periodica polytechnica

Mechanical Engineering
51/1 (2007) 23–32
doi: 10.3311/pp.me.2007-1.04
web: http://www.pp.bme.hu/me
© Periodica Polytechnica 2007

RESEARCH ARTICLE

Fast encryption of various types of messages

Igor Erosh / *Mikhail* Sergeev

Received 2007-03-18

Abstract

The problem of misleading illegal message interceptors while transferring messages is being solved in the article. Different variants of intensified defense of the most important fragments of messages with usage of both Boolean and matrix transformations in the GF (2) Galois field of transferred text are considered.

Keywords

transferring messages · important fragments · encryption · decryption.

Igor Erosh
Department of Computing Systems and Networks,Saint Petersburg State University of Aerospace Instrumentation, 190000, Saint Petersburg, Bolsaja Morskaja 67., Russia
e-mail: ierosh@sinet.spb.ru

Mikhail Sergeev
Department of Computing Systems and Networks,Saint Petersburg State University of Aerospace Instrumentation, 190000, Saint Petersburg, Bolsaja Morskaja 67., Russia

1 Introduction

There is quite a number of systems these days, known as cryptographic [1], which allow to protect transmitted information from unauthorized access. While estimating the usage characteristics of these cryptographic systems, the following factors can be used: the required computing resources to complete the encryption and decryption of information, the need for development of specialized hardware and its usage, the cost, etc. However, it is typical for modern distributed monitoring systems [2, 8] to fast process information of totally different types, such as commands sent to system modules, text messages, B&W and colour photos, photo plans, motion pictures and photo documents, and others. The required encryption/decryption speed in this type of systems can be preliminary estimated as reaching 6 MBits per sec, and more.

Given the high speed and diversity of information, as well as the fact that modern distributed monitoring systems are designed using modules of built-in type, choosing the encryption method calls for a thoroughly detailed approach.

Using exchange of information between two hypothetical subjects Alice and Bob as an example, this paper will discuss methods widely used in cryptography, as well as methods based on logical transforms of binary sequences employing multiplication of binary matrices by columns of fragments of messages being encrypted in the GF (2) field [3, 4].

2 Analysis of known approaches

One-Time-Pad.One-Time-Pad (Vernam's Cipher) [1] is built as follows: to the binary sequence of a message, e.g. S = 1011101001011101, Alice adds in mod2 a random key, e.g. K = 1001011100110111, and gets S⊕K= 0010110101101010. This sequence is sent to Bob.

Bob does symbol-by-symbol mod2 addition of the received message to the key Ḵ, thus restoring the original message:

S⊕K= 0010110101101010
K = 1001011100110111

S = 1011101001011101

According to C. Shannon's theory [5], One-Time-Pad can be an ideal secrecy system only when the following conditions are met:

a) the length of the key is equal to the length of the message;

b) the key is absolutely random;

c) the key is destroyed after every transmission session.

Meeting these requirements is very difficult. Thus, if there is a secret channel for the transmission of the key, then in some cases it is quite reasonable to send the message itself, but not the key, over this channel. Also, it is extremely difficult to obtain a true random key. Random sequence generators on feedback registers are known not to be able to provide true random sequences. To generate such sequences, it is necessary to have the number of 1s equal to the number of 0s, the number of pairs of 1s equal to the number of pairs of 0s, etc. Also, the probability of 2s, 3s and so on, must be decreasing following a certain rule. However, even if these conditions are met, the key will not be truly random, since it is obtained using deterministic methods, and when those are repeated, the same 'true' sequence will be realized. Sequences with good random properties can be produced using noise generators with subsequent processing [1]. Still, even when a sequence close to random is obtained, there remains a task of supplying this sequence to the sending and receiving sides. Shannon's third requirement is to destroy the key after every transmission. This means that a very difficult operation of supplying both sending and receiving sides with the same key is to be continually executed.

The encryption/decryption speed in this case is equal to one operation per bit of information. The proof of security is very strong, and with the message and the key long enough, the system will not be cracked.

A notion of ultimate secrecy that a system can provide is introduced in [6]. In this case the probability of every message *a posteriori* is equal to its *a priory* probability, and the unauthorized interceptor who grabs the message does not get any extra information.

However, using the One-Time-Pad, just mentioned, to control distributed systems transferring diverse information is not appropriate for the following reasons:

- providing intermittently long one-time keys to both sending and receiving sides causes significant delays and requires more sophisticated electronics;
- with a big number of simultaneously controlled objects in the system, every one of them would require a different key sequence generated and sent;
- ultimate proof of security for the tasks mentioned is not necessary, since the actual time of the information transferred is counted by hours, or days in the worst case.

Flow ciphers. Flow ciphers use pseudo-random sequence generators based on a few (usually three) feedback registers with their outputs combined. Pseudo-random properties are achieved here with non-linear functions [1]. Blaiser and Heintzmann have been the first to note that if the combining function allows the information of component function to pass through, then cracking such a cryptography system can be done much faster. In one of his works Thomas Siegentaler (1984) introduced the notion of the correlation-immune function that provides the best non-linearity when generator outputs are combined.

Flow ciphers usually require absolutely identical hardware implementation of generators on both sending and receiving sides. The general encryption scheme is close to that of the One-Time-Pad. The encryption/decryption speed is as high as 1 operation per bit. Software realization of the method usually gets more complicated, though, because the generators must be reset for every new sequence. As a result, the pseudo-random sequence has to be used for a long time, which makes the system's proof of security lower. In general, the proof of security of flow ciphers is determined by the parameters of the pseudo-random generators. In spite of all those shortcomings, many users are very interested in them because of their high speed [1]. Still, in the applications discussed, flow ciphers do not look efficient, since synchronization of a few pairs of pseudo-random generators is quite a daunting task.

DES-type ciphers. Methods of information protection based on DES (Data Encryption Standard) and its modifications (3DES, DEA, etc,), where encryption/decryption is done with a sequence of standard transformations of the key and the sequence to be encrypted (decrypted), have been widely spread for quite a while.

