Journal of Engineering Science
Fascicle
Topic

Electronics and Computer Science
Computers and Information Technology

Vol. XXV(2) (2018), pp. 45-56
ISSN 2587 3474
eISSN 2587 3482

EXTENDED RSA-M ALGORITHM AS A WAY OF INCREASE COMPUTATIONAL COMPLEXITY OF CRYPTOSYSTEMS

Vyacheslav Kunev

CEO of the IT-company, Ph.D. candidate, TUM
9, Studentilor Str., Chisinau, Republic of Moldova
E–mail: kunev@deeplace.md

Received: July, 02, 2018
Accepted: July, 30, 2018

Abstract. Applications of extended and combined formant analysis of modern number theory are considered to protect binary information from hacking and deliberate distortion in various IT systems based on a modified RSA-m cryptosystem with fast key changes and encryption algorithms with independent transformation keys with guaranteed small or medium short-term secrecy. Such an approach can serve as one of the ways to increase the computational complexity of modern information security systems, increasing their crypto resilience and making them capable of overcoming the possible crisis of modern cryptography in the post-quantum period. Outlines the results of the development of high-speed ciphers based on controlled combinations of transformations.

Keywords. *crypto systems, RSA-m, formant analysis, modernization, expansion, fusion of algorithms - product cipher, crypto resistance, computational complexity, quantum and post quantum cryptography*

Introduction

The relatively low operation speed, but the high resistant crypto of the RSA make the cryptographs-developers search for different ways of refining this system for its use in the normal flow of information or for protection of information with short-term secrecy, in real time mode (from few minutes to months). As an example we can name mobile conversations or voice communication. We have explored the possibility of using the standard RSA algorithm modified, conversed into new one, and so named RSA-m, with several modifications. The essence of changes is in the idea of sending not encrypted information through open or even secured channels, but only some information (data) about its encryption. There are offered several options of modernization, which require additional research. The purpose of this approach is to increase the operation speed of the algorithm, similar to RSA system, for use of its abilities while real-time information transmission.

The following approach to improving the cryptographic strength of a modernized crypto system being designed is aimed at increasing the speed of a classical RSA for its use in real-time data protection. It is based on a combination of transformations performed on data encrypted by formant analysis algorithms and classical RSA in the form of a predetermined and / or randomly changing their sequence of application and is a complete encryption algorithm [1-3]. Many algorithms are known, for example, symmetric encryption [4], obtained by repeatedly repeating other encryption algorithms in some way. The

simplest example is the Triple DES algorithm, which is a threefold encryption using the usual DES algorithm and has been known since 1978. There are both Double DES (double) and Quadruple DES (fourfold), the first of which is practically no stronger than the usual DES, and the second is extremely slow. For the same DES, more complex combination options are invented, for example, the Ladder-DES algorithm, so named because of its relatively complex "ladder" structure, the "steps" of which are ordinary DES. Another option for obtaining composite ciphers is the sequential use of several different encryption algorithms with independent keys. Such a composite encryption algorithm when using truly independent keys and strong applied algorithms is very strong, but has many drawbacks, in particular, a low encryption speed, therefore sequential encryption using various algorithms has not found wide application (see, for example, [4, 7, 8].

In this article we consider one of the several procedures, based not on the transmission of the information itself, but based on sending of some indirect data about this information in real time. The volume (of the length for bits) of this data is much less than the original information and, therefore, these data can be transmitted in encrypted form with required crypto resistant through the channels with limited speed and bandwidth (ex. 64 KB/s) based on the use of cryptosystem RSA, for example, but without significant delays in time. For this we can significantly reduce the amount of transmitted information, for example, presenting it in formant form, what will allow to reduce its encryption (decryption) time, commeasurable with the bandwidth of mobile communication. Another feature or direction of the modernization of encryption algorithms of mobile communication is based in this case on the use of short keys, but with the provision of the high speed of their change (replaceability.). The author intends to consider this idea in another publication.

1.1 Some definitions, properties, and axiomatics of formant analysis

Note that the possibilities of the formant analysis noted in [1], described in detail in [2, 3] and selectively presented in this section, noted below, are purely demonstrative in nature and for an interested reader can only serve as a convincing argument for the need to carefully study this a new direction in number theory.

