Olha KOROL[1], Alla HAVRYLOVA[2]

Supervisor: Olha KOROL[1]

MODELE MATEMATYCZNE HYBRYDOWYCH CRYPTO-CODÓW Z ZASTOSOWANIEM ALGORYTMU UMAC

Streszczenie: W obecnych czasach, istotną kwestią jest uwierzytelnienie krytycznych danych w systemach informacyjno-komunikacyjnych oraz w systemach cyber-fizycznych. Z jednej strony, obecne możliwości obliczniowe pozwalają na zwiększenie volumenu/liczby transmitowanych danych, natomiast – z drugiej strony, jest praktycznie niemożliwe, aby zapewnić stabilność mechanizmów uwierzytelniania. Badania w tym zakresie pokazują, ż jednym z obiecujących kierunków jest używanie specjalnych krypto-kodów w oparciu o schematy McEliece lub Niederreiter, które opierają się na kodach algebraiczno-geometrycznych oraz na algorytmach szybkiego uwierzytelniania. Autorzy proponują uzycie krypto-kodu McEliece w oparciu o zmodyfikowane kody eliptyczne wraz z uszkodzeniem w algorytmie caskadowym UMAC. Takie ujęcie problem umożliwia zachowanie wymaganego poziomu odporności (na błędy), efektywność generowania kodu MAC, a także universalności algorytmu UMAC, który umożliwia uwzględnienie własności kolizji jako dodatkowego identyfikatora w wielkich bazach danych.

Keywords: algorytm UMAC, McEliece hybrydowy krypto-kod, zmodyfikowane kody eliptyczne, algorytm MV2, uszkodzenie/awaria

MATHEMATICAL MODELS OF HYBRID CRYPTO-CODE CONSTRUCTIONS IN THE UMAC ALGORITHM

Summary: In the post-quantum period, the issue of data authentication in critical information-communication and cyber-physical systems becomes acute. On the one hand, computing resources provide the ability to increase the amount of transmitted data, on the other hand, it is practically impossible to ensure stability in authentication mechanisms. Research in this area demonstrates that one of the promising directions is the use of crypto-code constructions based on McEliece and Niederreiter schemes based on algebraic geometric codes, and fast authentication algorithms. The authors propose the use of McEliece crypto-code constructions based on modified elliptic codes with damage in the cascade UMAC algorithm. This approach provides the required levels of robustness, the efficiency of generating the MAC code, as well the universality property of the UMAC algorithm, which allows using collision properties as additional identifiers in large databases.

[1] Simon Kuznets Kharkiv National University of Economics, olha.korol@hneu.net

[2] Simon Kuznets Kharkiv National University of Economics

Key words: UMAC algorithm, McEliece hybrid crypto-code constructions, modified elliptic codes, MV2 algorithm, damage.

1. Introduction

In the post-quantum period, with the advent of a full-scale quantum computer and the rapid growth of computing resources, the contradiction between further consideration of the range of services based on Internet technologies and the impossibility of ensuring security is increasing. Basic security services such as confidentiality, integrity and authenticity are formed on the basis of symmetric and asymmetric systems, the stability of which in the post-quantum period is called into question. So, in March 2018 and February 2019 reports were published by NIST (National Institute of Standards and Technology) (USA) specialists, which confirm the possibility of breaking symmetric and asymmetric cryptography algorithms (including cryptoalgorithms and elliptic curve cryptography) based on Shor`s and Grover`s quantum cryptographic analysis algorithms.

In such conditions, an urgent problem is the search for new and modernization of known methods of providing security services. Among the known methods of ensuring authenticity, algorithms for generating MDC and MAC codes are distinguished. One of the promising algorithms for the formation of MAC-codes is the UMAC algorithm, which can provide the maximum speed of formation of hash codes and the property of universality. The property of universality guarantees an even distribution of hash codes between collisions and allows you to know this division in advance (the number of collisions). This property can be used as an additional identifier in large databases for fast information retrieval.

Thus, the proposed modernization of the UMAC cascade algorithm based on McEliece crypto-code constructions is an urgent scientific and technical problem.

For practical implementation, it is proposed to consider mathematical models for the formation of a hash code based on the UMAC algorithm, in which a code sequence of crypto-code constructions (CCC), hybrid crypto-code constructions of McEliece based on modified elliptic codes (MEC) (shortened and / or extended), and damage codes (DC).

