Problem 1. «Affine cipher»

Consider a 29-character alphabet {A,..., Z, α, β, γ}. Letters A,..., Z have numerical equivalents $0, \ldots, 25$, while numbers 26, 27 and 28 correspond to symbols α, β, γ.

We use a cryptosystem with plaintexts and ciphertexts being two-letter blocks, i. e. bigrams. For each bigram it is easy to find a numerical equivalent, it is an integer from 0 to $840 = 29^2 - 1$, determined by the rule $x \cdot 29 + y$, where x and y are the numerical equivalents of the letters of the bigram.

Encryption is implemented as an affine transformation $C = a \cdot P + b \mod 841$, where P is a plaintext, C is the corresponding ciphertext and the pair (a, b) is a secret key. Here a and b are integer numbers between 0 and 840. For example, if $a = 2$ and $b = 27$, then the bigram DP will be encrypted as $H\gamma$. In fact, for the bigram DP we put into the correspondence the number $3 \cdot 29 + 15 = 102$. Encrypting we get $2 \cdot 102 + 27 = 231$ that corresponds to the bigram $H\gamma$, since $231 = 7 \cdot 29 + 28$.

An analysis of the long ciphertext (for a fixed unknown key) showed that the bigrams "$\beta\ \gamma$", "UM" and "LC" are the most often found in this text. At the same time, we assume that the most frequent bigrams in English texts are "TH", "HE" and "IN".

Could you then decrypt the message "KEUDCR"? What about recovering of the key?

Problem 2. «Simple ideas for primes»

Problem for a special prize!

It is well known that prime numbers form a very special and mysterious class. They have too many applications in public-key cryptography (and not only there).

During many years (to be precise, hundreds of years) mathematicians think about simple constructions for prime numbers. Let us consider particular examples in this area.

- *Fermat numbers*, $F_k = 2^{2^k} + 1$, where integer k starts from 0, give us five prime numbers: $F_0 = 3$, $F_1 = 5$, $F_2 = 17$, $F_3 = 257$, $F_4 = 65\,537$. However, the next number, $F_5 = 4\,284\,967\,297 = 641 \cdot 6\,700\,417$, is composite as was proven latter. So far no more prime Fermat numbers were found.
- Several *Mersenne numbers*, $M_k = 2^k - 1$, are prime. For example, $M_2 = 3$, $M_3 = 7$, $M_5 = 31$, $M_7 = 127$, while M_{11} is already composite. Here we consider Mersenne numbers with k being prime, since it is a necessary condition for M_k to be prime. Up to now there are known 51 prime Mersenne numbers. The last one found prime Mersenne number is $M_{82\,589\,933}$; it was obtained in 2018 and up to now it is the biggest known prime number.
- From time to time, some original ideas appear. For instance, seven consecutive (by construction) numbers 31, 331, 3 331, 33 331, 333 331, 3 333 331 and 33 333 331 are prime! But the next number 333 333 331 is composite, since it can be divided by 17.

Let us say that Fermat prime numbers have the *sequence primality parameter* equal to 5, Mersenne prime numbers have it equal to 4, while for the last construction this parameter equals 7. So, the sequence primality parameter stands for the length of the longest subsequence of prime numbers in the sequence of numbers constructed.

Propose your own construction of integer numbers with the sequence primality parameter as big as possible. There is an important condition: every number from your construction should be presented explicitly (as in the examples above), i.e. the number should not depend on the previous numbers but should depend only on its index in the sequence.

Problem 3. «Mixed hashes»

Alice and Bob are exchanging with encrypted messages. To encrypt data, they use the **Present** cipher with an 80-bit secret key in ECB format. They record the information in the form of graphic files in `*.ppm` format.

The header of the `*.ppm` file consists of three lines of the form:

```
P6
X Y
255
```

where X and Y are the sizes of the graphic file in pixels horizontally and vertically, respectively.

So, in `mikki.ppm`, the header is:

```
P6
360 537
255
```

To be more secure, Alice and Bob decided that file headers should be removed before encryption. In order to be able to recover the file header, they agreed to transmit along with the encrypted file the hash value of the header itself, presented in **UTF-8** format. To do this, all header elements are written as a single line, using a space to separate the lines of the original header. The **sha-256** function is used as a hashing algorithm. So, for example, the following hash value will be generated for the header of `mikki.ppm`:

Heading `P6 360 537 255`

Sha-256 `999015795668c201db162926261ed979bc6e820aa1acfc385a0285685084d9f9`

Bob prepared eight files for Alice without headers, encrypted using the **Present** algorithm with the same secret key in ECB mode. He has sent the files themselves and hash values of the headers to Alice. While sending, the hash values were mixed up. So, Alice received eight files and eight hash values, but she does not know which hash value corresponds to which encrypted file. Could you help Alice to read the message from Bob?