The key notion (concept) of formant analysis $[1-3]$ is the definition of formant on the base by $p-F_p(M)-$ for the number M (or unknown x, algebraic expression or polynomial M), which means it (them) *linear representation in the form of a three-term (or three-dimensional) mathematical construction*: $F_p(M) = pk + q$

where p is the base of the formant, k is the kernel or the integer part after dividing M by the base of p, and q is a non-negative integer remainder (integer).

For example let's consider the notation of the formant of mathematical expression, and namely of the binomial $M = X^2 + 5Y$. It's formant appears as follows: $F_p(X^2 + 5Y)$, which means: *formant of the algebraic binomial of the form* $X^2 + 5Y$ *by base p*.

If p = 5, than the formant of the aforesaid binomial appears as follows: $F_5(X^2 + 5Y) = 5k + (1,4)$.

The mathematical meaning of the formant in this case means the following: *if* X^2 *is not divisible by 5, then with any whole* X *and Y, the remainder of this binomial by 5 will be equal to 1 or 4*.

A non-negative remainder (it can be zero) is called formant parenthesis (or just parenthesis), containing one or more numbers. Amount of numbers in the parenthesis defines formant dimensionality.

In the formant structure there is so-called core (k) of the formant, i.e. the integer part from the division of this number or algebraic expression on by base p.

Thus, the formant is completely defined by three values p, k and q.

Any number is uniquely defined by the formant on the given base. There are several types of the format. For more information, see [1].

1.2 Basic rules of formant arithmetic

It is known from theory [2], that when summing (subtracting) formant parenthesis of the different bases p (n- parenthesis and m– parenthesis) in resulting parenthesis will contain a elements of type (n×m), where many of elements are repeated. One can use the apparatus of string arithmetic to reduce the amount of computation. Let's remind its basics and essence.

Rule 1. If in the sum of two parentheses we will add any number to one parenthesis, and subtract the same number from another parenthesis, the sum of the parenthesis will not change.

Rule 2. If we increase or decrease all numbers in difference between 2 parentheses, the result will not change.

Rule 3. Any parenthesis can be represented as the sum of two 0- parentheses. 0-parentheses – is the parentheses, which contains zeros.

Rule 4. Base p can be added or subtracted from any number in parentheses, and it doesn't change the value of the parentheses. Parentheses are considered to be comparable if at least one number (remainder), equal to another.

The brackets are considered comparable if they have at least one number (remainder) equal to another, for example, the brackets [1 - 3, 5] = [2-4] are comparable, since the numbers 2 and 3 are common, and the brackets [1, 3, 5] and [2, 4] are incomparable, since they have no common remainders.

The rules of string arithmetic are used in transformations of formant equations when calculating the sum of two identical incomparable formants a prime base p.

An interesting question is the comparison of the formants in terms of their equality or inequality to each other. If two different formants are compared, this leads to a linear Diophantine equation with two unknowns. If three or more formants are compared, then, respectively, the number of unknowns increases.

Let us give an example of solving a linear equation by converting the formants a unknowns variables X and Y to two identical and comparable formants

Example 1. Find a solution to the linear equation:

$$55X = 73Y + 11 \tag{1}$$

Let's assume $X\ and\ Y$ as their formants of base 5 on:

$$X = 5k + A,\ \ Y = 5p + B \tag{2}$$

It is clear, that A and B are less than 5[4]. After substituting of the formant into the initial equation (2), we get two following equations:

$$55k - 73p = C \quad \text{and} \tag{3, a}$$

$$73B - 55A + 11 = 5C \tag{3 ,b}$$

The last equation (3,*b*) will be written in the following way:

$$73B + 11 = 5(C + 11A) \tag{4}$$

Since the left part of this equation should be divisible by 5, then the minimum solution $B = 3,$ and $73 \cdot 3 + 11 = 230$, which means:

$$C + 11A = 46 \text{ or } 11A = 46 - C.$$

It is not difficult to see that, that five possible values of A are possible: A: $0, 1, 2, 3, 4$, which have 5 corresponding values C, and namely: $46, 35, 24, 13, 2$. Next, from the original equation (2) it follows, that Y is divisible by 11, therefore

$$Y = 5p + B = 5p + 3 = 11D \quad or \; p = 6 + 11z.$$

But since $p < 11$ (Y should be less than 55), the only value p - is 6, i.e. $p = 6$. Then $Y = 33$, and from the initial equation (1) follows that $X = 44$.