DES is a block cipher with a 64-bit block size. The key length is 56 bits, and 8 bits are added for parity control. A block of open text (64 bits) is subjected to permutation before processing, and to inverse permutation after encryption. This procedure is not a requirement and does not change the proof of security of the algorithm [1], so it is often omitted in software realization. Then the block is broken into left (L) and right parts, 32 bits each. 16 rounds of transformation, called function f, are performed, where data is mixed with the key. In each round the bits of the key are shifted, then 48 bits are masked out from the 56 bits of the key. The right half of the data (32 bits) is increased to 48 bits by applying permutation with expanding, is mod2 added to the 48 bits of the shifted and permutated key, undergoes transformation with 8 blocks, called S-blocks, thus producing 32 new bits, and is permutated again. The result of the f transform is mod2 added to the left half. This yields a new right half. And the old right half becomes the new left half.

All DES transforms are in public domain, they have been researched and published many times. Their down side is low speed and the need for constant key changes on both sending and receiving sides. The encryption speed is low, the proof of security depends on the size of the key. With a 56 bit key the number of key matching variants is $> 6.4 \cdot 10^{16}$. In fact, the number of variants is significantly smaller, since keys containing too few or too many 1s and 0s obviously would not fit and should be excluded from all 56 bit keys. Thus, a key containing

0,1,2 or 3 1s can hardly be used.

Since the DES cryptosystem has been known well for quite a while, there have been numerous attempts to break it. It is believed that there are DES-breaking algorithms [1] that allow to decipher an unknown text on a modern computer in 3-5 minutes.

We do not believe that using DES-type ciphers in distributed systems with a large number of controlled clients makes sense.

RSA-type ciphers. An RSA-type cryptosystem with modules of size around hundreds of decimal digits provides strong proof of security, that can even be increased by choosing large basic prime modules P and Q. The idea of the RSA-type system is simple enough.

The receiving side chooses two large prime numbers P and Q, finds the product P*Q=N, and openly publishes N. Only those who can factor N can find the Euler function φ (N) = (P-1)(Q-1). Then the receiving side chooses the public key E and publishes it along with N. E must conform to the following: (E, φ (N)) = 1 and 1< E < φ (N). Then the secret key D is calculated from DE $\equiv$ 1 mod φ (N). Finding D is possible using the extended Euclidean algorithm. Since E and φ (N) are coprimes, this congruence always has a solution.

When the original text fragment m is sent to the receiver, it is encrypted by raising it to the open power of E in mod N, i.e. m^E $\equiv$ S$_1$ mod N. Decryption is done by raising S$_1$ to the secret power D (mod N):

$$(m^E)^D mod N \equiv m^{ED} mod N. \tag{1}$$

It is known from the Number Theory [6] that if a $\equiv$ b mod N is true, then we can find an integer t such that $a = b + N \cdot t$. Since the secret key D is obtained from DE $\equiv$ 1 mod φ (N), then

$$ED = 1 + \varphi(N) \cdot t. \tag{2}$$

The value of ED substituted into (2) gives
m^{ED} mod N$\equiv$ $m^{(1+\varphi(N)*t)}$ mod N $\equiv$ m mod N, since m $^{\varphi(N)\cdot t}$ mod N $\equiv$ 1 mod N according to the Euler's theorem [6], i.e. the receiver will restore the original message.

For example, let P = 23, Q =31. The product of these two prime numbers N = 713. The Euler function $\varphi(N) = 22 \cdot 30 =$ 660. The canonical form of 660 is 660 = $2^2 \cdot 3 \cdot 5 \cdot 11$. Thus, the public key can be set to E = 17 (it will be coprimes with all the primes of the factored 660, which conforms to the conditions of choosing the public key). The secret key can be found from the congruence $17 \cdot D \equiv 1$ mod 660. Using searching of variants, let us find D = 233, in fact, $17 \cdot 233 = 3961 \equiv 1$ mod 660.

Suppose we need to send a message whose decimal form is 39 (binary representation is 100111).

Let us encrypt it by raising it to the power 17 mod 713:

$$\begin{aligned}39^{17} &\equiv ((39^4)^2)^2 \cdot 39 \equiv ((469)^2)^2 \cdot 39 \equiv \\ &(357)^2 \cdot 39 \equiv 535 \cdot 39 \equiv 188 mod 713.\end{aligned}$$

Thus, the encrypted form of 39 is 188. The latter is sent to the receiving side. In this transform we used the congruence's property that any of its arithmetic fragments can be substituted by the remainder in a mod.

Decryption is done by raising 188 to the power of 233 in mod 713. It is not difficult to complete, but quite cumbersome. To do it, let us represent 233 in a binary form: 11101001 and write:
$188^{233} \equiv (((((((188)^2)^2)^2)^2)^2)^2)^2 \cdot ((((((188)^2)^2)^2)^2)^2)^2 \cdot (((((188)^2)^2)^2)^2)^2 \cdot (((188)^2)^2)^2 \cdot 188 \equiv mod 713 \equiv ((((((407)^2)^2)^2)^2)^2)^2 \cdot (((((407)^2)^2)^2)^2)^2 \cdot (((407)^2)^2)^2 \cdot ((407)^2)^2 \cdot 188 \equiv (((((233)^2)^2)^2)^2)^2 \cdot ((((233)^2)^2)^2)^2 \cdot (((233)^2)^2)^2 \cdot (233)^2 \cdot 188 \equiv ((((101)^2)^2)^2)^2 \cdot (((101)^2)^2)^2 \cdot ((101)^2)^2 \cdot 101 \cdot 188 \equiv (((219)^2)^2)^2 \cdot ((219)^2)^2 \cdot (219)^2 \cdot 450 \equiv ((190)^2)^2 \cdot (190)^2 \cdot 190 \cdot 450 \equiv (450)^2 \cdot 450 \cdot 653 \equiv 8 \cdot 94 \equiv 39 mod 713$.

The proof of security of the RSA system is based on the fact, that even knowing N, it is very difficult to factor it and find the two primes P and Q. Of course, with small primes P and Q, N can be factored pretty easily. However, if P and Q are as large as 100 decimal digits, the enumeration will be in the range of 10^{100}. Since there are no general polynomial algorithms to factor numbers into primes, and the enumeration algorithm is exponential, factoring so large numbers cannot be performed on any existing computers.