Hash values received are

```
602a4a8fff652291fdc0e049e3900dae608af64e5e4d2c5d4332603c9938171d
f40e838809ddaa770428a4b2adc1fff0c38a84abe496940d534af1232c2467d5
aa105295e25e11c8c42e4393c008428d965d42c6cb1b906e30be99f94f473bb5
70f87d0b880efcdbe159011126db397a1231966991ae9252b278623aeb9c0450
77a39d581d3d469084686c90ba08a5fb6ce621a552155730019f6c02cb4c0cb6
456ae6a020aa2d54c0c00a71d63033f6c7ca6cbc1424507668cf54b80325dc01
bd0fd461d87fba0d5e61bed6a399acdfc92b12769f9b3178f9752e30f1aeb81d
372df01b994c2b14969592fd2e78d27e7ee472a07c7ac3dfdf41d345b2f8e305
```

Problem 4. «Column functions»

Problem for a special prize!

Alice wants to construct a super strong symmetric cipher. On this way she solves some hard mathematical problems.

Consider 2^n pairwise distinct vectorial one-to-one functions, $G_i : \mathbb{F}_2^n \to \mathbb{F}_2^n$, where $i = 1, \ldots, 2^n$. Applying these functions we construct a special binary matrix and then try to determine some its properties.

For $n = 2^m$, $m \geqslant 5$, define a binary matrix M of size $2^n \times n2^n$ as follows. The i-th row, $i = 1, \ldots, 2^n$, is a concatenation of values $G_i(0, 0, \ldots, 0, 0)$, $G_i(0, 0, \ldots, 0, 1)$, …, $G_i(1, 1, \ldots, 1, 1)$. The columns of the matrix M can be interpreted as vectors of values of $n2^n$ Boolean functions in n variables. We call them *column functions*.

Prove or disprove the following **conjecture** for at least one $m \geqslant 5$: for any matrix formed in the way described above there exist $2^{n/2}$ column functions $f_1, \ldots, f_{2^{n/2}}$ such that there is a nonzero Boolean function $f : \mathbb{F}_2^{2^{n/2}} \to \mathbb{F}_2$ satisfying the following conditions:

- for every $x \in \mathbb{F}_2^n$
$$f(f_1(x), f_2(x), \ldots, f_{2^{n/2}}(x)) = 0;$$
- for every $y \in \mathbb{F}_2^{2^{n/2}}$ the value $f(y)$ can be calculated using not more than $2^{n/2}$ addition and multiplication operations modulo 2.

Example. Let $m = 1$, then $n = 2$ and we construct matrix of size 4×8. Consider one-to-one vectorial Boolean functions G_1, G_2, G_3, G_4 from $\mathbb{F}_2^2$ to $\mathbb{F}_2^2$ defined by their vectors of values $(0, 1, 2, 3)$, $(0, 2, 1, 3)$, $(0, 3, 1, 2)$ and $(3, 2, 1, 0)$ respectively. Then the resulting matrix is

$$\begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

We need to find $2^{n/2} = 2$ column functions. Let f_1 and f_2 be defined as the first and the second columns of the matrix respectively, and $f(x_1, x_2) = x_1 \oplus x_2$ with the addition modulo 2. Then, $f(f_1(x), f_2(x)) \equiv 0$ since $f_1(x) = f_2(x)$ for any $x \in \mathbb{F}_2^n$.

Also, let f_1 and f_2 be the fifth and the sixth columns of the matrix. Then, giving $f(x_1, x_2) = x_1 x_2$ with the multiplication modulo 2, we obtain $f(f_1(x), f_2(x)) \equiv 0$ since $f_1(x) \neq f_2(x)$ for any $x \in \mathbb{F}_2^n$.

In the both cases the functions f can be calculated using only one operation. Note that the existence of such f implies that f_1 and f_2 are *algebraically dependent*.

Problem 5. «Primes»

Marcus invented a new cryptosystem. To start to work with it one should choose two big prime numbers p and q, then calculate $n = p \cdot q$ and $m = p + q$. The number $n \cdot m$ will be used in the cryptosystem.

