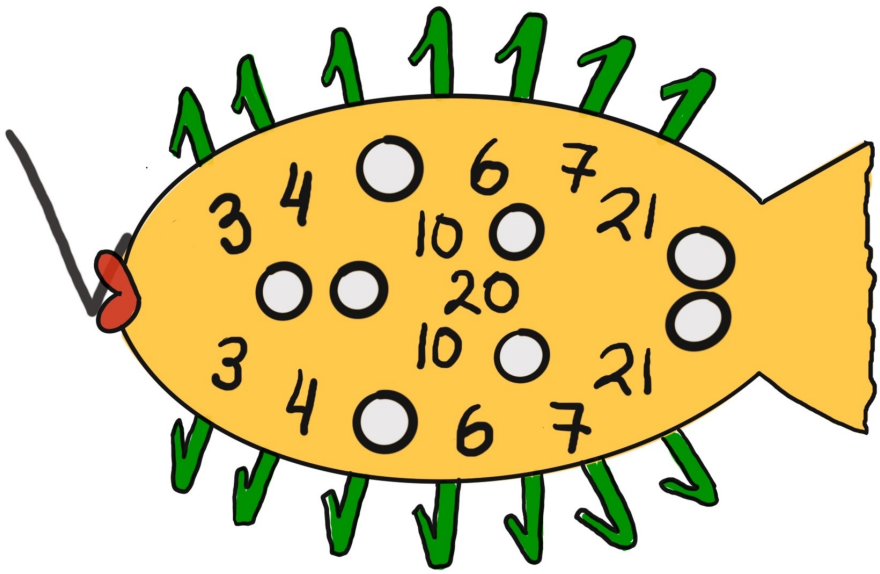
Problem 1. Cryptographic Fish

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.



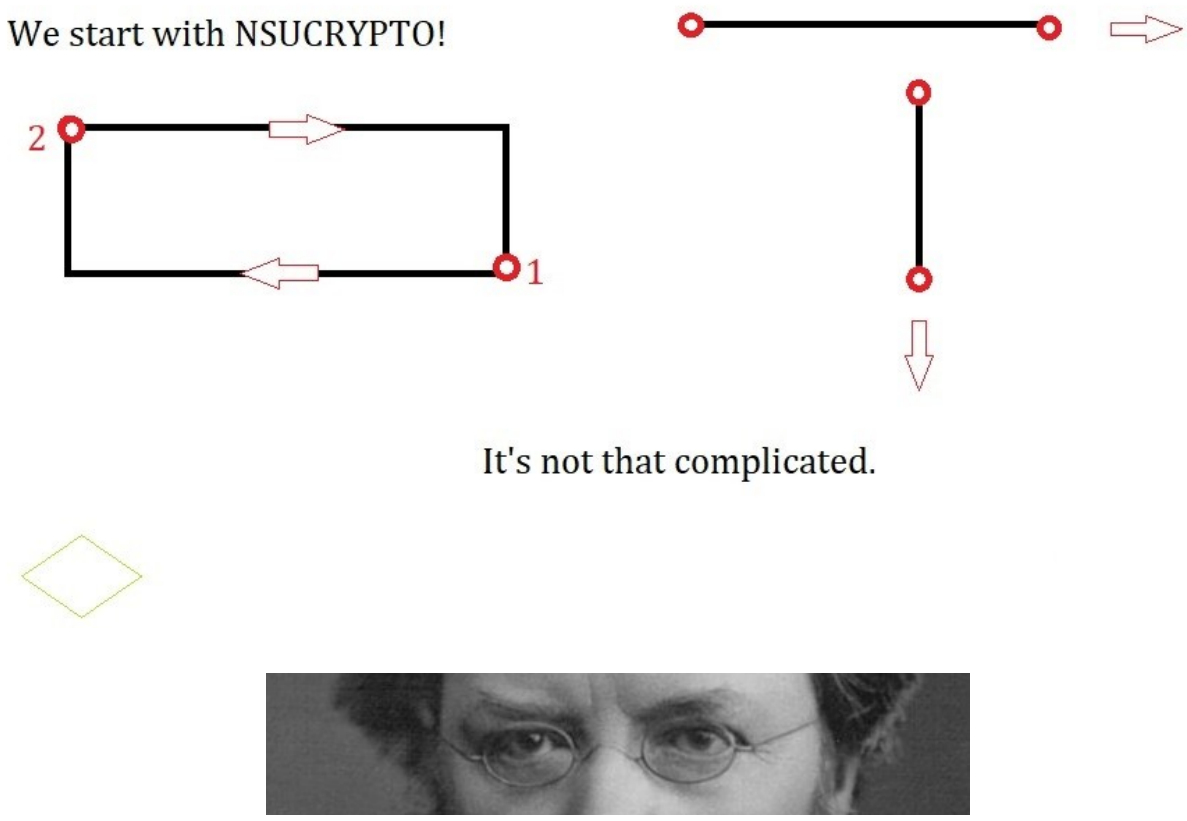
Please, could you find a key from it? It is a sum of the missed circled numbers.

