Problem 1. «RSA signature»

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

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

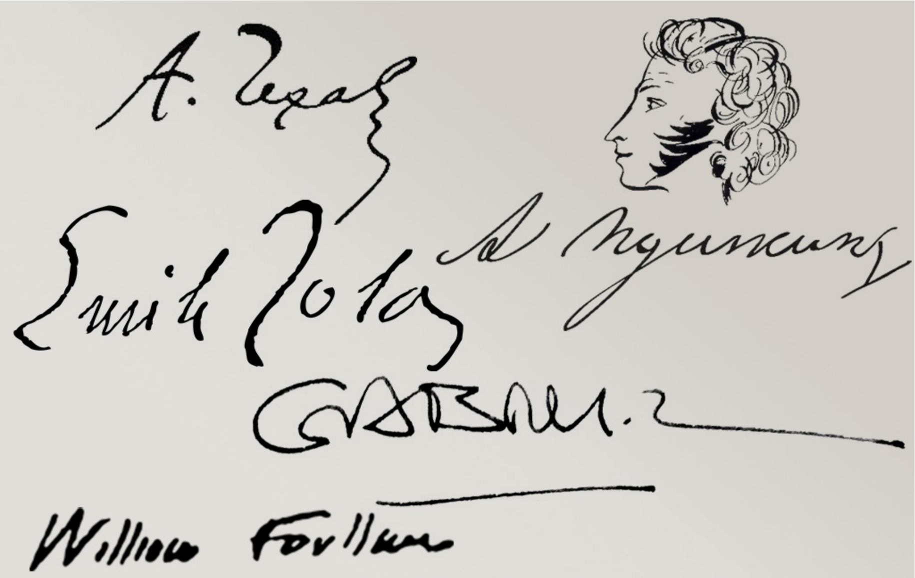
but he doesn't know the value

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

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.



Problem 2. «AntCipher 2.0»

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 $i, i \geq 1$, 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.

Problem 3. «Steganography and codes»

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.

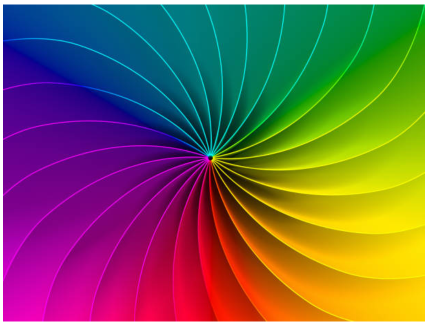
What does the Sam do with the picture? He should change it in such a way that nobody can visually fix the changing.

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.

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



Problem 4. «Weak key schedule for DES»