2. Mathematical model for the formation of a hash code based on the UMAC cascade algorithm

2.1. Input data for the mathematical model of the hash code formation

The following input data are used to construct mathematical models for the formation of the hash code of the transmitted message and the pseudo-random substrate:

M – transmitted plain text;

I – plaintext information symbols (k-bit information vector over $GF(q)$) ;

K – the secret key;

$Taglen$ – an integer from the set of valid values $\{4, 8, 12, 16\}$, specifying the length of the message authenticity code Tag in bytes;

$Hash(K,M,Taglen)$ – key universal hashing function of an information message *M* using a secret key K ;

Y_{L1I} – generic hash-function value (UHASH-hash) of the first level hashing;

Y_{L3I} – hash-function value (Carter-Wegman-hash) of the third level hashing;

T – data block;

Blocklen – data block length (bytes);

Keylen – secret key length (32 bytes);

Tag – integrity and authenticity control code;

K_{L1I} – secret key of the first level of hashing, consisting of subkeys K_1, K_2, …, K_n;

K_{L3I} – secret key of the second level of hashing, consisting of keys K_{L31} (subkeys K_1, K_2, …, K_n) and K_{L32} (subkeys K_1, K_2, …, K_n);

Numbyte – pseudo-random key sequence length (number of subkeys);

K' – pseudo-random key sequence;

Index – subkey number;

$Wordbits \in [64, 128]$;

Maxwordrange – positive integer less then $2^{Wordbits}$;

k – key K_{L2} dependent an integer from a range $[0,..., prime(Wordbits)-1]$, $prime(x)$ – the largest prime number less than 2^x ;

$M_P = Y_{L1} = Hash_{L1}(K_{L1}, M)$ – data subject to polynomial hashing.

2.2. Mathematical model of the hash code formation

The UMAC algorithm consists of three levels (three hashing algorithms).

The first level of the hash code is generated by a specialized universal hashing function U-hash, which does the splitting of the array-string M dimension up to 2^{64} bytes for blocks M_i with 1024 byte with subsequent transformation of each block by the function $NH(K_{L1}, M_i)$:

$$Y_{L1} = Hash_{L1}(K_{L1}, M) = NH(K_{L1}, M_0) \| NH(K_{L1}, M_1) \| ... \| NH(K_{L1}, M_{n-1}),$$

where $n = \left\lceil \frac{Length(M)}{1024} \right\rceil$, $[x]$ – the integer part of number x, $Length(M)$ – byte length of information message of M length, K_{L1} represented as sequences of four-byte subblocks:

$$K_{L1} = K_{L1_1} \| K_{L1_2} \| ... \| K_{L1_t}.$$

Then (taking the initial state $Hash_{L1_i} = 0$) for all $j = 1, 9, 17..., t-7$ the following operations are performed:

$$Hash_{L1_i} = Hash_{L1_i} +_{64} ((M_{i_{j+0}} +_{32} K_{L1_{j+0}}) \times_{64} (M_{i_{j+4}} +_{32} K_{L1_{j+4}}))\,,$$

$$Hash_{L1_i} = Hash_{L1_i} +_{64} ((M_{i_{j+1}} +_{32} K_{L1_{j+1}}) \times_{64} (M_{i_{j+5}} +_{32} K_{L1_{j+5}}))\,,$$

$$Hash_{L1_i} = Hash_{L1_i} +_{64} ((M_{i_{j+2}} +_{32} K_{L1_{j+2}}) \times_{64} (M_{i_{j+6}} +_{32} K_{L1_{j+6}}))\,,$$

$$Hash_{L1_i} = Hash_{L1_i} +_{64} ((M_{i_{j+3}} +_{32} K_{L1_{j+3}}) \times_{64} (M_{i_{j+7}} +_{32} K_{L1_{j+7}}))\,,$$

where $+_{64}$, $+_{32}$ – modulo addition operations 2^{64} and 2^{32}, respectively; $\times_{64}$ – modulo operation 2^{64}.
As a result of calculations, an eight-byte value is formed $Hash_{L1_i}$

The second level of the hash code uses polynomial key transform $Poly$ based on the Carter-Werman polynomial hashing scheme. The result of the work of this level is to obtain a hash code:
$Y_{L2} = Hash_{L2}\left(K_{L2}, Y_{L1}\right) = Poly(Wordbits, Maxwordrange, k, M_P)$.