Thus the general solution of the original equation will be the following:

$$X = 4 + 73m, \;\; Y = 3 + 55m \tag{5}$$

Note. Linear equations solution, as we can see, can be found even not using well known methods (ex. Euler algorithm). But the peculiarities of the formant approach are purely methodical ones, and it is not an option to count on their great efficiency. Another case is if formant analysis is used for nonlinear equations solution. Let's show the possibilities of the formant analysis in working with nonlinear Diophantine equations.

Ex. 2. Find the equation minimum of solution

$$X^2 = 19Y + 7 \tag{6}$$

Since the formant X^2 can be represented[5] as $5p + (1,4)$, we have:

$$F_5[X^2] = 5p + (1,4) = 19Y + 7 = F_5[?] \tag{7}$$

Reducing the formants of the left and right parts of the equation to a common base

[4] Because they are formant remainders by base 5.

[5] The formant of the square of any number on the basis of 5 always has 1 and 4 in remainder.

$p = 95$, we obtain the bracket of remainders for the formant of the left part of the equation:

$(1,4; 6,9: 11,14: 16,19: 21,24: \mathbf{26}, 29: 31,34: 36,39: 41,44: 46,49: 51,54:$
$56,59: 61, \mathbf{64}: 66,69: 71,74: 76,79: 81,84: 86,89: 91,94)$

and for bracket of right part of formant: $(7, \mathbf{26}, 45, \mathbf{64}, 83)$.

Thus we can see, that the formant remainders of the left and right parts of equation have only two equal numbers 26 and 64, but only 64 is a square, that is why $X = 8$, from which follows that $Y = 3$; these are the smallest solution of the original equation (6).

2 Information protection according to the RSA-mab algorithm

The idea of the proposed approach is to use the numerical formants introduced in [1] and explained in detail in [2] and [3]. As shown above, numerical formants allow you to represent any number as the simplest linear structure. In this case, time spent on the operations on the calculation of formant and the encryption/decryption will be significantly lower than time of direct use of the algorithms of classical RSA cryptosystem.

It is known [1], that linear formants, regardless of the bit longue of the number and, with the use of only 3 parameters *can significantly reduce the informational length of the digital message*. The advantage of such approach is in the fact, that so-called the base or (module, the divisor) of formant can be any composite or simple number of significantly smaller, than it is required for the number encryption in classic RSA cryptosystem, but with a high frequency of its replacement

Below will be described several algorithms for the use of linear formants for the transmission of information having short-term secrecy. The advantage of these algorithms is a significant reduction in the total encryption and decryption time, even taking into account the use of additional code operations necessary to transmit a message.

2.1 AB1 Algorithm

Let us recall [2, 3], that any integer N number in the formant analysis can be represented in the form of binomial construction: $N = pk + q$, where p is the base of formant, k – formant's core, and q - integer remainder. Knowing these three parameters allows us very easy to restore the original number. Types, properties and characteristics of the formant algebra are described in [1].

Using of formant representation of integers, now it will be necessary to encrypt not the number N itself, but three small numbers. The difference is that N is a large number of the order of 10^{20} ... 10^{500} and higher, while p, k and q are any integers, simple or composite, the length of which is determined only by the requirement of the desired transmission rate in the open channel. It is recommended to choose the formant base p as some integer number, approximately equal to the RSA key width, or similarly to the secret keys that correspond to the block cipher, but do not reduce the transmission speed. This requirement makes it possible to use RSA keys of medium length, and the presence in the system of a high-speed generator of primes [5] will allow you to quickly change secret keys, creating additional difficulties for cyber saboteur.

For the implementation of the RSA-m algorithm in ROM a dynamic database is created. It can be for example in the form of P matrix, the indexed cells which store preliminary generated information. These indexed cells are used for matrix construction. For example matrix 100×100, can contain such information for 10 000 different formants.

And what is especially important is there that will also be enough short secret keys used to encrypt formant messages.

After each single use of all values p_{ij} of $\boldsymbol{P}$ matrix, algorithm provides automatic update of all matrix cells, located both on the transmission side and on the receiving side.

Depending on the temporal requirements to degree crypto resistibility of the algorithm encryption, matrixs in ROM of microprocessor can be built with fixed or flexible updating program, with automatic or manual transmission of one matrix data array to another. In one of the cases it can be one and the same matrix, where cell names are changed on the base of indexes. In this case values of the formant remainders q and cores k are encrypted by the RSA-m algorithm, crypto resistibility of which is guaranteed by the real-time change with a large frequency of secret keys for each discrete value (binary d-bit number) of the analogue signal or system (number, byte, block) in the open digital code message. On the receiving side the encrypted message is decrypted by a special procedure, which recognizes transmitted addresses of cells and decrypts other parameters of each formant. After that the true value of the "number message" as a formants sequence is restored. The message itself can be a text in any language, as image of any class or type, as speech or musical track, etc.