Problem 2. «Decrypt with a hint»

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

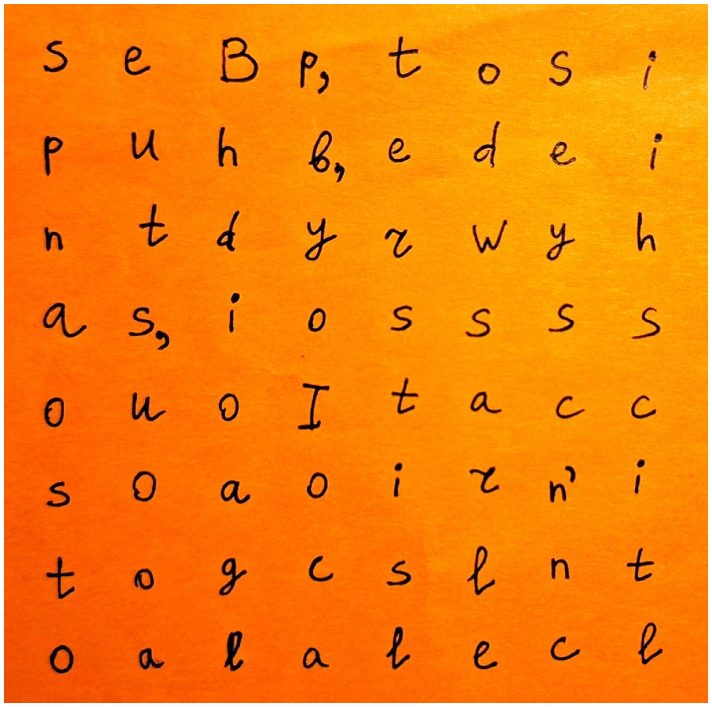
LFUZGAEPPXLSOANA

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

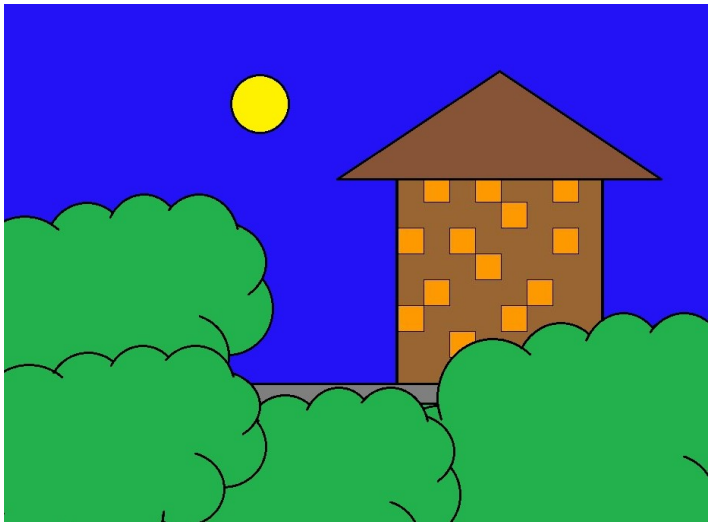


Problem 3. Alice's house

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

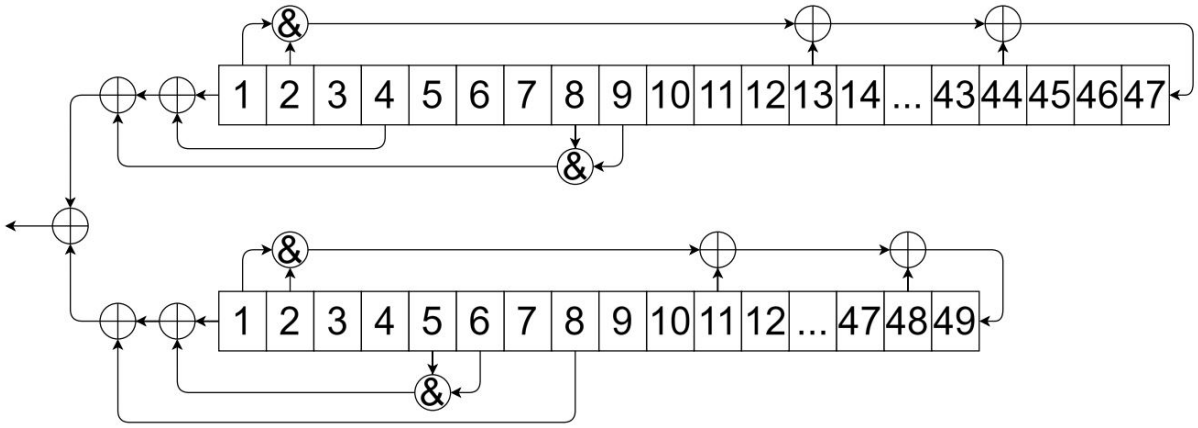


He guessed that it was a message from Alice. Bob looked around and immediately read it. What was the letter about?



Problem 4. «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?