According to the UMAC algorithm specification as $prime(x)$ the following constants are used: $prime(36) = 2^{36} - 5$, $prime(64) = 2^{64} - 59$, $prime(128) = 2^{128} - 159$. Bit length M_P denoted as $Bytelength(M_P)$. Depending on the length M_P the following features are used in the implementation of the second level of hashing:

– if the length of the received data M_P does not exceed 2^{17} byte then polynomial hashing $Poly$ executed with parameters $Wordbits = 64$; $Maxwordrange = 2^{64} - 2^{32}$; $k = k64$ – the string formed by the first eight bytes of the key K_{L2} and a special eight-byte mask;

– if the length of the received data M_P exceeds 2^{17} bytes (but does not exceed 2^{64} bytes), then first 2^{17} data bytes are processed by the polynomial hashing function $Poly(64, 2^{64} - 2^{32}, k64, M_P)$, and the remaining data bytes are processed by the function $Poly$ with parameters $Wordbits = 128$; $Maxwordrange = 2^{128} - 2^{96}$; $k = k128$ – the string formed by the last 16 bytes of the key K_{L2} and a special 16 byte mask.

Hashed data M_P split into blocks by $Wordbytes = Wordbits / 8$ bytes:

$$M_P = M_{P_1} \| M_{P_2} \| \ldots \| M_{P_n}\,,$$

where $n = Bytelength(M_P) / Wordbytes$.

The result of hashing is the value of the polynomial function

$$Y_{L2} = \left(M_{P_n} + kM_{P_{n-1}} + \ldots + k^{n-1}M_{P_1} + k^n\right)\mathrm{mod}(p)\,,$$

which is calculated by the iterative procedure (for all $i = 1, 2, \ldots, n$):

$$Poly_i = \left(kPoly_{i-1} + M_{P_i}\right)\mathrm{mod}(p),\ Poly_0 = 1,\ p = prime(Wordbits)$$

using Horner's scheme:

$$M_{P_n} + kM_{P_{n-1}} + ... + k^{n-1}M_{P_1} + k^n = (((k + M_{P_1})k + M_{P_2})k + ... + M_{P_{n-1}})k + M_{P_n}.$$

Computed hash value $Y_{L2} = Poly_n$ is an integer from the range $[0,..., prime(Wordbits) - 1]$.

The third level of the hash code $Hash_{L3}\left(K_{L3_1}, K_{L3_2}, Y_{L2}\right)$ is performed on the result of polynomial hashing and converts data of length up to 16 bytes supplied to its input into a hash code Y of fixed length 32 bits. This level is comparable to a symmetric stream cipher, where the hash code is eventually added to a pseudo-random substrate, which ensures the strength of the MAC code.

The initial data of the third level of hashing are two key sequences K_{L3_1} and K_{L3_2} lengths of 64 and 4 bytes, respectively, as well as an input 16 byte sequence Y_{L2}

Hashed data Y_{L2} and the key sequence K_{L3_1} evenly split into eight blocks, each of which is represented as an integer Y_{L2_i} and $K_{L3_{1i}}$, $i = 1, 2,...,8$.

Hash value Y_{L3} calculated as follows:

$$Y_{L3} = \left(\left(\left(\sum_{i=1}^{m} Y_{L2_i} K_{L3_{1i}}\right)\mathrm{mod}(prime(36))\right)\mathrm{mod}(2^{32})\right) xor(K_{L3_2}),$$

However, in [1] it is shown that when the symmetric AES algorithm is used to form the substrate, the universality property is “removed”. Therefore, the authors propose an improvement of the algorithm by using McEliece on MEC and DC to construct the substrate CCC (GCCC). This approach not only preserves the property of universality, but also ensures the stability of the MAC code in the post-quantum period. Let us consider a mathematical model of the formation of a base on the basis of McEliece GCCC using MES. This approach allows you to reduce energy costs for implementation and provide an additional level of security, due to the initialization vectors, which define the symbols of the reduction and / or lengthening of the codeword. The use of defective cryptography reduces capacitive costs by 12-15 times (Galois field 2^4, instead of 2^{10}–2^{13}, as required for the complete McEliece scheme).

3. Mathematical model of the formation of a pseudo-random substrate *Pad* based on the McEliece GCCC

3.1. Input data for the mathematical model for the formation of a pseudo-random substrate *Pad*

The mathematical model of McEliece GCCC on modified elliptic codes (MEC) based on shortening (reduction of information symbols) or lengthening (adding information symbols) with damage is formally set by a set of input elements [2–4], given below.