Figure 1 show the block diagram of AB1 algorithm.

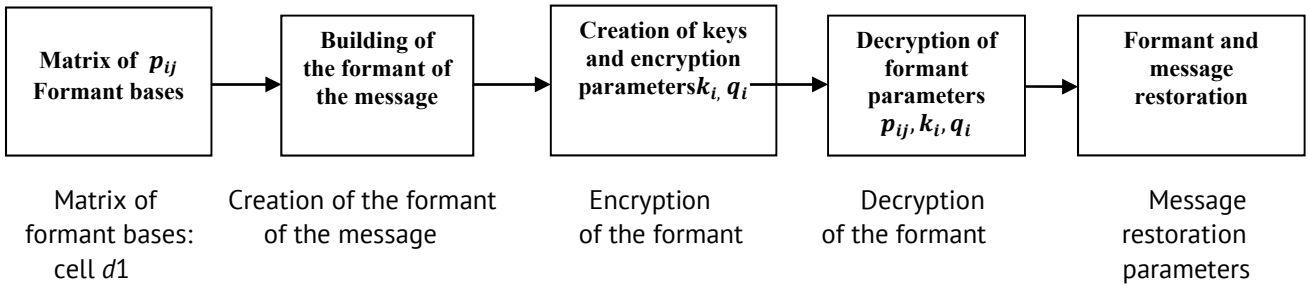

Figure 1. AB1 algorithm blockdiagram

1. Transmitted analogue signal, after ADC, forms the block: binary mesage in 32 (64) bit;
2. From the basic matrix block in ROM of the MPC formant the base $\boldsymbol{p_{ij}}$ is randomly selected and written $d1$ cell;
3. The digital block 64 bit according to p.1 ir represented further in the form of formant and its core $k_i = d2$ and remainder $q_i = d3$ are defined;
4. Crypto keys for encryption of numbers $k_i = d2\ and\ q_i = d3$ are chosen;
5. The d1d2d3 message about the formant is formed;
6. The d1d2d3 message about formant is encrypted;
7. Encrypted data are transmitted through an open communication channel;
8. Block 64… bit is received;
9. From the received block 64… bit the formant base $\boldsymbol{p_{ij}}$ is derived;
10. The numbers $k_i = d2\ and\ q_i = d3$ are derived from the block.
11. Formant-message: $F = p_{ij} \cdot k_i + q_i = p_i \cdot d2 + d3$ is restored.

Let's consider a simple example that illustrates the use of RSA-m algorithm.

Ex.3. Let's encrypt the message **"COIN"**. For simplicity we will use small numbers (in practice much larger (on several orders) numbers are used).

Let's choose two simple numbers $p = 3\ and\ q = 11$. Their product $N = 3 \cdot 11 = 33$.

We will find $(p-1)(q-1) = 2 \cdot 10 = 20$. Therefore, as a private key d, we can choose smaller and mutually simple with 20, for example, d=3 (or another: 7, 11, 13, 17, 19,...).

Now we select the public key – the some number e. There can be chosen any number, for which the following relation satisfied: $(\mathrm{e} \cdot d) = 1(mod\ 20)$,for example, if $d = 3$;then it should be implemented $(\mathrm{e} \cdot 3)(mod\ 20) = 1$; this condition is satisfied, for example, $e = 7$. Is real, $7 \cdot 3 = 21;\ 21(mod 20) = 1$.

Let's present the encrypted message as a sequence of whole numbers, applying as an example the following (random) correspondence: $O \rightarrow 1, I \rightarrow 2, C \rightarrow 3, N = 4$. In this case, the code message takes the form «$COIN$» $= (3,1,2,4) = S1$. Let's encrypt this message with the help of public key $\{e, N\} = \{7,33\}$.