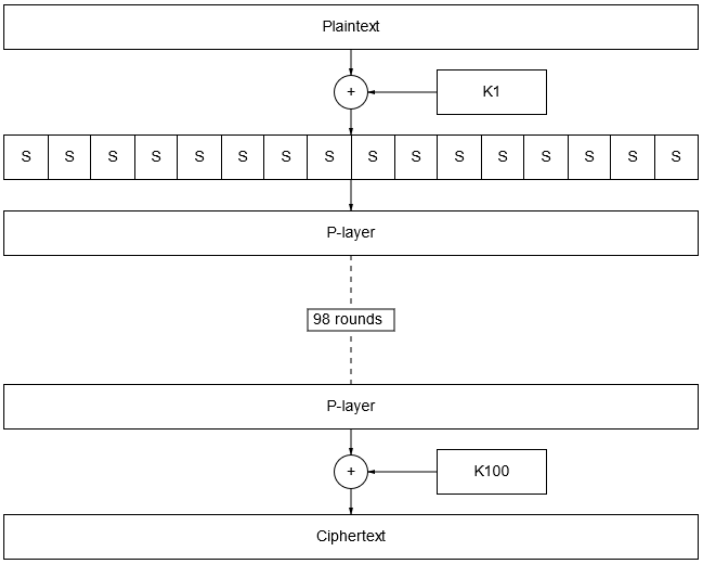
Problem 5. «Unsecure SP-network»

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 6. «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 7. «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

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

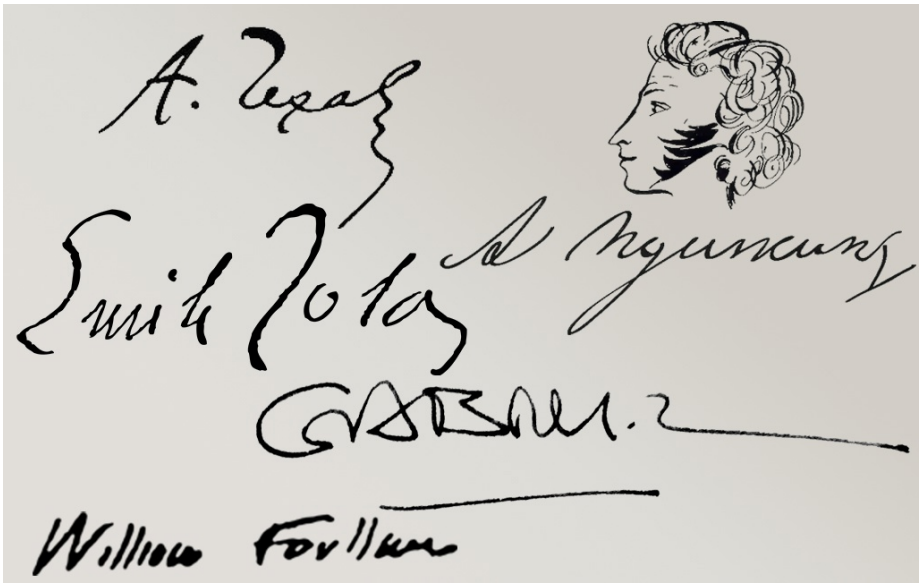
but he doesn't know the value

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

where

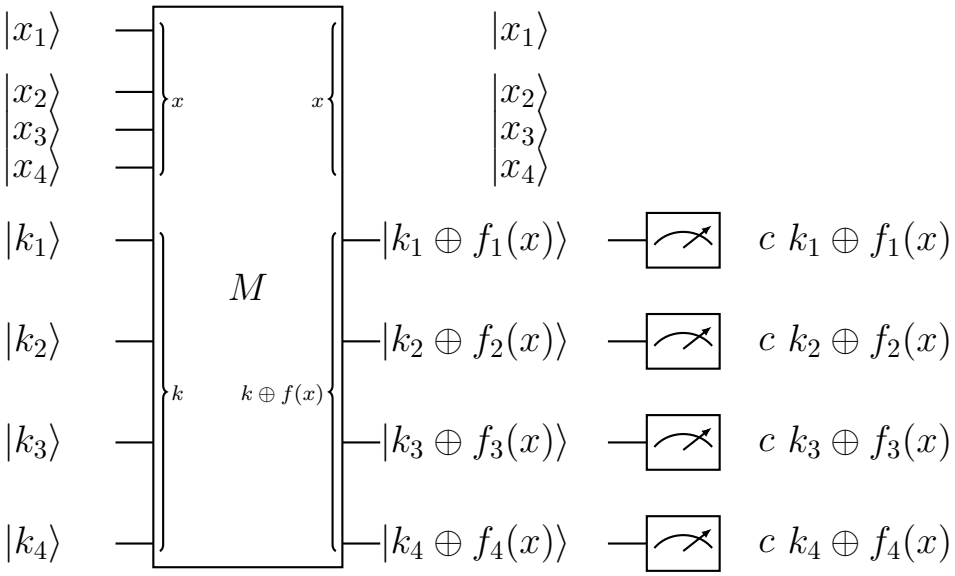
$$d_p := d \mod (p-1), \quad d_q := d \mod (q-1).$$

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 8. «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.