Plain text *M*, consisting of information symbols *I*, at that $\forall \mathrm{I}_\mathrm{j} \in GF(q)$:

– when shortening characters:

$$M_i = \{I_0, I_{h_1}, .. I_{h_j}, I_{k\text{-}1}\},$$

where h_j – information symbols equal to zero

– when lengthening characters:

$$M_i = \{I_0, I_{h_{r1}}, .. I_{h_{rj}}, I_{k\text{-}1}\},$$

where h_r – extension information symbols k

$|\mathrm{h}| = \frac{1}{2}k$, т. е. $\mathrm{I_i} = 0$, $\forall \mathrm{I_i} \in \mathrm{h}$,

A plurality of closed texts (codograms):

$$C = \{C_1, C_2, ..., C_{q^k}\}, \text{где } \forall c^*_{X_j} \in GF(q)$$

– when shortening characters:

$$C_i = (c^*_{X_0}, c^*_{h_1}, ..., c^*_{h_j}, c^*_{X_{n-1}})$$

– when lengthening characters:

$$C_i = (c^*_{X_0}, c^*_{h_{r1}}, ..., c^*_{h_{rj}}, c^*_{X_{n-1}})$$

A plurality of direct mappings (based on the use of public key - generating matrix):

$$\phi = \{\phi_1, \phi_2, ..., \phi_s\}$$

– when shortening characters:

$$\phi_i : M \to C_{k-h_j}, i = 1, 2, ..., s$$

– when lengthening characters:

$$\varphi_i : M \to C_{h_r},\ i = 1, 2, ..., s,$$

A plurality of reverse mappings (based on the use of a private (private) key - masking matrices):

$$\phi^{-1} = \{\phi_1^{-1}, \phi_2^{-1}, ..., \phi_s^{-1}\}$$

– when shortening characters

where $\phi_i^{-1} : C_{k-h_j} \to M$, $i = 1, 2, ..., s$

– when lengthening characters

where $\varphi_i^{-1} : C_{h_r} \to M$, $i = 1, 2, ..., s$

A plurality of keys, parametrizing direct mapping (the public key of the authorized user):

$$K_{a_i} = \{K_{1_{a_i}}, K_{2_{a_i}}, ..., K_{s_{a_i}}\} = \{G_X^{EC_1}{}_{a_i}, G_X^{EC_2}{}_{a_i}, ..., G_X^{ECs}{}_{a_i}\},$$

where $G_X^{ECi}{}_{a_i}$ – generating $n \times k$ matrix of an algebrogeometric block disguised as a random $(n,\ k,\ d)$ - code with elements from $GF(q)$,

a_i – the set of coefficients of a curve polynomial $a_1 ... a_6$, $\forall a_i \in GF(q)$, specifying a specific set of points on a curve from space P^2

– when shortening characters:

$$\phi_i : M \xrightarrow{K_{ia_i}} C_{k-h_j};\ \ i = 1, 2, ..., s$$

– when lengthening characters:

$$\varphi_i : M \xrightarrow{K_{ia_i}} C_{h_r};\ i = 1, 2, ..., s$$

A plurality of keys that parameterize the inverse mappings (private (private) key of the authorized user):

$$K^* = \{K_1^*, K_2^*, ..., K_s^*\} = \{\{X, P, D\}_1, \{X, P, D\}_2, ..., \{X, P, D\}_s\},$$

$$\{X, P, D\}_i = \{X^i, P^i, D^i\},$$

where X^i – masking non-degenerate randomly equiprobably generated by the key source $k \times k$ matrix with elements from $GF(q)$;

P^i – permutation randomly equally likely generated by the key source $n \times n$ matrix with elements from $GF(q)$;

D^i – diagonal generated by key source $n \times n$ matrix with elements from $GF(q)$, i.e. $\phi_i^{-1} : C \xrightarrow{\mathrm{K}_i^*} M$, $i = 1, 2, ..., s$.

