added to the CNF as unit clauses. If a modern SAT solver (e.g. Kissat[3]) is run on the CNF, a satisfying assignment will be found quickly. From the assignment, the keystream in iteration 1704 can be easily extracted as a 64-bit sequence. The verification process is the same as that for the analytical solution

4.11. Problem "Steganography and codes"

Formulation

Sam and Betty use public channel for their private communication. They want that nobody knows about the fact of their dialog.

They agreed that Sam can send to Betty one of the following sixteen messages:

0 — «Everything is OK», 1 — «I miss you», 2 — «I miss you too much!»,
3 — «Call me, please», 4 — «Where are you?», 5 — «YES!», 6 — «NO!»,
7 — «I said NO!», 8 — «I don't know», 9 — «I'm working now»,
10 — «I'm walking now», 11 — «I'm not available now», 12 — «I will come soon»,
13 — «I'm studying cryptography and think that it is a very great thing!»,
14 — «Go to the NSUCRYPTO next year with me!»,
15 — «Bye, bye! See you tomorrow».

Sam takes any picture in RGB format, changes the first pixel of it in some way and publishes the modified picture on his web-cite. Betty downloads the picture, analyzes it and takes out the message for her.

What does the Sam do with the picture? He should change it in such a way that nobody can visually fix the changing.

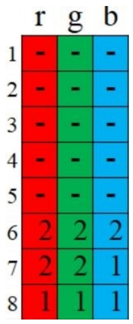
One pixel of a picture in format RGB is represented with 24 bits:
8 bits for brightness of red color $(r_1, \ldots, r_8)$,
8 bits for brightness of green color $(g_1, \ldots, g_8)$,
and 8 bits for brightness of blue color $(b_1, \ldots, b_8)$.

It is not possible for Sam to change bits r, g and b with numbers $1, \ldots, 5$ since it makes a changing to be visual. If Sam changes one of bits r_6, r_7, g_6, g_7 and b_6, let us say that it costs 2 coins, while changing of one bit between r_8, g_8, b_7 and b_8 costs 1 coin.

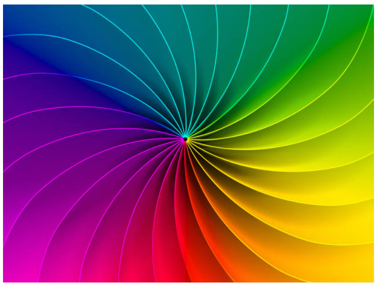
[2] For example, https://www.h-schmidt.net/FloatConverter/IEEE754.html

[3] https://github.com/arminbiere/kissat

Propose a method of coding a message (through given 16 types) in one pixel such it costs not more than 2 coins (in this case the changing of the picture is still not visual). Propose also the method for Betty how to extract secret messages. It is important that she has no access to the original picture.



	r	g	b
1	-	-	-
2	-	-	-
3	-	-	-
4	-	-	-
5	-	-	-
6	2	2	2
7	2	2	1
8	1	1	1



Solution

The solution solution uses error-correcting codes and syndrome decoding. Sam and Betty can use $(9, 5, 5)$-code, which is perfect with respect to weighted Hamming metric and has the minimal distance of 5 under this metric. 4 of the codeword positions have weight 1, and other 5 positions have weight 2. This code has $9 - 5 = 4$ check symbols and 16 different syndromes S_i, $i = 1, \dots, 16$. Each syndrome has corresponding unique error pattern, with weight in weighted Hamming metric not greater than 2. The 16 patterns are: 1) zero vector (no error occured), 2) $C_4^1 + C_4^2 = 10$ vectors with single or two 1's on the first 4 positions of weight 1, 3) 5 vectors which have single 1 on the 5 positions of weight 2.

We can consider the tuple of $(r_8, g_8, b_7, b_8, r_6, r_7, g_6, g_7, b_6)$ as vector a, the syndrome of which equals to S with respect to parity-check matrix H of our (9,5,5)-code.

$$a \times H^T = S$$

To transmit any of given 16 (notice, that we can represent the message as 4-bit vector I) messages we can change the initial vector a in such way, that the syndrome of this changed vector b is equal to I.

$$b \times H^T = I$$

Thus, we only need to determine, which vector e out of 16 possible patterns produce the syndrome $S \oplus I$. Since there are 16 different syndromes, each of which has unique corresponding error pattern with weight not greater than 2, such vector e can always be found.

Any possible vector e with weight not greater than 2 meets the condition of the problem, i.e. the cost of the changes will not exceed 2 coins.

There were 24 solutions that gained the full score between all 93 solutions sent to us.