The complexity of encryption/decryption, which significantly lowers the transmission speed, can also be considered a deficiency of the RSA cryptosystem. Changing modules is also a fairly difficult task, in both finding pairs of primes and their storing.

Realization of RSA-type algorithms in computer code is very ineffective, because it requires a lot of compound arithmetic operations. Hardwarewise, it calls for use of specialized processors, but even then the time required is very long. In a general case, using RSA for monitoring in distributed net systems is also ineffective.

3 Two-step message transmission employing modular arithmetic

A message can be sent directly from Alice to Bob in a few sessions using very large modules.

Let us first consider the case where Alice and Bob, over an open channel, agree to use for their message encryption some very large prime number P. For example, let P be 200 decimal digits long. Then Alice can cipher her message by raising it to some power x, known to her only, and send it to Bob. Bob can raise the received message to the power y, known only to him, and send it back to Alice. Alice will 'remove' her power x and send the message to Bob. Bob 'removes' his power y and reads the message.

Generally, the scheme looks as follows:

Alice takes the message which she wants to send to Bob and raises it to some power x, $(x, P - 1) = 1$:
$m^x \equiv S_1 mod P$
and sends it to Bob.

Bob raises S$_1$ to some power y, $(y, P) = 1$: $S_1^Y \equiv m^{xy} \equiv S_2 mod P$.

and sends it back to Alice.

Alice must remove her key x, and for this she has to take an x-th root from S_2. This can be done the following way:

If S_2 is raised to the power $k_1 = \dfrac{1 + (P - 1)t1}{x}$, where t_1 is integer, such that the numerator of the fraction k_1 is divided by x without a remainder, then the result will be $m^y \equiv S_3 \bmod P$.

This value (S_3) Alice sends to Bob.

To remove his key, Bob can raise S$_3$ to the power of k$_2$ $=\frac{1+(P-1)t2}{x}$, and as a result will restore the original message m.

Example 1 *Let P = 103. P-1 = 102 = 2 · 51. Alice wants to send a message m = 83.*

Alice raises 83 to a power known only to her, for example x = 35 (x and 102 are twin primes) in mod 103:

$83^{35} \equiv ((((83)^2)^2)^2)^2)^2 \cdot (83)^2 \cdot 83 \equiv (((91)^2)^2)^2)^2 \cdot 91 \cdot 83 \equiv (((41)^2)^2)^2 \cdot 34 \equiv ((33)^2)^2 \cdot 34 \equiv (59)^2 \cdot 34 \equiv 7$ *mod 103 and sends the result (number 7) to Bob. Bob raises the received number to the power known only to him, for example, y* $= 67$ *(this number is also a twin prime of 102):*

$(7)^{67} \equiv (((((7)^2)^2)^2)^2)^2)^2 \cdot (7)^2 \cdot 7 \equiv ((((32)^2)^2)^2)^2 \cdot 34 \equiv (((97)^2)^2)^2 \cdot 34 \equiv ((36)^2)^2 \cdot 34 \equiv$

(60)$^2 \cdot 34 \equiv$ *36 mod 103 and sends the result (36) to Alice.*

Alice 'removes' her power x, resolving

$k_1 \equiv \frac{1+102*12}{35} \equiv$ *35 mod 103 and raising 36 to the power of 35:*

(36)$^{35} \equiv (((((36)^2)^2)^2)^2)^2 \cdot (36)^2 \cdot 36 \equiv ((((60)^2)^2)^2)^2 \cdot 60 \cdot 36 \equiv (((98)^2)^2)^2 \cdot 100 \equiv ((25)^2)^2 \cdot 100 \equiv (7)^2 \cdot 100 \equiv$ *59 mod 103.*

The result (number 59) is sent to Bob. Bob 'removes"' his power y resolving

$K_2 \equiv \frac{1+102*44}{67} \equiv$ *67 mod 103 and raising 59 to the power of 67:*

(59)$^{67} \equiv ((((((59)^2)^2)^2)^2)^2)^2 \cdot (59)^2 \cdot 59 \equiv (((((82)^2)^2)^2)^2)^2 \cdot 82 \cdot 59 \equiv ((((29)^2)^2)^2)^2 \cdot 100 \equiv (((17)^2)^2)^2 \cdot 100 \equiv ((83)^2)^2 \cdot 100 \equiv (91)^2 \cdot 100 \equiv 41 \cdot 100 \equiv$ *83 mod 103.*

4 Forming the common key over the open channel

To form the common key over the open channel, Diffie and Hellman's idea can be used [1].

Let us assume that Alice and Bob have agreed to use some very large prime number P as a module. Besides, Alice and Bob chose for this P a primitive root g, i.e. the least number a, which conforms to $g^a \equiv 1$ mod P and is equal to P-1. More precise definition of 'primitive root in mod N' can be found in [6].

Alice raises the primitive root g to the power known only to her, finds the remainder in mod P g$^{k1} \equiv$ S$_1$ mod P, and openly sends the remainder S$_1$ to Bob. Bob raises the primitive root g to the power k$_2$ known to him only, obtains the remainder S$_2$ in mod P, and sends S$_2$ to Alice. Then Alice raises S$_2$ to the power of k$_2$, and Bob raises S$_1$ to the power of k$_2$ in mod Đ, and they both get the same key K = g^{k1k1} mod P.

Further on, Alice and Bob can both use this common key, for example, when working with a symmetrical DES system. If the key is to be changed, the exchange procedure is repeated with new k$_1$ and k$_2$. A deficiency of this method of key distribution is that common key formation is significantly delayed, a special calculator to find the remainders in large mod is needed, and synchronization has to be used while transferring diverse information from A to B.