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 contains an open message, and the `Book_Cipher.txt` file 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
```

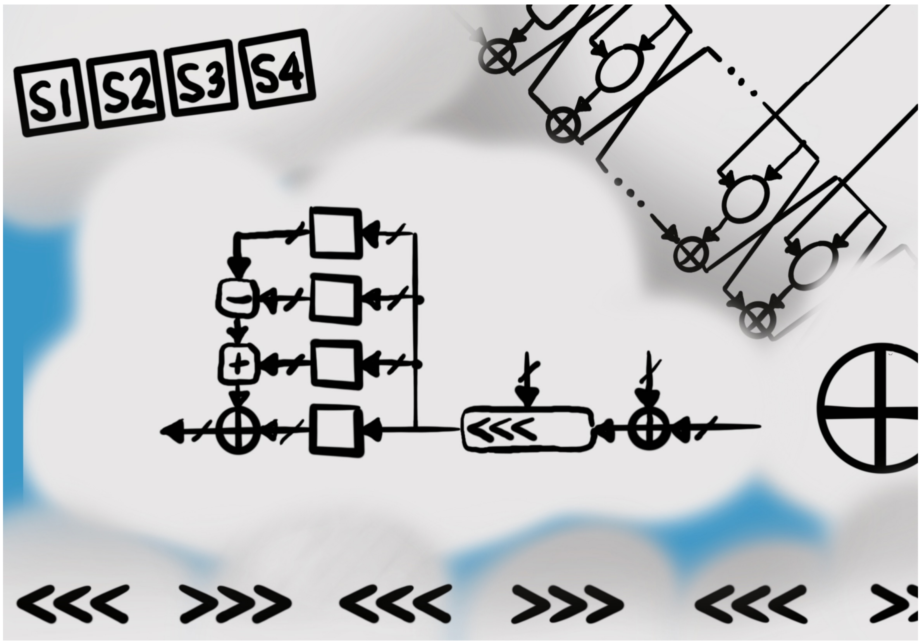
Some explanations. Description of DES algorithm can be found in the web, see for inst. `https://csrc.nist.gov/files/pubs/fips/46/final/docs/nbs.fips.46.pdf` Consider an example. We are talking about the correct implementation of DES, where all 16 round subkeys are used correctly. For example, if we take 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`.

Problem 5. «Reverse engineering»

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?



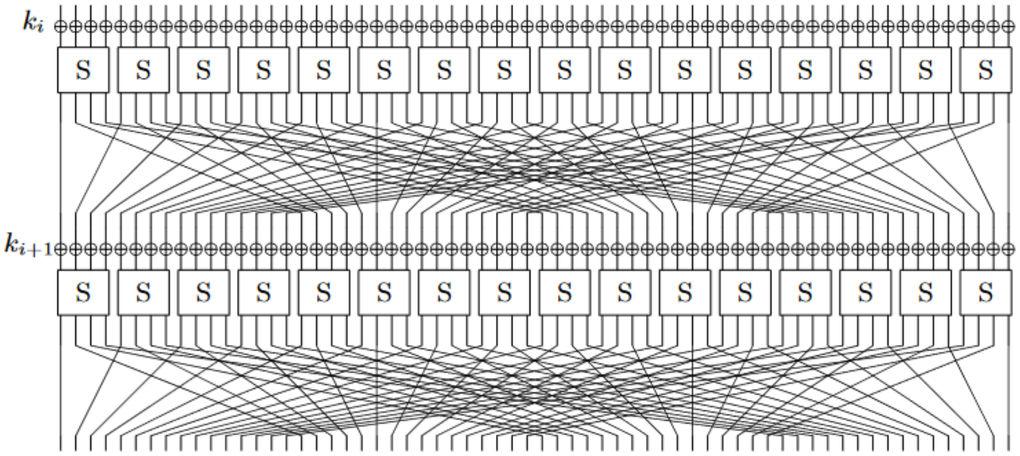
Problem 6. «Open competition: NSUCRYPTO lightweight cipher»

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.

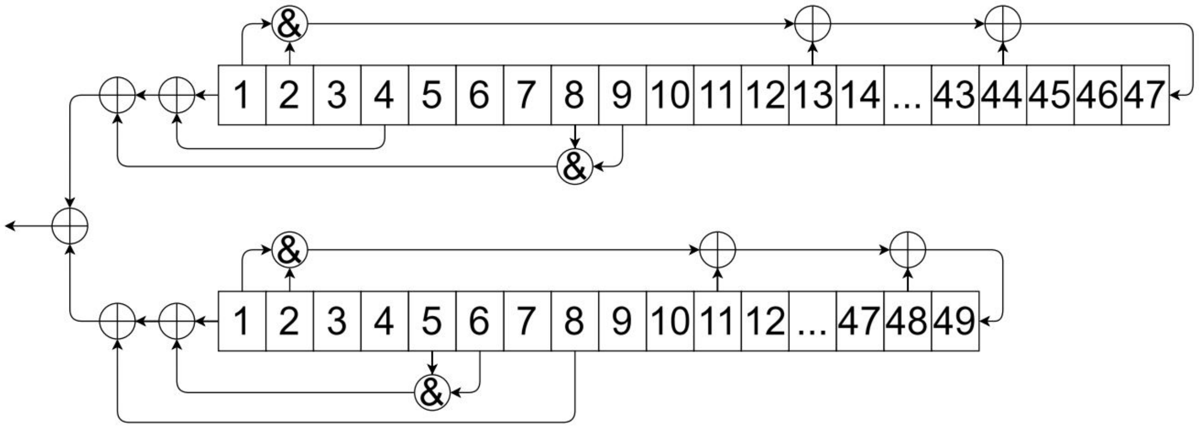
Block size — 64 bits.
Key size — 80, 96 or 128 bits.
Number of rounds — 32.
Structure — arbitrary. So, SPN, ARX, Feistel schemes can be applied or some new types of the structure can be proposed.

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.



Problem 7. «A nonlinear generator»

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. Also, the states $A(8192)$ and $B(8192)$ are known:

$A(8192) = (00101001110001001110111001100001010100000101110)$,