$\mathrm{CryptoT1} = (\mathrm{P}^7)(mod\ N) = (3^7)(mod\ 33) = \mathbf{2187}(\boldsymbol{mod}\ \mathbf{33}) = \mathbf{9}$; *first we raise to a power 7 encrypted number, then divide the result by the module N. The remainder of the division gives the result of encryption.*

1. $\mathrm{CryptoT2} = (\mathrm{A}^7)(mod\ N) = (1^7)(mod\ 33) = \mathbf{1}(\boldsymbol{mod}\ \mathbf{33}) = \mathbf{1}$,
2. $\mathrm{CryptoT3} = (\mathrm{E}^7)(mod\ N) = (2^?)(mod\ 33) = \mathbf{128}(\boldsymbol{mod}\ \mathbf{33}) = \mathbf{29}$.
3. $\mathrm{CryptoT4} = (\mathrm{H}^7)(mod\ N) = (4^7)(mod\ 33) = \mathbf{16384}(\boldsymbol{mod}\ \mathbf{33}) = \mathbf{16}$.

Thus, public message $\mathrm{S1} = \mathbf{COIN} = (3,1,2,4)$ can be described (represented) in the form of encrypted SE1 message, i.e. numerical sequence of numbers, as $\mathrm{SE1} = (\mathbf{9}, \mathbf{1}, \mathbf{29}, \mathbf{16})$**,** *which, for example, corresponds to the text* **«TORY»** if while alphabet encoding was preliminary et, that $T = 9, R = 29, Y = 16$.

5. Now let's create, as an example, a methodically possible encrypted message for transmission over an open channel, taking into account insertions of service information, which can be of any sequence and content, for example, of the following kind:

$$\underbrace{003}_{for\ 3\ ranks}\ \underbrace{023}_{cell\ adress}\ \underbrace{009}_{T}\ \underbrace{001}_{O}\ \underbrace{029}_{R}\ \underbrace{016}_{Y}\ \ldots\ \underbrace{[0?101\&]}_{operation\ information}$$

• The first three decimal digits - the number of digits of the number to process part of the code; it determines the length of the machine word - that is, every three decimal digits.

• The second group of three decimal digits is the number of the matrix cell in the ROM MC or PLC, which must be selected from the memory by the controller on the receiving side.

• The third group of decimal digits is a sequence encoded by RSA-m algorithms (it can contain any number of "triples" depending on the length of the transmitted bits -16, 32, 64, etc.). The last 6 (or more) digits - information about the working protocol: the end of the sent message, parity check, etc. Thus, sending 32, 64 ... -bits, the first 6 and the last 6 single-digit digits should be selected, which contain all the information about the "structure" of the message, necessary for subsequent decryption.

As a result, was created an encrypted message {9, 1, 29, 16} which is the result of cryptography with a public key {7,33}. On the receiving side in the cell $p_i = p_{23}$ there are corresponding numbers: «private key d» and crypto key module: d=3; N=33.That is why private message «912916» is easily decrypted.

Let's decode the received encrypted message $(9,1,29,16) = \mathbf{TORY}$ on the basis of private key $\{d, N\} = \{3,33\}$. We get

Initial T1 $= (9^3)(mod\ 33) = 729(mod\ 33) = \mathbf{3}$,

Initial T2 $= (1^3)(mod\ 33 = 1(mod\ 33) = \mathbf{1}$,

Initial T3 $= (29^3)(mod\ 33 = 24389(mod\ 33) = \mathbf{2}$.

Initial T4 $= (16^3)(mod\ 33) = 4096(mod\ 33)$ $\mathbf{4}$; $(3,1,2,4) \rightarrow$«$\boldsymbol{COIN}$» QED.

2.2 AB2 Algorithm

Variant 1. Indexing memory cells

There are 10 000 of cells (100 lines ×100 columns in a matrix. The cells are numerated in a natural order and can be represented also in the form of two-index variable p_{ij}, where $ij = 00,01, \ldots 99$. For example the cell №457 has the index p_{0457}, and the cell №4057 will have the number or address-index p_{4057}.

1. Each cell $P_{ij}(e,d,n)$ of P matrix will contain now the index number, encrypted by RSA-m system. For example, the cell №0009 will contain the digit 3; cell №0001 will contain digit 1; cell №0029 – digit 2; and cell №0016 – digit 4, which correspond to the decryption with the key $d = 3$; $N = 33$; and encryption with the key $e = 7$. In the same way all other cells of this array will be filled.
2. Controlled mixing algorithm of cells of the variant 1 does not change the contents of the cell. They only changes its' index address.

In order to increase crypto resistibility degree of RSA-mAB algorithm it is recommended to change randomly the length of encrypted blocks, with corresponding change of Crypto keys length. The number of such arrays and their volume depends on the long-termness of secret information and on the volume of ROM controller, where RSA-mAB will be implemented.