Example 2 *Let P = 103. The primitive root for this P will be g = 2. To confirm this, let us put P-1 in its canonical form:* $102 = 2 \cdot 51$. *Since neither* 2^2, *nor* 2^{51} *cannot be compared to 1 in mod 103, 102 is the lowest power which proves* $2^{102} \equiv$ *1 mod 103 (Fermat theorem).*

Suppose Alice chose k$_1$ *= 7, and Bob chose k*$_2$ *= 11. Then the following data exchange will take place:*

$2^7 \equiv$ *25 mod 103*

211 $\equiv$ *91 mod 103*

and they both obtain the same key:

A: 91$^7 \equiv$ ***38*** *mod 103 B: 25*$^{11} \equiv$ ***38*** *mod 103*

The eavesdropper, knowing P and g and having caught S$_1$ *and S*$_2$*, will not be able to calculate the common key g*^{k1k2} *mod P. This is assured by the following.*

Let a^x *= b, and x is to be found. Now write*

x log a = log b, and find x = $\frac{log\ a}{log\ b}$. *That is, the solution is easily found.*

If we have

$$a^x \equiv b mod P, \tag{3}$$

in which the values of a, b, and P are known, then x in a general case is found by enumeration. The task of resolving the congruence of type (2) is called a task of discrete logarithmization. Despite the obvious achievements in this field in the recent past, for arbitrary modules P the solution is close to total enumeration. If P is in the range of 100 decimal digits, then the values of k$_1$ *and k*$_2$ *will be in the same range, and enumeration of 10*^{100}*will be required to resolve the congruence, which cannot be done – in any reasonable time limit.*

The reviewed method is very effective, but not appropriate for the tasks being considered in this paper because of the need for multiple exchange of information in key formation, and the complexity of the encryption/decryption algorithms (arithmetic operations of raising to a power, dividing, and finding the least positive remainders).

So, none of the methods reviewed so far can be effectively used while building an information exchange system with a large number of controlled modules in a distributed net monitoring system.

5 Encryption based on matrices with unit determinants over the GF (2) field

An encryption/decryption method utilizing multiplication of matrices with unit determinants over the GF (2) Galois field has been proposed in [3]. This approach is quite attractive, because only elementary Boolean operations (mod2 additions) are used in encryption/decryption. Both software and hardware implementations can be done quickly. Matrices with determinants equal to 1 in the Galois field produce cyclic subgroups that are attained by raising the original matrices to a certain power. Thus, in [3] the well-developed mathematics of matrix algebra, along with the methods of group theory, is proposed for encryption/decryption, which allows for encryption systems analysis and synthesis with the goal of their optimization.

Encryption utilizing matrices over the Galois field agrees very well with applying correction codes in both hardware and software, which allows for information protection from both transmission errors and eavesdropping. Let Us look at some examples of message encryption/decryption that use matrix multiplication in the GF (2) field. We have a matrix with elements from the GF (2) field, whose determinant is equal to 1 and is calculated in mod2. This means that all matrices with odd determinants will have a determinant equal to 1 in the field, and all matrices with even determinants will have a zero determinant in the field. Besides, the message fragment T to be encrypted will also be represented in a binary sequence. The result of encryption will be the product of the matrix multiplied by a column of the message from the left. For example, if the given matrix M is of size (7, 7) and the fragment T to be encrypted is 7 bits long (decimal representation of letter 's' in ASCII), then the result of encryption will be the following:

$$\mathbf{M} = \begin{pmatrix} 1110100 \\ 1001001 \\ 1010100 \\ 0001011 \\ 1010010 \\ 0000111 \\ 0010100 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}$$

Or, for short: $M \cdot s = @$, since the binary code 1000101is ASCII code of '@'.

To decipher the message, we have to find M^{-1}, the inverse of M, such that $M^{-1} \cdot M = I$, where I is a unit matrix like this:

$$I = \begin{pmatrix} 1000000 \\ 0100000 \\ 0010000 \\ 0001000 \\ 0000100 \\ 0000010 \\ 0000001 \end{pmatrix}, \text{ then, since } M^{-1} = \begin{pmatrix} 0010001 \\ 1010000 \\ 0101100 \\ 0100111 \\ 0101101 \\ 0111001 \\ 0010110 \end{pmatrix}, \text{ it}$$

$$\text{is easy to verify, that } \begin{pmatrix} 0010001 \\ 1010000 \\ 0101100 \\ 0100111 \\ 0101101 \\ 0111001 \\ 0010110 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}, \text{ or}$$

$M^{-1} \cdot @ = s$.

In the example above we encrypted a binary row of the message, and the size of matrix n was equal to the length of the binary sequence. Now, let us consider the case where the controlling command is much longer than n = 7, for example, this: 10111100101101. We will break the command into two-bit blocks, and will get 7 blocks. Encryption/decryption is done in the same fashion, using one multiplication in every step:

$$\text{Encryption: } \begin{pmatrix} 1110100 \\ 1001001 \\ 1010100 \\ 0001011 \\ 1010010 \\ 0000111 \\ 0010100 \end{pmatrix} \cdot \begin{pmatrix} 10 \\ 11 \\ 11 \\ 00 \\ 10 \\ 11 \\ 01 \end{pmatrix} = \begin{pmatrix} 00 \\ 11 \\ 11 \\ 10 \\ 10 \\ 00 \\ 01 \end{pmatrix}, \text{ decryp-}$$

$$\text{tion: } \begin{pmatrix} 0010001 \\ 1010000 \\ 0101100 \\ 0100111 \\ 0101101 \\ 0111001 \\ 0010110 \end{pmatrix} \cdot \begin{pmatrix} 00 \\ 11 \\ 11 \\ 10 \\ 10 \\ 00 \\ 01 \end{pmatrix} = \begin{pmatrix} 10 \\ 11 \\ 11 \\ 00 \\ 10 \\ 11 \\ 01 \end{pmatrix}.$$

In a general case, if the encrypted sequence is of length N, and the size of the matrix is n, the entire sequence is broken into n blocks, each being N'/n long, where N' is a sequence padded with 0s or 1s, such that it can be divided by n without a remainder.

While encrypting text messages, it is possible and advisable to encrypt messages with fragments that are multiples of a few letters, better yet if those fragments contain only parts of letters.

Suppose we want to encrypt the message 'We go home.'