While testing the system Marcus has noticed that for the chosen numbers p and q the resulting number $n \cdot m$ ends with 2023. Is this possible?

2 users
0 information about the key
2 prime numbers
3 operations

?

Problem 6. «An aggregated signature»

Problem for a special prize!

Suppose that a big international organization, say **NSUCRYPTO association**, decided to organize its own news journal in the area of cryptography. The organization wants to publish only news that are verified by a large group of cryptographers. For this goal 10 000 leading experts in cryptography were invited to join the editorial board of the journal.

The following publishing politics was accepted. The news can be published if and only if it is signed by all members of the editorial board. But cryptographers do not want to use 10 000 individual signatures. Since they are cryptographers, they think about the aggregated postquantum signature that can not be divided into separate individual signatures.

So, **NSUCRYPTO association** kindly asks you to propose such a signature scheme. There are several requirements for it:

* the size of the signature should be not big. It can be about several kilobytes;
* the size of the public key (for checking the signature) should be small. It is desired that the key size will be constant (or close to constant) even if the number of experts is increased, say up to 20 000;
* signature verification should not take more than 2 minutes;
* the signature should be resistant to attacks that use quantum computers.

Problem 7. «A unique decoding»

Problem for a special prize!

Consider a binary error-correcting code C of length n. Recall that it is just a subset of $\mathbb{F}_2^n$ and we transmit only elements of C over a noisy communication channel. Sending $x \in C$, some bits of x can be inverted in the channel. Getting $y \in \mathbb{F}_2^n$, a receiver decodes it into the nearest by Hamming metrics element of C. The Hamming weight $wt(x \oplus y)$ of $x \oplus y$ which is equal to the number of ones in $x \oplus y$ is the exact number of errors. Here $\oplus$ states for XORing. Error-correcting codes are of great interest in communication theory and post-quantum cryptography.

Consider the principle of the maximum-likelihood decoding. Obtaining some $y \in \mathbb{F}_2^n$, we suppose that the number of errors happened, say d_y, is minimal possible, i.e.

$$d_y = \min_{x \in C} wt(x \oplus y).$$

Next, let $\mathcal{D}(y) = \{x \in C : wt(x \oplus y) = d_y\}$. Finally, we decode y into any $x \in \mathcal{D}(y)$.

We are interested in all cases of codes for which $|\mathcal{D}(y)| = 1$ for all $y \in \mathbb{F}_2^n$. In other words for such code every binary vector $y \in \mathbb{F}_2^n$ can be decoded in the unique way.

Q1 What codes C can provide this property?

Q2 What codes C that are linear subspaces of $\mathbb{F}_2^n$ can provide this property?

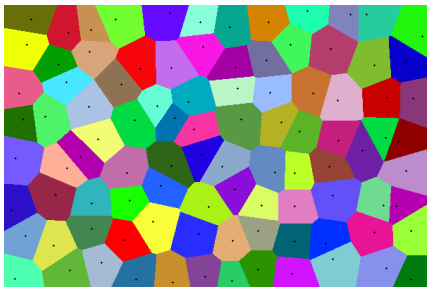
Remarks.

1) An example. There are so-called perfect codes C that allow to divide $\mathbb{F}_2^n$ into non-intersecting balls $B_r(x) = \{y \in \mathbb{F}_2^n : wt(x \oplus y) \leqslant r\}$ of some radius r centered in all $x \in C$. In other words, C is *perfect* if for some r it holds

$$\bigcup_{x \in C} B_r(x) = \mathbb{F}_2^n \text{ and } B_r(x) \cap B_r(x') = \emptyset \text{ for } x \neq x', \text{ where } x, x' \in C.$$

It is not difficult to see that any such code provides $|\mathcal{D}(y)| = 1$ for all $y \in \mathbb{F}_2^n$. But what else?

2) Some notions related to *Voronoi diagrams* can be helpful, see general mathematical definitions at `https://en.wikipedia.org/wiki/Voronoi_diagram`. In our problem we are looking for codes with non-intersecting discrete Voronoi cells for all $x \in C$.



Problem 8. «Algebraic cryptanalysis»

Bob decided to construct a new stream cipher **BOB-0.1**.