A plurality of defective texts *CFT:*

$$CFT = \{CFT_1, CFT_2, ..., CFT_{q^k}\}$$

A plurality of damage *CHD:*

$$CHD = \{CHD_1, CHD_2, ..., CHD_{q^k}\}$$

A plurality of direct damage (through the use of key – K^i_{MV2}, and algorithm *MV*2):

$$E = \{E^1_{K_{MV2}}, E^2_{K_{MV2}}, ..., \phi^S_{K_{MV2}}\}, \quad i = 1, 2, ..., s;$$

$f(x)_i$ – flag (damage, *CHD*), $C(x)_i$ – remainder (defective text, *CFT*);
$f(x)=n-|C(x)|$, if $|C(x)|>y$, where y – some parameter, $y \in_Y Z_{q^m}$, $0 \langle y \langle n$

A plurality of mappings *MV2* F_n^r:
– is given by a bijective mapping between the set of permutations $\{S_1, S_2, ..., S_{2^n}\}$ and plurality $\# F_n^r$, $\# F_n^r = \#\{(c, f)\} = 2^n!$;

A plurality of meaningful text (based on the use of the keya – K^i_{MV2}, и алгоритма *MV*2):

$$E^{-1} = \{E^1_{K_{MV2}}{}^{-1}, E^2_{K_{MV2}}{}^{-1}, ..., E^S_{K_{MV2}}{}^{-1}\},$$

where $E^{-1}_{K_{MV2}} : \|f(x)_i\| + \|C(x)_i\| \to M$, $i = 1, 2, ..., s$;
$f(x)_i$ – flag (damage, *CHD*), $C(x)_i$ – remainder (defective text, *CFT*);
$f(x)=n-|C(x)|$, if $|C(x)|> y$, where y – some parameter, $y \in_Y Z_{q^m}$

A plurality of key conversion codes flawed:

$$K^i_{MV2} \in K_{MV2}$$

Algebrogeometric block *(n, k, d)*-code C_{k-h_j} (shortened) / C_{h_r} (lengthened) above *GF(q),* i.e. a set of code words $C_i \in C_{k-h_j}$ (when shortening) / $C_i \in C_{h_r}$ (when lengthening), that the condition is satisfied $C_i H^T = 0$, where H – parity check matrix of an algebraic geometric block code;

a_i – set defines a specific set of curve points from space P^2 to form the generating matrix;

h_j – information symbols equal to zero, $/h/=1/2k$, т.е. $I_i = 0,\ \forall I_i \in h$;

h_r – Information lengthening symbols k, $/h/=1/2k$, т.е. $I_i = 0,\ \forall I_i \in h$;

Masking matrix mappings, given by a set of matrices $\{X, P, D\}_i$, where X – non-degenerate $k \times k$ matrix over $GF(q)$; P – permutation $n \times n$ matrix over $GF(q)$ with one nonzero element in each row and in each column of the matrix; D – diagonal $n \times n$ matrix over $GF(q)$ with nonzero elements on the main diagonal;

y – some parameter $y \in_Y Z_{q^m}, Z_{q^m} = \{0,1,...2^n - 1\}$;

n – some parameter $n \in_Y Z_{q^n}, Z_{q^n} = \{1,...2^n\}$;

a plurality of mappings $MV2\ F_n^r$.

3.2. Mathematical model of the formation of a pseudo-random substrate *Pad*

3.2.1. Formation of *Pad* as CCC on MEC

Let us consider the formation of a pseudo-random substrate *Pad* as a McEliece CCC with the possibility of modifications (shortening or lengthening).

This modified (shortened / extended) algebraic geometric *(n, k, d)*-code C_{k-h_j} (shortening) / C_{h_r} (lengthening) with a fast decoding algorithm disguises itself as a random *(n, k, d)-code* C_{k-h_j} * (shortening) / C_{h_r} * (lengthening) by multiplying the generating matrix G^{EC}-code C_{k-h_j} (shortening) / C_{h_r} (lengthening) on the masking matrix, which are kept in secret X^u, P^u and D^u [2–4], which ensures the formation of the public key of the authorized user:

$$G_X^{ECu} = X^u \cdot G^{EC} \cdot P^u \cdot D^u, \quad u \in \{1,2,...,s\},$$

where G^{EC} – generating $n \times k$ matrix of algebrogeometric block (n, k, d)-code with elements from $GF(q)$, built using user-selected coefficients of the curve polynomial $a_1 \dots a_6$, $\forall a_i \in GF(q)$, which uniquely define a specific set of points of a curve from space P^2.