Considering spaces, this message can be represented in ASCII code in a decimal form this way: 87 101 32 103 111 32 116 111 32 104 111 109 101 46. Since the number of symbols in the message is 14, let us break it into 7 fragments, 2 symbols each, encode them with ASCII, and multiply matrix **M** by a column of binary representations of the symbols. This yields

125 68 24 51 87 59 79 74 46 106 36 98 38 5.

If the sequence obtained is multiplied by matrix M^{-1} from the left, we will restore the original message 'We go to home'.

The entire message can be broken into 2 blocks, 7 letters each, and encryption can be done in two steps: first, the first block, then the second. Decryption should also be done by blocks. However, it is likely that better results will be obtained if the entire 24-letter message is broken into 2 letter blocks and encrypted in one step. Encryption will be faster and the result will be more difficult to crack. A still better method is to use fragments contain only parts of the letters, for example $2\,\frac{3}{7}$ of each letter.

Images can be encrypted using the same matrix. To do this, the entire image can be broken into 7 fragments, and these fragments are mod2 added together in accordance with the 1s of the matrix M. The resulting image will lack contours, so the interceptor will not be able to detect it.

So, the discussed method, using multiplication of the message to be encrypted by the matrix M from the left in the GF (2) field, allows for high speed encryption of diverse messages: commands, texts, images, without changing the encrypting matrix.

Since in this encryption only elementary Boolean mod2 addition operations are used, encryption and decryption are very fast. This is especially noticeable in comparison with those methods that require the use of arithmetic operations for encryption and decryption of the messages (RSA, systems using modular arithmetic), including hardware realization in programmable logic modules [?9].

Matrix Properties in the GF (2) field. Let us look at some useful properties of matrices with the elements from the GF (2) field. The power of the set (the number) of (n, n) matrices with the elements {0,1} is 2^{n^2}. This is quite obvious, since the number of elements of any such matrix is n^2. Among those matrices, however, there are some with even determinants. In the GF (2) field they have zero determinants. We exclude from the set of (n, n) matrices all singular matrices (the ones that have zero determinants). The remaining set of matrices constitutes non-communicative group on multiplication. And in fact, a set of these matrices is enclosed around multiplication, the association axiom holds true, the neutral element (it is the unit matrix I) axiom holds true, and the inverse element axiom holds true, i.e. for every matrix **M** there can be found a matrix of the same size, such that $M \cdot M^{-1}$ = I. The only axiom that fails is the general case communicative axiom. And the number of matrices of the (n, n) size is finite.

Number of matrices of (n, n) size with unit determinant in the GF (2) field. To find the number of (n, n) matrices with the determinant equal to 1, we will use the following procedure. If there is a matrix of (n-1, n-1) size with the determinant equal to 1, then a matrix of (n, n) size with the unit determinant can be obtained by adding to it from below a row of n-1 zeros and one 1, with an arbitrary column placed over the 1. Let Us find out how many different matrices of (n, n) size can be obtained from each (n-1, n-1) matrix.

Since the column over the only 1 can be arbitrary, the number of matrices should be increased by 2^{n-1}. Besides, in the obtained matrix a number of various linear combinations of rows can be performed, putting the result in place of the last row, as well as adding to the (n-1, n-1) matrix a column of n-1 zeros and one 1, performing the same linear combinations and putting the result in place of the last column. Then, if the number of matrices of the size (n-1, n-1) with the determinant equal to 1 is P (n-1), then the number of matrices of the size (n, n) with the determinant equal to 1 will be

$P(n) = (2^n - 1) \cdot 2^{n-1} \cdot P(n-1)$.

It is easy to determine that the number of (1,1) matrices with the determinant equal to 1 is 1 (this is the matrix (1)), and the number of (2,2) matrices with the determinant equal to 1 is 6. We write these matrices below:

$$\begin{pmatrix} 10 \\ 01 \end{pmatrix}, \begin{pmatrix} 11 \\ 01 \end{pmatrix}, \begin{pmatrix} 01 \\ 10 \end{pmatrix}, \begin{pmatrix} 10 \\ 11 \end{pmatrix}, \begin{pmatrix} 11 \\ 10 \end{pmatrix}, \begin{pmatrix} 01 \\ 11 \end{pmatrix}.$$

Therefore, the number of (3,3) matrices with the determinant equal to 1 will be $(2^3 - 1) \cdot 2^{3-1} \cdot 6 = 7 \cdot 4 \cdot 6 = 168$. Similarly, we find the number of (n, n) matrices with the unit determinant from:

$$P(n) = (2^n - 1)(2^{n-1} - 1)\ldots 3 \cdot 2^{n(n-1)/2} \qquad (4)$$

To use this estimate is very inconvenient, so let us find the low estimate of the number of matrices in the form of the power of 2.

The low estimate of the number of (n, n) matrices with the determinant equal to 1 can be obtained in the following way.

Let Us take P = $(2^n - 1)(2^{n-1} - 1)\ldots(2^2 - 1)$. We multiply its members in general and leave only the first 3 major members. If we denote R = n+(n-1)+(n-2)+…+3+2 = $(n^2+n-2)/2$, then the first three major members will take the form 2^R- 2^{R-2}-2^{R-3}… The following members will be either positive or negative, but their powers will be decreasing. Then P will be in this range:

$$2^R > (2^n - 1)(2^{n-1} - 1)\ldots(2^2 - 1) > 2^{R-1}.$$

We can take 2^{R-1} for the low estimate. Then the low estimate of the number of (n, n) matrices with the determinant equal to 1 will be:

$$N > 2^{n^2-2}$$

Table 1 shows the number of all the (n, n) matrices in the GF (2) Galois field (column 2), the precise estimate of the number of matrices with the determinant equal to 1 (column 3), and the low estimate of the number of (n, n) matrices with the determinant equal to 1 (column 4).

The low estimate of (20,20) matrices is 2^{398}. Using the obvious $2^{10} > 10^3$, we reckon that the number of (20,20) matrices with the unit determinant is larger than $2.5 \cdot 10^{119}$. This number is so large that without knowing M and only knowing, for example, its size, the eavesdropper, trying to crack the message, will not be able to enumerate all the matrices of this size.