$B(8192) = (0000010000101001000011000001010111001110000100101)$.

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

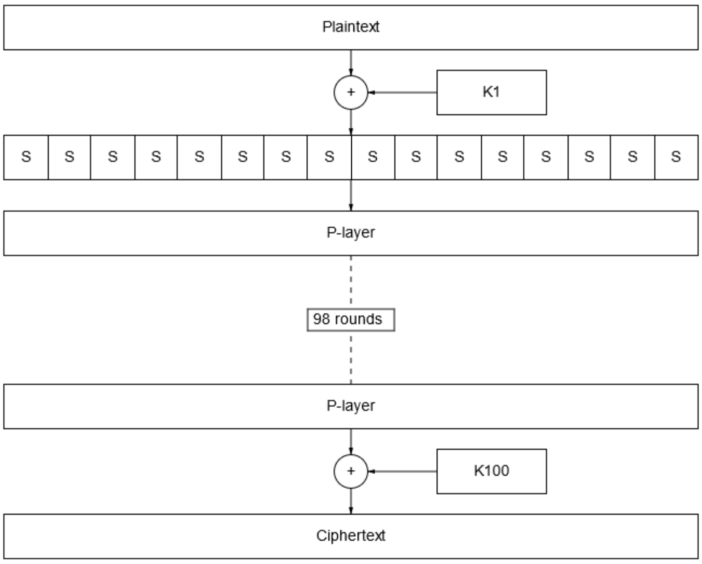
Problem 8. «Unsecure SP-network»

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

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

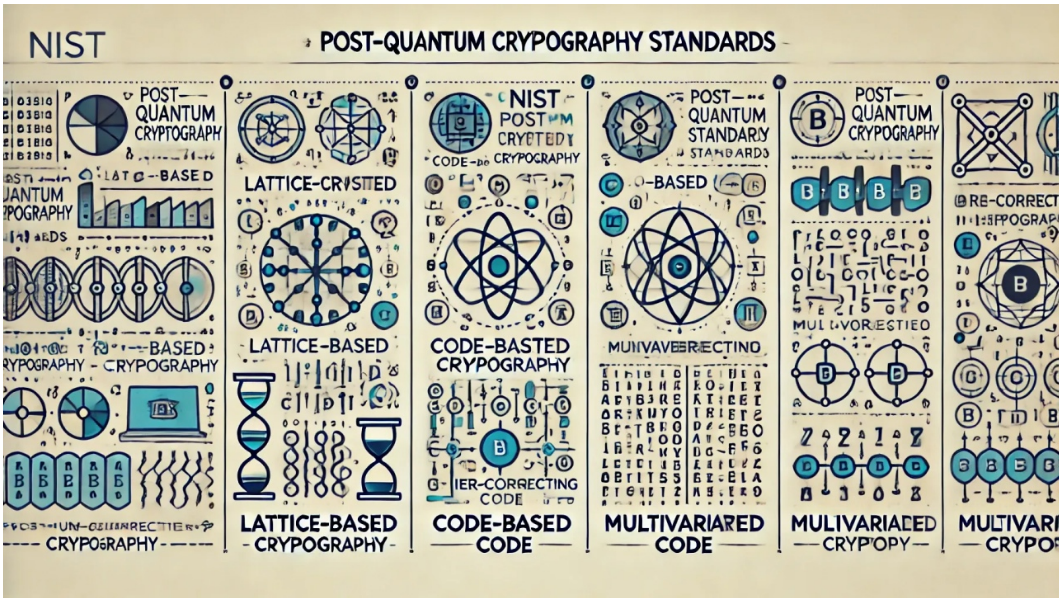
where m is the initial message (plaintext), $m = (m_1, \ldots, m_{32})$, where $m_i \in \mathbb{F}_2$.

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?

Problem 9. «Post-quantum signature»

Problem for a special prize!

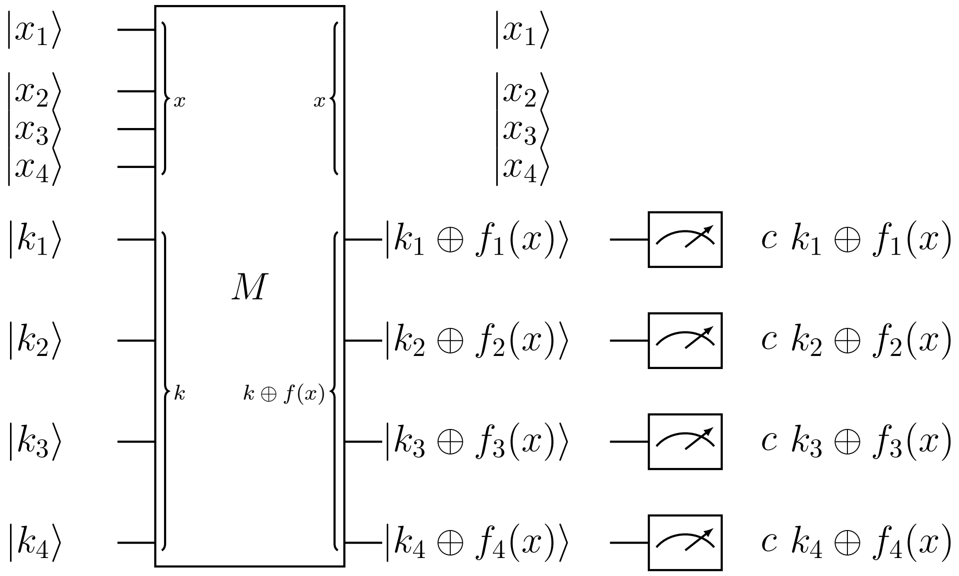
Represent any of widely known (by your choice) post-quantum digital signature schemes as a Mealy finite-state machine with minimum resource consumption of its hardware implementation as well as adequate cryptographic strength. Describe the state diagram of the machine and substantiate your solution.



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

Problem 10. «Unknown function»

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 measered and the result is observed.

Problem 11. «A simple hash function»

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 $H = (1 + 3 - 4 - 8 + 7 - 5) - (5 + 1 - 2 - 2 + 9 - 3) = -6 - 8 = -14$.

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.