An alternative approach was demonstrated by the professional team of Victoria Vysotskaya and Lev Vysotsky (Moscow, Russia) and the student team of Ivan Baksheev and Dmitriy Baryshev (Novosibirsk, Russia). Both teams reduced the given problem to proper graph coloring problem: each possible configuration of the bits in the pixel, which we consider as a vector $(r_8, g_8, b_7, b_8, r_6, r_7, g_6, g_7, b_6)$, is a vertex in graph G. An edge connects two vertices if one vector can be obtained from the other by inverting two «cheap» bits or one arbitrary bit, as per cost constraint. Since these transformations are symmetric, graph G is undirected, and each vertex has 15 neighbours, four of which are obtained from the initial vector by inverting one of «cheap» bits, and other eleven — by inverting only one arbitrary bit.

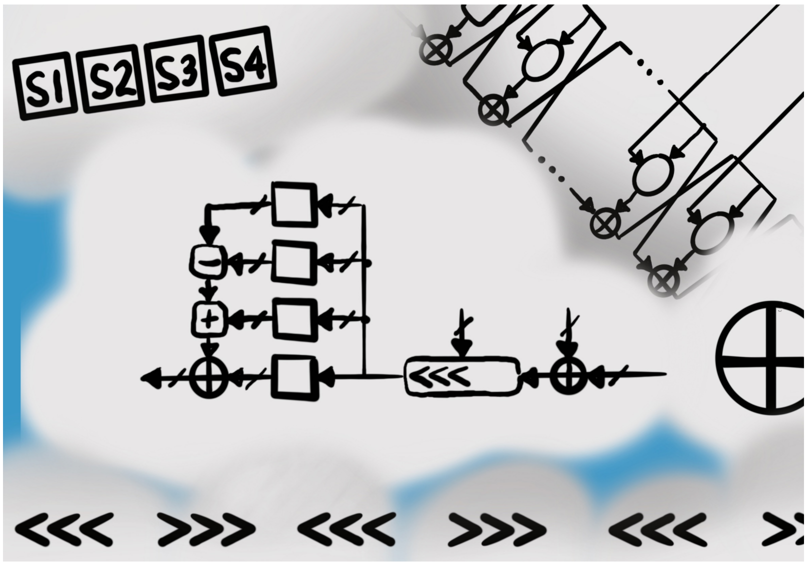
4.12. Problem “Reverse engineering”

Formulation

After reverse engineering of a realization for some unknown cryptographic algorithm, Bob obtained the following Boolean function:

$$f_{2n}(x_1, \ldots, x_{2n}) = \bigoplus_{i=1}^{n} x_i x_{i+n} \prod_{j=i+1}^{n} (x_j \oplus x_{j+n}).$$

He tried to understand what is a cryptographic sense of it. And soon a simple association ran through his head. What is this function?



Solution

We can represent f_{2n} as

$$f_{2n}(x, y) = \bigoplus_{i=1}^{n} x_i y_i \prod_{j=i+1}^{n} (x_j \oplus y_j),$$

where $x, y \in \mathbb{F}_2^n$. Next, it is not difficult to see that $f_{2n}(x, y)$ is the carry bit $c_n(x, y)$ if we sum x and y as integers:

$$\sum_{i=1}^{n} x_i 2^{i-1} + \sum_{i=1}^{n} y_i 2^{i-1},$$

i.e. $x + y = (x + y) \mod 2^n + f_{2n}(x, y)2^n$.

Indeed, the carry bit of the sum of $x'' = x'2^n + x$ and $y'' = y'2^n + y$, where $x, y \in \mathbb{F}_2^n$ and $x', y' \in \mathbb{F}_2$, is the majority function $z \in \mathbb{F}_2^3 \mapsto z_1 z_2 \oplus z_1 z_3 \oplus z_2 z_3$ of the inputs $c_n(x, y), x'$ and y', i.e.

$$c_{n+1}(x'', y'') = c_n(x, y)(x' \oplus y') \oplus x'y'.$$

It is clear that the same recurrence relation holds for f_{2n} and $f_2(x, y) = x_1 y_1 = c_1(x, y)$.

Finally, we show that this function can be a part of some ciphers. Indeed, since f_{2n} is a part of the addition modulo 2^{32}, it is used in ARX schemes and many other primitives such as CAST, MAGMA, MD5, etc.

In the second round, 15 teams proposed completely correct solutions. Some participants presented another interpretation of the function f_{2n} but they missed the cryptographic context of the problem.

4.13. Problem "Open competition: NSUCRYPTO lightweight cipher"

Formulation

Problem for a special prize!

NSUCRYPTO team organizes an open competition to develop a new light-weight block cipher. There are some requirements for it.