He used the binary key of length 8, say $K = (k_1, \ldots, k_8)$. Then he generated the binary sequence β such that $\beta_n = k_n$ for all $n = 1, \ldots, 8$ and for $n > 8$ it is defined as $\beta_n = \beta_{n-1} \oplus \beta_{n-8}$. Then Bob constructed the secret sequence γ for XORing it with a binary plaintext. The sequence γ is generated by the following rule: $\gamma_n = \beta_n \cdot \beta_{n+2} \oplus \beta_{n+7}$ for $n \geqslant 1$.

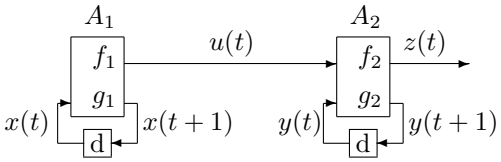
Alice intercepted the eight secret bits of γ after the first 1200 missed bits. These bits are `00100001`. Is she able to recover the original key K?

Problem 9. «Finite-state machines»

Problem for a special prize!

Alice decided to invent some generator that produces a sequence of maximal possible period relatively to its state size. Since she knows about finite-state machine, her generator G is constructed using two such machines A_1 and A_2:

- $A_1 = (\mathbb{F}_2^n, \mathbb{F}_2, g_1, f_1)$ with the state-transition function $g_1 : \mathbb{F}_2^n \to \mathbb{F}_2^n$ and the output function $f_1 : \mathbb{F}_2^n \to \mathbb{F}_2$, $n \geqslant 1$;
- $A_2 = (\mathbb{F}_2, \mathbb{F}_2^m, \mathbb{F}_2, g_2, f_2)$ with the state-transition function $g_2 : \mathbb{F}_2 \times \mathbb{F}_2^m \to \mathbb{F}_2^m$ and the output function $f_2 : \mathbb{F}_2 \times \mathbb{F}_2^m \to \mathbb{F}_2$, $m \geqslant 1$.



For any $t = 1, 2, \ldots$, let

1. $x(t)$ and $y(t)$ be the states of A_1 and A_2 respectively, $x(1)$ and $y(1)$ be the initial states;
2. $x(t+1) = g_1(x(t))$ be the next state of A_1 and $u(t) = f_1(x(t))$ be the output bit of A_1;
3. $y(t+1) = g_2(u(t), y(t))$ be the next state of A_2 and $z(t) = f_2(u(t), y(t))$ be the output bit of A_2.

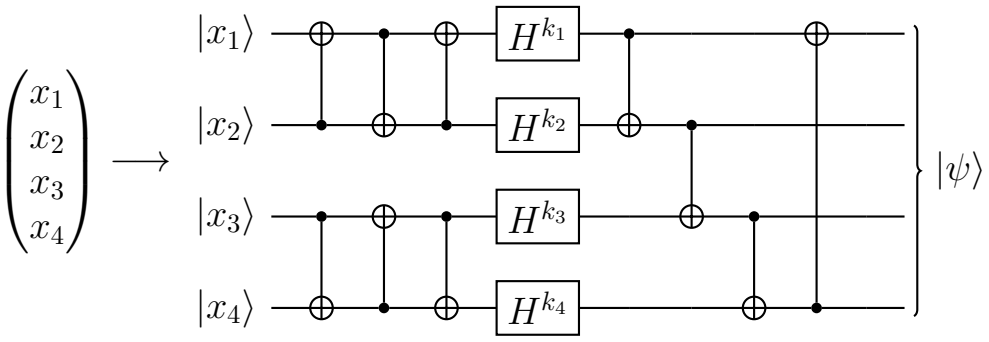
The sequence $z(1), z(2), z(3), \ldots$ is the output of the generator G. It is not difficult to see that it is eventually periodic whose the smallest period does not exceed 2^{n+m}.

Due to experiments, Alice noticed that the least period of the output sequence of G is less than 2^{n+m} if the Hamming weight of f_1 is even. Help Alice to prove or disprove this conjecture.

Remark. Recall that the Hamming weight of a Boolean function is the number of arguments on which it takes the value one.