Let us take an arbitrary matrix of the size (n, n) with the determinant equal to 1 in the GF(2) field and raise it subsequently to the powers 2,3,4,… A number s, such that M^s = I, will be found. This comes from the following theorem:

Theorem 1 (7) . *Let G be an arbitrary finite group. Then the set of powers of any of its elements g ∈ G: {g,g^2,g^3,…,g^s = e, where e is a neutral element of G} forms an always communicative group, called a cyclic group; here s is the order of the cyclic group, and*

$g^{-1} = g^{s-1}$.

Thus, for any matrix M with the determinant equal to 1 put subsequently to the powers 2,3,4,... there will always be some

Tab. 1.

n	Number of all matrices N = 2^{n^2}	Number of matrices with determinant equal to 1 (precise)	Low estimate for matrices of the size (n,n)
1	2	1	0.5
2	2^4	6	2^2
3	2^9	7*3*2^3	2^7
4	2^{16}	15*7*3*2	2^{14}
5	2^{25}	$31 \cdot 15 \cdot 7 \cdot 3 \cdot 2^{10}$	2^{23}
6	2^{36}	$63 \cdot 31 \cdot 15 \cdot 7 \cdot 3 \cdot 2^{15}$	2^{34}
7	2^{49}	$127 \cdot 63 \cdot 31 \cdot 15 \cdot 7 \cdot 3 \cdot 2^{21}$	2^{47}
8	2^{64}	$255 \cdot 127 \cdot 63 \cdot 31 \cdot 15 \cdot 7 \cdot 3 \cdot 2^{28}$	2^{62}
9	2^{81}	$511 \cdot 255 \cdot 127 \cdot 63 \cdot 31 \cdot 15 \cdot 7 \cdot 3 \cdot 2^{36}$	2^{79}
10	2^{100}	$1023 \cdot 511 \cdot 255 \cdot 127 \cdot 63 \cdot 31 \cdot 15 \cdot 7 \cdot 3 \cdot 2^{45}$	2^{98}

power s, such that $M^s = I$. Here s is the rank of the cyclic group produced by the matrix M, and $M^{-1} = M^{s-1}$.

We should note that s will always be a divisor (3), since the rank of any subgroup is a divisor of the rank of the [7].

Valid operations on matrices. Let us look at a set of matrices of the (n, n) size with the elements from the GF (p) field. All operations will be performed in the finite field GF (p). These operations will be: matrix multiplication with elements calculation in the GF (p) field, matrix division, multiplication of a matrix by a vector, determinants calculation, linear combination of matrices' rows. The rows (columns) of such matrices can be viewed as vectors, addition of two vectors being defined as addition of corresponding components of those vectors in the field, and multiplication of a vector by a scalar being defined as multiplication of every component by a scalar from the field GF (p). In this paper we are mainly interested in square matrices of the size (n, n).

The multiplication operation in the field is done in the regular way. The operation of dividing the matrix $M_1$by the matrix$M_2$is done by multiplication of the matrix$M_1$by a matrix inverse to the matrix M_2, i.e.

$$\frac{M_1}{M_2} = M_1 \cdot M_2^{-1}$$

Because a set of matrices forms a group, the division operation is valid.

For these matrices we can set forth and prove the following

1 Multiplication of two matrices $\mathbf{M}_1$ and $\mathbf{M}_2$ of the size (n, n) with the unit determinants in the GF (2) field yields a matrix with the unit determinant in the GF (2) field.

2 Permutation of any rows (and columns) of a matrix does not change its determinant in the GF (2) field. The rank of a cyclic group produced by this matrix does not change, which is easily proven by an example.

3 Replacing any row of a matrix with a linear combination (mod2 addition) of this row with any other rows of this matrix does not change the determinant of the matrix, but changes the rank of the cyclic group, produced by this matrix.

Randomizing properties of multiplication in the GF (2) field. One important property of the encryption method employing matrices in the *GF (2)* field should be noted here. Suppose we have a sequence of 1s and 0s, in which the probability of 1 is p, and the probability of 0 is respectively 1-p. We choose an arbitrary pair of symbols in this sequence and mod2 add them together. The probability of 1 in the resulting sequence will be P (1) = 2p(1-p), and the probability of 0 1- 2p(1-p) respectively. Now, let us mod2 add together 3 arbitrary bits of this sequence. Then the probability of 1 in the resulting sequence will be 3p $(1\text{-}p)^2 + p^3$. Having this in mind, let us build a table of probabilities of 1 after n mod2 additions (the probability of 0 will always be P (0)=1-P (1), so it will not be shown in Table 2).

To infer the formula in Table 2, the property of mod2 addition of n arguments of the function was used. The function resolves to 1, if the number of 1s in its true/false table is odd, and to 0 otherwise.

The right column of the table shows that even if the sequence is too asymmetrical (p = 0,9), after nine mod2 additions of the bits, the resulting sequence becomes practically pseudo-random (P (1) and P (0) $\approx$ 0.5).

To perform multiplication of the matrix by a column in the GF (2) field, the fragments to be encrypted are mod2 added together, and the larger the size of the matrix, the more additions are done, since the number of 1s in each row of the matrix is approximately equal to the half of the elements of the row, and is growing with n. Thus, the discussed method of message encryption randomizes the result, and the encrypted message should look like random.

Potential properties of the encryption method that uses matrices in the GF (2) field. If matrices producing cyclic groups of very big ranks are used for encryption, it is possible to solve some problems similar to those solved with modular arithmetic.

It is possible to transmit a secret message and form a secret key, if matrices over the GF (2) field are used, similar to the Diffie and Hellman's idea [1].

To show this, let us make three obvious assumptions.