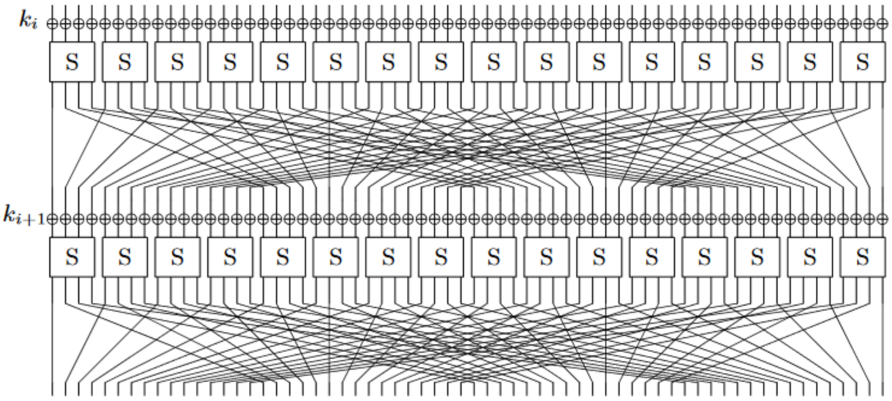
Block size — 64 bits.

Key size — 80, 96 or 128 bits.

Number of rounds — 32.

Structure — arbitrary. So, SPN, ARX, Feistel schemes can be applied or some new types of the structure can be proposed.

We kindly ask you first to study the well-known light-weight cipher PRESENT (2007). Try to realize what can be done better than in this cipher. Compare your solution with PRESENT: in realization, in cryptanalysis (linear, differential, algebraic, etc.). Give necessary arguments in favor of your decision.



Solution

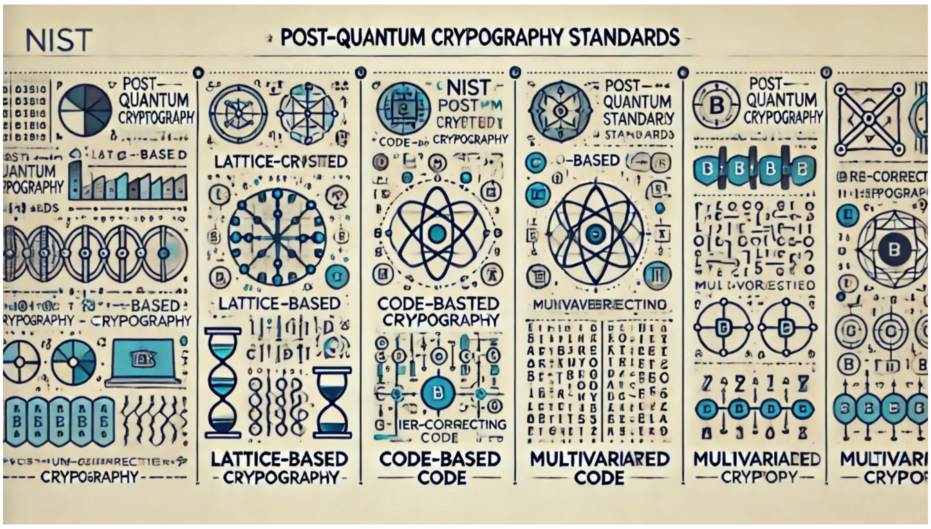
Participants proposed several ideas related to SP, Feistel networks and other types of cipher constructing. Unfortunately, there were too many attempts for solutions generated by ChatGPT and other services. The large part of them is of little content. But we can mention solution of Xiaopeng Zhao, Jianqiang Nii and Pengyu Qian from China. They proposed a bit-slice lightweight block cipher as a 28-round SP-network. S-layer is constructed with S-boxes of type 4 → 4, linear component operates with xor and shift operations. Key schedule is proposed with application of LFSR. Authors provide some research on resistance of the cipher to linear, differential and integral cryptanalysis. Due to the short time of the Olympiad round, this analysis seems to us the most serious.

4.14. Problem "Post-quantum signature"

Formulation

Problem for a special prize!

Represent any of widely known (by your choice) post-quantum digital signature schemes as a Mealy finite-state machine with minimum resource consumption of its hardware implementation as well as adequate cryptographic strength. Describe the state diagram of the machine and substantiate your solution.



This nice picture is taken from
`https://www.linkedin.com/pulse/understanding-nists-post-quantum-cryptography-shadab-hussain-wxt9e`

Solution

In fact this problem is still unsolved. There were several attempts but not too successeful.

5. Acknowledgement

We are very grateful to Irina Khilchuk for her kind helpful in preparing of this article.

REFERENCES

1. *Agievich S., Gorodilova A., Idrisova V., Kolomeec N., Shushuev G., Tokareva N.* Mathematical problems of the second international student's Olympiad in cryptography, Cryptologia, **41**:6 (2017), 534–565.
2. *Agievich S., Gorodilova A., Kolomeec N., Nikova S., Preneel B., Rijmen V., Shushuev G., Tokareva N., Vitkup V.* Problems, solutions and experience of the first international student's Olympiad in cryptography, Prikladnaya Diskretnaya Matematika (Applied Discrete Mathematics), 3 (2015), 41–62.
3. *Ayat S. M., Ghahramani M.* A recursive algorithm for solving "a secret sharing" problem", Cryptologia, **43**:6 (2019), 497–503.
4. *Geut K., Kirienko K., Sadkov P., Taskin R., Titov S.* On explicit constructions for solving the problem "A secret sharing", Prikladnaya Diskretnaya Matematika. Prilozhenie, 10 (2017), 68–70 (in Russian).