Variant 2. The use of the Formant analysis

The block-message with the length of 32 (64)… bit is created.

1. From the matrix of the bases **P** there are randomly selected the base p_{ij} of formant. Formant address is recorded as message $d1$ (the base p_{ij} of formant, which is being created will be stored in cell $d1$).
2. The numeric view of the information block is formed as the formant - all the parameters of it (the core $k_i = d2$ and remainder $= d3$) are defined by the selected base p_{ij} and written in the messages $d2$ and $d3$.
3. A message $d1d2d3$ is being generated for transmission.
4. Keys for encryption of cores k_i and remainders q_i are generated.
5. The message about the $d1d2d3$ formant is encrypted.
6. Encrypted message is transmitted into the public communication channel.
7. The block of 64… bit is received.
8. Coordinate-address p_{ij} is extracted from the received block.
9. The value of the formant base is restored.
10. $k_i\ and\ q_i$ are extracted from the corresponding block.
11. The formant is restored on the base of encrypted message on standard formula.

As a similar example let's consider the encryption of the message «EDA" or its' digital code 651. Let's present the digit 651 code as a formant, i.e. as a sum of

multiplication and remainder, and as a basis choice, for example, random simple numbers 237, 54, 119 etc. Matrix of key in RSA-m system from variant 1 is replaced by the matrix of formant bases and forms the set of randomly selected bases of different lengths and properties (simple and composite). So, our message is: S2 = 651.

- As a similar example let's consider the encryption of the message «EDA» or its' digital code 651. Let's present the digit 651 code as a formant, i.e. as a sum of multiplication and remainder, and as a basis choice, for example, random simple numbers 237, 54, 119 etc. Matrix of key in RSA-m system from variant 1 is replaced by the matrix of formant bases and forms the set of randomly selected bases of different lengths and properties (simple and composite). So, our message is: S2 = 651.

- Decryption of this block gives: $\underbrace{54}_{base}\ \underbrace{12}_{core}\ \underbrace{3}_{remainder}$
- Restore the formant: $S2$=$54\times 12+3=651\rightarrow EDA$.

3. Analysis of the robustness of RSA-m algorithms depending on the variable parameters of the algorithm

These examples show that voice encryption occurs without the direct participation of subscribers in the key exchange process. The communication channel does not explicitly include formant parameters or RSA algorithm. In the author's patent [2] different algorithms are claimed, requiring the transfer of either all the parameters of the formants or separately, eventually forming the so-called. hybrid or composite cipher. [3 - 9].

If only cell indices are transmitted, where the information for recovering the formant is located, then breaking into such a cipher will require a continuous search of n samples (where n is the sampling frequency), which is estimated by a huge time value:

Indeed, with a sampling frequency of a voice signal of 8 kHz, an attacker to recover even a second conversation, which gives scant information about the essence of the calls and subscribers!, Will need to be completed (8 · ⟦10⟧ ^ 3)! operations matching the sequence of a discrete sequence for 1 sec, well, let it be in 10 sec (then declassifying a mobile conversation will go with a significant delay and go beyond the guaranteed secrecy period!). And this is only for one period, for 1 sec. And in order to recognize the voice, to make out the meaning of what has been said, it is necessary to process at least 10 periods (10 seconds of low-frequency periods of oscillations, that is, the envelope of the carrier frequency!). We consider that in order to do this with a continuous search (guessing or selecting speech by voice or by meaning are different time tasks!), You need to do the previous assessment at least 10 times, i.e. even at 1000! This is a very large number:

$$10\cdot\left((8\cdot 10^3)!\right)\gg 1000!=4{,}023872600770937735437024339 23\mathrm{E}+2567 \text{ sec} =$$

$$= 1{,}27596163139616239708175556165 34\mathrm{E}+2551\approx 10^{2551} \text{ billion years}$$

a huge number that is difficult to estimate immediately and goes beyond common sense in estimating real-time hacking, Considering that today's modern computers are capable of performing up to ⟦10⟧ ^ 6 multiplication operations per second, then performing so many comparisons of permutations will require even more low sampling rate of 1 kHz already ⟦10⟧ ^ 2551 billion years! Long have to wait. And with increasing n, Hz this time

exponentially grows. With an increase in the sampling rate of the voice signal from 8 to 12 kHz, the number of years for breaking will grow more than even with n = 1 kHz

1000! = 1,2759616313961623970817555616534E + 2560 years

Or

= 1.2759616313961623970817555616534E + 2551 billion years

When n = 12 kHz, this number will grow tens of thousands of times.