1 To raise any matrix to any (even very large) power is easy. It

Tab. 2. Probability of 1 in a sequence obtained from mod2 addition of some bits of the original (probability P (1) = p) sequence

n	P(1)	P(1) with p = 0.9	Subtraction mod P(1)- 0.5
1	p	0.9	0.4
2	2p(p-1)	0.18	0.32
3	$3p(1-p)^2+p^3$	0.756	0.256
4	$4p(1-p)^3 + 4p^3(1-p)$	0.2952	0.2048
5	$5p(1-p)^4 + 10p^3(1-p)^2 + p^5$	0.66384	0.16384
6	$6p(1-p)^5 + 20p^3(1-p)^3 + 6p^5(1-p)$	0.368928	0.131072
7	$7p(1-p)^6 + 35p^3(1-p)^4 + 21p^5(1-p)^2 + p^7$	0.60485193	0.10485193
8	$8p(1-p)^7+56p^3(1-p)^5+56p^5(1-p)^3+8p^7(1-p)$	0.41611392	0.08388608
9	$9p(1-p)^8+84p^3(1-p)^6+126p^5(1-p)^4 + 36p^7(1-p)^2+p^9$	0.567108864	0.067108864
n	$P(1) = \sum_{k=0}^{s} C_n^{2k+1} p^{2k+1} (1-p)^{n-(2k+1)}$, where		
	$s = \left[\frac{n-1}{2}\right]$, $[x]$- rounded to integer of a lower value		

can be done using an algorithm, similar to the one that calculates the remainders (mod) when raising a number to a large power.

2 Finding an inverse matrix $\mathbf{M}^{-1}$ for any matrix $\mathbf{M}$ is also a relatively simple task (a system of n^2linear equations is to be solved). The complexity of this algorithm is O (n^2).

3 To find the rank of the group produced by the powers of the matrix is an enumeration task. The addressee normally does not have to know the rank of the group, its low estimate is enough. The eavesdropper, however, will have to do full enumeration, even knowing the low estimate.

In this case, the encryption of the matrix M of a big rank can openly be published. The secret message T (the text, the key, the key matrix, etc) is encrypted by Alice as M^aT. Bob additionally encrypt the message by the matrix M^b and sends it back to Alice. Alice removes her key by multiplying the message by the matrix inverse toM then Bob removes his key and reads the message. It should be noted that encryption is done simpler and faster than finding the remainders (mod). If the rank of the matrix is higher than 10^{40}, then the eavesdropper will have to enumerate all the variants, what seems a very difficult computing job.

Sine raising a matrix to a power and finding its inverse are considered computational tasks solvable with polynomial algorithms, and enumeration of all the powers of a matrix is done with exponential algorithms, the unauthorized interceptor will not be able to decrypt the message with matrices of large size. Of course, to provide this, matrices of significantly larger size have to be used, e.g. (150,150), then the upper limit of the power of the cyclic group will be $2^{150} > 10^{45}$. However, the algorithm described above allows to build matrices of this large size.

There is another way to create a public key. Alice raises the matrix $\mathbf{M}$ to her secret power $k_1$and sends $\mathbf{M}^{k1}$ to Bob. Bob raises the same matrix $\mathbf{M}$ to the power k_2 and sends $\mathbf{M}^{k2}$ to Alice. Alice raises the matrix received from Bob to the power k_1, and Bob raises the matrix received from Alice to the power k_2. Both get the common key $\mathbf{M}^{k1k2}$. If the rank of matrix $\mathbf{M}$ is equal to s, and the following conditions are met:

1 (s, k_1) = 1 and

2 (s, k_2) = 1,

then the rank of the matrix $\mathbf{M}^{k1k2}$is also equal to s. Otherwise, if s > 10^{30}, then k_1 and k_2 can be chosen from the same range, and it is unlikely that the interceptor will find M ^{k1k2}.

Alice and Bob can exchange a few messages using the key matrix M ^{k1k2} and then create a new common matrix.

A few key matrices of different sizes $M_1(n_1)$, $M_2(n_2)\ldots$ can be formed this way to encrypt the message T with their sequence, every time breaking the message into blocks of different sizes, for example, encrypting $M_1M_2 \cdot T$, and decrypting in the inverse order using inverse matrices $M_2^{-1}M_1^{-1}M_1M_2 \cdot T = T$.

While transmitting messages over channels with high interference, the following scheme can be used: encryption – coding – transmission – decoding – decryption of the message. All operations prove to be of the same type and can use the same equipment.

Additional reinforcement of protection with double and triple encryption. To strengthen protection of the entire message or its fragments, encryption with two or even three different matrices can be performed. With matrices M_1 and M_2 of the size (18,18) and (20,20) respectively, we choose an arbitrary text:

'The problem of misleading of illegal message interceptor while transferring messages is being solved in the article. Different variants of intensified defense of some most important fragments of messages with usage of both Boolean and matrix transformations in the GF (2) Galois field of transferred text are considered.'

We do subsequent encryption of this text with matrix M_1 first, and then with matrix M_2.

$$\mathrm{M}_1 = \begin{pmatrix} 111010001011011110 \\ 101001101110100111 \\ 111111000001001011 \\ 100101100100001111 \\ 100000101011100100 \\ 110010001101110000 \\ 011000100011001100 \\ 010100001001101101 \\ 101111100100100100 \\ 000000100110000010 \\ 010011010010010111 \\ 001111100100100100 \\ 011111111111100010 \\ 000000001001001101 \\ 101011001101110110 \\ 001001101010000000 \\ 000111000101001000 \\ 010010011001010111 \end{pmatrix},$$

$$M_2 = \begin{pmatrix} 1100101011110011001\,0 \\ 11110110100111010011 \\ 10001100101111010100 \\ 11011011111001001100 \\ 10100001010010011010 \\ 00001010100000101100 \\ 11101010010010011101 \\ 10100110001000100100 \\ 10110011011010001111 \\ 10000101100111101111 \\ 10001011110110111110 \\ 11101000000110010001 \\ 00011011010110000100 \\ 01010110101101001111 \\ 00001111110010010100 \\ 00110111001110001111 \\ 01010101101101100011 \\ 10111110010010111011 \\ 10110111110000101100 \\ 10010110010101111100 \end{pmatrix}$$

After encryption with the fist matrixM_1 it will look like this:

74 80 97 91 24 108 98 85 27 14 116 79 96 114 23 121 71 39 40 43 10 104 38 0 40 5 100 64 100 8 109 104 108 39 9 44 101 96 90 62 125 20 115 93 42 16 103 10 45 38 104 45 31 60 100 103 76 106 59 93 46 91 42 24 114 69 118 48 113 119 5 40 34 124 79 108 109 86 63 74 50 19 62 91 115 36 120 117 79 44 102 111 4 119 47 82 48 16 60 77 41 85 108 48 57 33 69 106 55 8 15 72 40 113 123 88 81 38 32 48 72 37 16 52 54 99 50 45 78 108 96 67 106 69 33 68 115 1 104 97 36 114 82 104 59 125 28 49 49 20 107 7 55 69 37 68 46 100 118 34 64 47 41 42 73 61 104 26 111 11 125 71 44 14 119 113 96 114 23 49 32 101 80 35 38 6 53 77 125 15 45 9 121 112 47 113 0 119 100 38 30 100 51 26 38 27 43 31 48 88 52 108 121 100 76 49 119 42 24 101 32 88 42 1 56 77 23 94 85 47 125 41 89 25 51 58 78 38 34 66 102 12 37 6 47 5 44 102 56 101 28 61 55 125 119 82 56 56 12 37 51 71 72 71 119 2 4 110 106 69 15 103 59 98 62 47 85 56 51 95 22 19 93 119 38 120 0 16 103 124 30 53 116 29 125 64 63 87 123 77 46 55 54 60 20 47 44 96 16 98 102 86 43 64 97 94 60 4 36 33 102 121 68 43

And after the second encryption:

8 18 82 29 6 22 93 97 62 65 121 30 114 52 104 89 116 34 73 34 56 42 65 77 27 36 123 73 96 96 20 122 96 46 69 15 6 37 109 12 82 18 1 10 45 34 114 31 47 96 29 56 100 42 78 73 5 30 103 69 70 106 33 29 95 1 19 43 68 93 49 1 7 30 111 112 44 51 109 62 1 104 18 86 15 75 3 9 69 78 37 24 114 116 127 16 118 108 18 92 123 42 102 108 38 71 83 115 18 88 20 33 104 103 94 90 68 106 72 24 101 119 26 5 44 25 40 73 110 126 21 71 89 19 71 106 95 18 97 80 47 89 109 105 120 121 44 61 101 23 122 105 62 125 89 47 63 124 4 80 1 100 64 123 82 43 18 68 95 46 119 36 3 63 77 110 40 56 13 105 58 75 86 96 94 54 80 54 69 12 49 20 117 105 85 101 73 44 7 111 34 110 77 23 77 44 108 79 54 55 73 46 111 98 3 7 71 119 97 68 82 111 8 126 116 33 27 56 65 52 24 98 9 19 3 42 1 56 46 5 60 84 117 53 17 76 105 57 104 20 125 29 52 110 96 83 83 37 32 73 98 106 10 1 34 74 44 116 38 52 18 103 123 103 28 53 94 27 40 78 85 26 122 52 87 84 19 7 100 102 84 50 43 19 54 67 22 94 90 24 120 102 4 17 119 113 25 24 122 16 90 57 127 85 116 79 64 124 108 98 63 80 70 20

Having twice multiplied from the left the encrypted text by the matrix M_2^{-1}, and then by the matrix M_1^{-1}, we restore the original text.

$$M_1^{-1} = \begin{pmatrix} 000000001001000000 \\ 010110011000010010 \\ 010000000011101100 \\ 000100001111010001 \\ 000101101011011011 \\ 101101010100010010 \\ 110001000100101011 \\ 010001000000000101 \\ 001000101101010010 \\ 010011101101100111 \\ 000100111110000011 \\ 100001000101000010 \\ 010010000111010011 \\ 111010111000101101 \\ 011110011000111111 \\ 001100011100101011 \\ 100110010011001111 \\ 111011101100010100 \end{pmatrix}$$

and

$$M_2^{-1} = \begin{pmatrix} 00011110000000111101 1 \\ 11000101010101001100 \\ 11011110011011101111 \\ 10110110010001000100 \\ 01100110010110101100 \\ 00110011110110111010 \\ 01100010101010100100 \\ 00101000001000110111 \\ 01001010011111010011 \\ 00001111100011110011 \\ 00101010000010011100 \\ 00100010110100100010 \\ 00001010101110010010 \\ 01110011110110000101 \\ 00000111011010101110 \\ 11101101001110101111 \\ 11110000100111001101 \\ 10111101011110111000 \\ 11111000110110100000 \\ 10100110001001101011 \end{pmatrix}$$

6 Conclusions

The discussed method has a number of advantages over those widely used. In particular, it employs logical, not arithmetical, processing procedures, which significantly speeds up encryption/decryption. Method modifications allow for flexible changes of the volume of the information to be encrypted, from separate commands and text messages to colour images and video streams, at the same time providing increased proof of security of encryption at a lower speed, and vice versa.

Besides, software and hardware implementations are compatible with correction codes, which allows to protect information from both communication channel interference and unauthorized access.

References

1 *Bruce Shchneier Applied cryptography. Protocols, Algorithms and Source Code*, John Wiley & Sons, New York - Chichester - Brisbane - Toronto - Singapore.

2 **Sergeev MB, Astaokivich A, Vostrikov AA, Chudinovskij JG**, *Information-controlled systems on the Internet base Information-controlled systems*, 2002 (Russian).

3 **Erosh IL**, *Transmission with an implication. Information-controlled systems*, 2004 (Russian).

4 **Erosh IL, Skuratov VV**, *Addressed messages' transfer by means of matrix under GF field. Problems of information safety*, 2004 (Russian).

5 **Shannon CE**, *Communication theory of secrecy systems*, Bell System Technical Journal **28** (1949), 656-715.

6 **Adams WW, Goldstein LJ**, *Introduction to Number Theory*, Englewood Cliffs, Prentice-Hall, 1976.

7 **Erosh IL**, *Elements of the theory sampled groups*, 1998 (Russian).

8 **Sergeev MB, Simakov VV, Chernecov PL, Vostrikov A**, *Perspectives of the usage "Embedded Internet" technology in preventive security systems and teleconferences through the Internet channel, satellite communication or GSM channel*, International forum "Science and education integration in XXI century" (S. Petersburg, 11 September 2003 7), pp. 103-109 (Russian).