5. *Geut K. L., Titov S. S.* On the blocking of two-dimensional affine varieties, Prikladnaya Diskretnaya Matematika. Prilozhenie, 12 (2019), 7–10 (in Russian).
6. *Gorodilova A., Agievich S., Carlet C., Gorkunov E., Idrisova V., Kolomeec N., Kutsenko A., Nikova S., Oblaukhov A., Picek S., Preneel B., Rijmen V., Tokareva N.* Problems and solutions of the Fourth International Students' Olympiad in Cryptography (NSUCRYPTO), Cryptologia, **43**:2 (2019), 138–174.
7. *Gorodilova A., Agievich S., Carlet C., Hou X., Idrisova V., Kolomeec N., Kutsenko A., Mariot L., Oblaukhov A., Picek S., Preneel B., Rosie R., Tokareva N.* The Fifth International Students' Olympiad in Cryptography — NSUCRYPTO: problems and their solutions, Cryptologia, **44**:3 (2020), 223–256.
8. *Gorodilova A., Tokareva N., Agievich S., Carlet C., Gorkunov E., Idrisova V., Kolomeec N., Kutsenko A., Lebedev R., Nikova S., Oblaukhov A., Pankratova I., Pudovkina M., Rijmen V., Udovenko A.* On the Sixth International Olympiad in Cryptography NSUCRYPTO, Journal of Applied and Industrial Mathematics, **14**:4 (2020), 623–647.
9. *Gorodilova A. A., Tokareva N. N., Agievich S. V., Carlet C., Idrisova V. A., Kalgin K. V., Kolegov D. N., Kutsenko A. V., Mouha N., Pudovkina M. A., Udovenko A. N.* The Seventh International Olympiad in Cryptography: problems and solutions, Siberian Electronic Mathematical Reports, **18**:2 (2021), A4–A29.
10. *Gorodilova A. A., Tokareva N. N., Agievich S. V., Beterov I.I., Beyne T., Budaghyan L., Carlet, C., Dhooghe S., Idrisova V.A., Kolomeec N.A., Kutsenko A.V., Malygina E.S., Mouha N., Pudovkina M.A., Sica F., Udovenko A.N.* An overview of the Eight International Olympiad in Cryptography "Non-Stop University CRYPTO", Siberian Electronic Mathematical Reports, **19**:1 (2022), A9–A37.
11. *Idrisova V.A., Tokareva N.N., A. A. Gorodilova, I. I. Beterov, T. A. Bonich, E. A. Ishchukova, N. A. Kolomeec, A. V. Kutsenko, E. S. Malygina, I. A. Pankratova, M. A. Pudovkina, A. N. Udovenko* Mathematical problems and solutions of the Ninth International Olympiad in cryptography NSUCRYPTO // Prikladnaya Diskretnaya Matematika (Applied Discrete Mathematics). 2023, No. 4, P. 29-54.
12. *Kiss R., Nagy G. P.* On the nonexistence of certain orthogonal arrays of strength four, Prikladnaya Diskretnaya Matematika (Applied Discrete Mathematics), 52 (2021), 65–68.
13. *Tokareva N., Gorodilova A., Agievich S., Idrisova V., Kolomeec N., Kutsenko A., Oblaukhov A., Shushuev G.* Mathematical methods in solutions of the problems from the Third International Students' Olympiad in Cryptography, Prikladnaya Diskretnaya Matematika (Applied Discrete Mathematics), 40 (2018), 34–58.
14. *Tokareva N. N., Zaikin O. S., Idrisova V. A., Ishchukova E. A., Kolomeec N. A., Kalgin K. V., Kolomeec N. A., Kutsenko A. V., Kyazhin S. N., Malygina E. S., Pankratova I. A.* Mathematical problems and solutions of the Tenth International Olympiad in Cryptography NSUCRYPTO // Prikladnaya Diskretnaya Matematika (Applied Discrete Mathematics). In press. 2025.
15. `https://nsucrypto.nsu.ru/`
16. `https://nsucrypto.nsu.ru/archive/2021/total_results/#data`
17. `https://nsucrypto.nsu.ru/media/MediaFile/Book.txt`
18. `https://nsucrypto.nsu.ru/media/MediaFile/Book_cipher.txt`
19. `https://nsucrypto.nsu.ru/media/MediaFile/keystream.txt`