Problem 10. «Quantum encryption»

Bob works in a field of quantum mechanics and he has some ideas how it can be applied for the encryption of secret messages. He developed a toy cipher that encrypts 4-bit words by using 4-bit secret key (k_1, k_2, k_3, k_4) and the following quantum circuit:



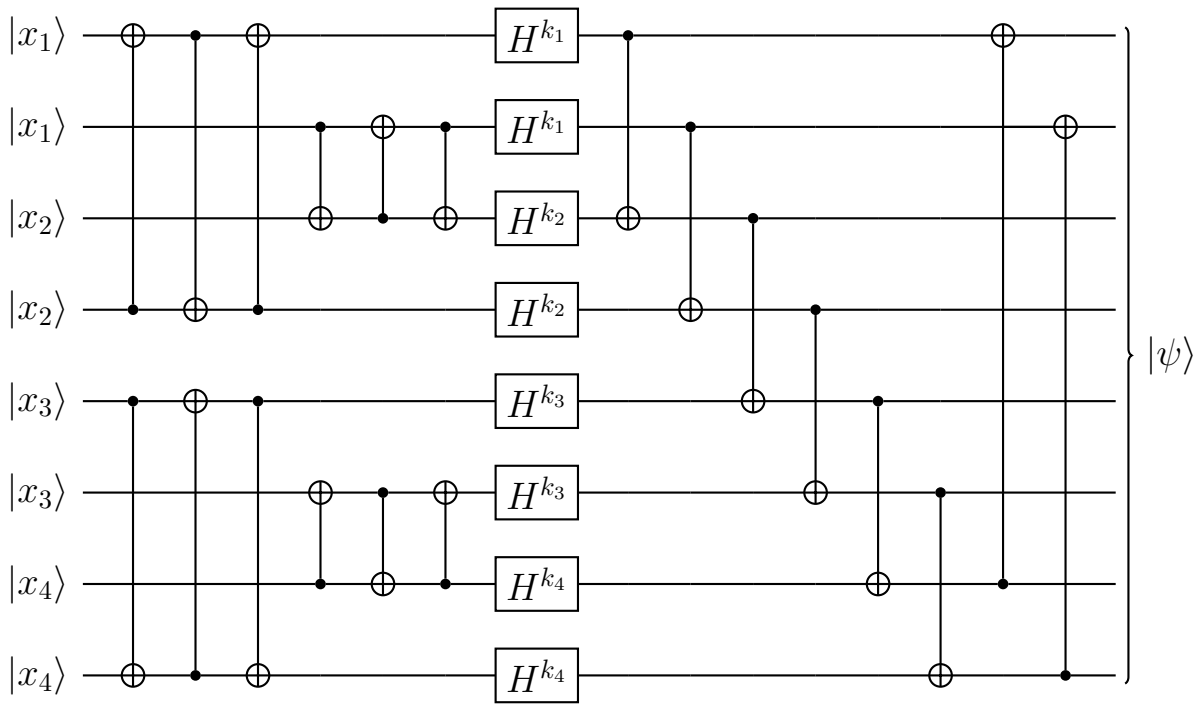
This cipher operates with 4-bit plaintext (x_1, x_2, x_3, x_4) that is initially encoded to the corresponding 4-qubit «plainstate» $|x_1, x_2, x_3, x_4\rangle$. This quantum state is an input of the circuit that consists of several single-qubit gates. Note that any quantum gate is a unitary operator that acts on the space of the states of the corresponding quantum system. The used gates are

Hadamard gate	$\|x\rangle$ — H — $\frac{\|0\rangle+(-1)^x\|1\rangle}{\sqrt{2}}$	acts on a single qubit in the state $\|x\rangle$, $x \in \{0,1\}$
CNOT gate	$\|x\rangle$ — $\|x\rangle$ $\|y\rangle$ — $\|y \oplus x\rangle$	acts on a pair of qubits in the states $\|x\rangle, \|y\rangle$, $x, y \in \{0,1\}$

The notation H^b, where $b \in \{0, 1\}$, means that if $b = 0$, the identity gate I is applied instead of H, while for $b = 1$ the gate H is considered.

The result of the encryption is the «cipherstate» $|\psi\rangle$ that is further transmitted via the quantum channel. The decryption procedure takes the state $|\psi\rangle$ and applies the inverse circuit.

Bob was advised to increase the number of qubits in order to reduce the effect of possible errors in quantum computation and quantum channel, so he decided to modify the circuit and make copies of qubits of the plainstate. The resulting circuit is the following:



Alice looked at the cipher and claimed that she would be able to reveal the secret key (k_1, k_2, k_3, k_4) if she knew some number N of certain amplitudes of the state $|\psi\rangle$. The state $|\psi\rangle$ is characterized by 256 amplitudes, so essentially we have $N \leqslant 256$.

Could you check the assumption of Alice and find the least possible value of N if the claim is correct?