Such estimates we get about the robustness of the algorithms of FA, if we increase the sampling rate. But the FA allows introducing uncertainty when encrypting and decrypting if in the RSA crypto lock equation you enter an additional 2 parameters k and a.

As is known, the asymmetric RSA system uses the properties of one-sided functions for an integer argument that satisfy the conditions for the existence of a solution of one of the types of Diophantine equations with a parameter a = 1, see, for example, [1]. The advanced RSA algorithm (modernized “crypto lock” or RSA-m algorithm) is understood as the following diophantine equation linking open (e, n) and closed (p, q, d) RSA keys:

$$e \cdot d = k\varphi\,(n) + a = k \cdot \varphi\,(p) \cdot \varphi\,(q) + 1 = k(p-1)(q-1) + a;\ where\ a > 1, \quad (1)$$

When expression (1) is a crypto lock of the usual, classic RSA.

If a potential adversary (at least for a while) does not know about the use in (1) of the parameter a ≠ 1, i.e. the use of the advanced RSA-m algorithm, such an upgrade will ensure high RSA reliability even with a small key length.

If the transferred information blocks are small (short), and it is necessary to ensure high cryptographic strength (for which the long key is chosen), then in this case, instead of adding text to the empty bits, you can perform an exponentiation operation. The cryptographic stability of such a system will not be lower than that of the classical RSA with the same key length, and the a parameter, which is undefined for the enemy, will make decryption difficult. It should be noted that in the case of placing meaningful information in the M-blocks, finding a will not be easy, especially if the values of a vary from block to block!

The use of the advanced RSA algorithm introduces an additional indefinite parameter a to be determined, which potentially increases the “hacking” time and, in the case of encryption of short-term (secrecy) information, can serve as a means of improving the system's cryptographic strength (for example, during operational negotiations). The introduction of an additional indefinite parameter is possible and would increase the robustness if it were large, i.e. not "susceptible to" “Attack by brute force”, i.e. elementary selection, but since the proposed approach imposes certain restrictions on its size, it only provides a significant increase in durability with frequent changes of keys.

4 Advantages of the systems using RSA-m algorithm with quick key change

1. Encryption and decryption keys are constantly changing, and are randomly generated (or chosen from key matrix), what eliminates the possibility to predict the next pare of keys. For the period of time, required for the open key factorization, crypto-keys as well as the definition of the floating code will be changed many times. Moreover, in order to open the system cracker must know the module N and public key. He cannot determine

even the length (number of digits) of the number N. Which means that he doesn't know what number to factorize. That is why crypto resistibility *of such system is almost absolute*!

2. Main RSA system disadvantage - insufficient performance - eliminated by our algorithms. Considered system doesn't require providing the crypto resistibility for a very long time. It's enough to guarantee, that the key will not be decrypted for some time period, sufficient for the codes to be changed. In this case we can be satisfied with quite short keys, the generation of which will not take a long time.

3. The suggested system can be implemented in access control systems, based on different physical principles of "key" and "crypto-lock" interaction. It can be both contact systems and Off-contact systems, based on the exchange of radio signals, infra-red and optical interaction, etc.

4. One of the current trends in cryptography today is the development of new methods that will provide security, even if quantum computers (still fantastic) succeed in breaking down (cracking) traditional methods and crypto systems, such as RSA. The formant analysis (FA - a new direction in number theory) are an evolutionary way of developing the means of cryptographic information protection based on the modified RSA-m algorithm. This is explained by the fact that they allow the use of existing approaches to the protection of information against the background of the exponential growth of the computing power of rapidly developing so-called. quantum computers. Today, there is already a problem with the RSA algorithm — after some time, solving the problem of decomposing a prime number to factors will cease to be an unsolvable problem for the current level of computing power of computers. And it will happen in the near future (approximately in 15 years). Today, an increase in the computational complexity of the problem can already be solved, for example, by changing one and/or introducing new, additional, parameters into the proposed implementation of formant analysis algorithms for data encryption/decryption. The proposed implementation of our method is considered in relation to voice protection, where there are significant time constraints on the process of encrypting / decrypting data, on the one hand, and there is a requirement to achieve guaranteed cryptographic resistance, on the other hand.