Remark. Let us briefly formulate the 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 possess $|\alpha_0|^2 + |\alpha_1|^2 = 1$. The amplitudes α_0 and α_1 have the following physical meaning: after the measurement of a qubit which has the state $|\psi\rangle$, it will be found in the state $|0\rangle$ with probability $|\alpha_0|^2$ and in the state $|1\rangle$ with probability $|\alpha_1|^2$.

In order to operate with multi-qubit systems, we consider the bilinear operation $\otimes : |x\rangle, |y\rangle \to |x\rangle \otimes |y\rangle$ on $x, y \in \{0, 1\}$ which is defined on pairs $|x\rangle, |y\rangle$, and by bilinearity is expanded on the space of all linear combinations of $|0\rangle$ and $|1\rangle$. When we have two qubits in states $|\psi\rangle$ and $|\varphi\rangle$ correspondingly, the state of the whole system of these two qubits is $|\psi\rangle \otimes |\varphi\rangle$. In general, for two qubits we have $|\psi\rangle = \alpha_{00}|0\rangle \otimes |0\rangle + \alpha_{01} |0\rangle \otimes |1\rangle + \alpha_{10} |1\rangle \otimes |0\rangle + \alpha_{11} |1\rangle \otimes |1\rangle$. The physical meaning of complex numbers α_{ij} is the same as for one qubit, so we have the essential restriction $|\alpha_{00}|^2 + |\alpha_{01}|^2 + |\alpha_{10}|^2 + |\alpha_{11}|^2 = 1$. By induction, this process is expanded on the case of three qubits and more. Thus, the general form of the state of n qubits is

$$|\psi\rangle = \sum_{x \in \mathbb{F}_2^n} \alpha_x |x\rangle,$$

where amplitudes $\alpha_{00\ldots0}, \alpha_{00\ldots01}, \ldots, \alpha_{11\ldots1}$ have the same physical meaning as discussed before. Here we use more brief notation $|x_1\rangle \otimes |x_2\rangle \otimes \ldots \otimes |x_n\rangle \equiv |x_1, x_2, \ldots, x_n\rangle \equiv |x_1 x_2 \ldots x_n\rangle$.

Problem 11. «AntCipher»

Sam studies microelectronics, while his hobbies are biology and cryptography. He decided to unite all these areas in a research project aimed at constructing a tiny GPS tracker for an ant to monitor its movements.

The tracker consists of 3 modules: GPS, encryption, transmission. Once a minute, coordinates are determined, encrypted, and transmitted to a Sam's computer, where they are automatically decrypted. Due to the size limitation, the encryption module takes only a 2-bit plaintext and produces a 2-bit ciphertext, so the coordinates are divided into 2-bit blocks which are given to the encryption module. Sam has just developed a symmetric cipher called **AntCipher** for this purpose.

The cipher must be represented by the equation CNF=True, where CNF is a conjunction of disjunctions of literals, yet literal is a Boolean variable or its negation. In the Sam's CNF, x_1 and x_2 correspond to the plaintext, x_9 and x_{10} correspond to the ciphertext, while the remaining 6 variables are auxiliary. The equation is as follows:

$$(x_1 \vee x_2 \vee x_9) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_9) \wedge (\neg x_1 \vee x_2 \vee \neg x_9) \wedge (x_1 \vee \neg x_2 \vee x_9) \wedge$$
$$(x_1 \vee x_2 \vee x_3) \wedge (\neg x_9 \vee \neg x_{10} \vee \neg x_3) \wedge (x_1 \vee \neg x_2 \vee x_4) \wedge (\neg x_9 \vee x_{10} \vee \neg x_4) \wedge$$
$$(\neg x_1 \vee x_2 \vee x_5) \wedge (x_9 \vee \neg x_{10} \vee \neg x_5) \wedge (\neg x_1 \vee \neg x_2 \vee x_6) \wedge (x_9 \vee x_{10} \vee \neg x_6) \wedge$$
$$(x_1 \vee x_2 \vee x_3 \vee x_4 \vee \neg x_7) \wedge (x_2 \vee x_3 \vee x_4 \vee \neg x_7 \vee \neg x_8) = True$$

The problem is that, due to the limitations, the CNF must consist of at most 20 literals and at most 16 variables, while the presented one consists of 46 literals and 10 variables. Please, help Sam to construct an equivalent CNF that fits the limits. By equivalent it is meant that for each pair of plaintext-variables' values, the same pair of ciphertext-variables' values is derived in the equation.