5. Quantum computers will destroy (hack) the most popular public-key cryptographic systems, including RSA, DSA and ECDSA. But the next generation of cryptographic algorithms and systems that will resist attacks using quantum computers are already being developed: in particular, post-quantum encryption systems and public-key signature systems. The modified RSA-m answers these tasks, since its algorithms provide for the possibilities of unlimited growth of the computational complexity of the problem by changing the existing and introducing new parameters, which makes RSA-m more computationally complex compared to the RSA algorithm, which does not have such capabilities. At the same time, all mutable parameters are achievable and easily implemented within the existing hardware and do not lead to critical time delays of the parameters, namely, the encryption / decryption rate.

6. Inclusion in the process of encrypting additional cryptographic parameters associated with the use of formants, as well as the introduction of an additional parameter "a" in the crypto lock equation, complicates the task of hacking, requiring additional time for these operations, which is very long. With voice sampling rates from 4 to 12 kHz, this time exceeds all imaginable values of common sense! Therefore, such a crypto system with a combined and modernized RSA-m algorithm is not afraid of the impending crisis of

modern cryptography associated with the advent of fantastically high-speed quantum computers.

Conclusion

All the above examples highlight the ability to send encrypted voice messages in real time at a transfer rate of about 64 Kbps, for example, using any communication channel, or social networks like Telegram or Messenger

It is clear, that other variant of the formant use for message transmission can also exist, Ex., [3], where 10 different approaches are described. We considered the most obvious ones. For example, matrix M1 12x12 contains 64x64 = 4096 decimal numbers (digits), which will be encrypted with 64 (and more) different keys.

AB2 algorithm uses one and the same matrix, cell content of which remainds unchanged, but changes their address. The redistribution of cells content of matrix M can be done in different ways, for example, randomly or according to some algorithms, which will be applied to each matrix cell. It can be for example the ratio of $p_{ij} = p_{i+k,j+l}$, changing k and j in a such way, so that the algorithmwould affect all the cells, or just some of them.

In our library of algorithms there is and advanced AB-univ algorithm, that allows using message will be more difficult to decrypt in a really reasonable time. Even after writting a message on hard media, it could not be decrypted in a relatively short time, because diversant will require tens of years, because he doesn't know key length (range), block length, not to speak of the rule of transition from one pair of keys to another. Besides, he doesn't know keys at all. This is *internal information of security system*.Each external communcation can of one and the same type, but its' content will differ in the meaning of the current information in each session on communication and the same phonemes will be presented in messages by different codes.

References

1. Balabanov, A. A., Kunev, V.V. *Zashhishhyonny'e IT - sistemy' na osnove algoritmov formatnogo analiza. (Protected IT-systems based on the formant analyses algorithms)*. Lambert, Germany, 2016, 215 p., ISBN 978-3-659-94826-8.
2. Balabanov, A. A., Kunev V.V. *Sposob shhifrovaniya dvoichnoj informaczii i ustrojstvo dlya ego osushestvleniya. (Method of binary information encryption and application for its implementation)*: MD, Patent № a 2016 0046.
3. Balabanov, A. A., Agafonov, A.F., Riku V.A. Algoritm by*'stroj generaczii klyuchej v kriptograficeskoj sisteme* RSA (The algorithm of wick keys generation on the cryptographic RSA system), / VNTR N7_2009; / № 9 (37), 2010 r. http://www.vntr.ru/ftpgetfile.php?id=323 http://www.vntr.ru/ftpgetfile.php?id=451
4. Panasenko, S. P. *Algoritmy' shhifrovaniya. Speczial'ny'j spravochnik*. BXB - Peterburg, 2009, ISBN 978-5-9775-0319-8, str. 576.
5. Sajt o post kvantovoj kriptografii https://pqcrypto.org/
6. Bernstein, DanielJ., Buchmann, Johannes, Dahmen,Erik(Eds.): Post-Quantum Cryptography
7. Anderson R., Biham E. Two Practical and Provably Secure Block Ciphers: BEAR and LION// http://citeseer.ist.psu.edu — 1995.
8. Kelsey J.Re: Chainingciphers //http://cypherpunks.venona.com.
9. Moldovyan, A. A., Moldovyan, N. A. Eremeev, M. A. *Kriptografiya: ot primitivov k sintezu algoritmov*. BXB - Peterburg, 2004, ISBN 5-94157-524-6.