Formation of closed text $C_j \in C_{k-h_j}$ (shortening) / $C_j \in C_{h_r}$ (lengthening) by the entered plain text M and the given public key $G_X^{ECu}{}_{a_i}$, $u \in \{1,2,...,s\}$ is carried out by forming a codeword of a masked code with the addition of a randomly generated vector $e = (e_0, e_1, ..., e_{n-1})$: $C_j = \phi_u\left(M_i, G_X^u\right) = M_i \times \left(G_X^u\right)^T + e$, moreover, the Hamming weight (the number of nonzero elements) of the vector e does not exceed the correcting ability of the used algebraic block code:

$$0 \le w(e) \le t = \left\lfloor \frac{d-1}{2} \right\rfloor, \ \lfloor x \rfloor \text{ – integer part of a real number } x.$$

For each closed text that is generated $C_j \in C_{k-h_j}$ (shortening) / $C_j \in C_{h_r}$ (lengthening), corresponding vector $e=(e_0,e_1,...,e_{n-1})$ is a one-time session key, that is, for a specific E_j vector e generated randomly, equally probable and independently of other closed texts.
To algorithm *MV*2 fed (shortening / lengthening):

$$C_j^* = C_j - C_{k-h_j},\ E_{K_{MV2}} : C_j^* \to \left\|f(x)_i\right\| + \left\|C(x)_i\right\|$$
$$C_j^* = C_{h_r},\ E_{K_{MV2}} : C_j^* \to \left\|f(x)_i\right\| + \left\|C(x)_i\right\| .$$

To the communication channel $\left\|f(x)_i\right\|$ и $\left\|C(x)_i\right\|$ wherein the transmission can be carried out on one or on two independent channels.

3.2.2. Formation of *Pad* in the form of the GKKK on the MEC with damage

Taking into account the modifications obtained during the formation of the codogram in clause 3.2.1, damage is carried out according to the following scheme:

1. Let's form a subset of points *h(GF(q))*: *$(P_{x1}, P_{x2}, ..., P_{xx})$, $h \subseteq EC(GF(q))$, $/h/=x$* and keep it secret.
   The initial data for this is the final field *GF(q)*, elliptical curve $y^2z+a_1xyz+a_3yz^2=x^3+a_2x^2z+a_4xz+a_6z^3$, as well as a set of its points *EC(GF(q)):* *$(P_1, P_2, ..., P_N)$* over *GF(q)*.

2. Let's form the initialization vector:

when shortening characters	when lengthening characters
$IV_1 = EC - h_j$	$IV_1,\ IV_2 = EC - h_r$

3. Let's form on the entered information vector *I* the codeword *c*. Если *(n, k, d)*-code over *GF(q)* is given by its generating matrix, then $c=I\cdot G$.

4. Let's generate a random error vector *e* such that $w(e) \le t$, $t = \lfloor (d-1)/2 \rfloor$. Add the generated vector to the codeword, we get the codeword: $c^*=c+e$.

5. Let's form a codogram by:

- when shortening characters:	- when lengthening characters:
adding (lengthening) initialization vector characters: $c_X^*=c^*+IV_1$;	remove (shorten) the initialization vector characters: $c_X^*=c^*-IV_2$.

6. We will form a damaged text (remainder) and a flag (damage):

$$C_j^* = C_j - C_{k-h_j},\ E_{K_{MV2}} : C_j^* \to \left\|f(x)_i\right\| + \left\|C(x)_i\right\|$$
$$C_j^* = C_{h_r},\ E_{K_{MV2}} : C_j^* \to \left\|f(x)_i\right\| + \left\|C(x)_i\right\|$$

Thus, it is proposed to use a sequence of defective text as a background, which ensures the preservation of the universality property, the required level of reliability and, unlike the proposals for modifying the UMAC algorithm in [5], provide the required level of efficiency.

4. Conclusions

Generation of authentication codes for transmitted messages, which are hash codes generated by the UMAC algorithm and a pseudo-random backing *Pad*, formed on the basis of the McEliece GCCC on the MEC with damage, in the form of mathematical models, provides the required level of MAC code stability, the efficiency of its formation and retains the property of universality.

REFERENCES

1. KUZNETSOV A.A.: Investigation of collisional properties of UMAC message authentication codes, A. A. Kuznetsov, O. G. Korol, S. P. Evseev, Applied Radioelectronics. Publishing house of KNUR, 11(2012)2. 171–183.
2. YEVSEIEV S., KOROL O., KOTS H.: Construction of hybrid security systems based on the crypto-code structures and flawed codes, Eastern-European Journal of Enterprise Technologies, 4/9(2017)88, 4 – 20.
3. YEVSEIEV S. and other: Practical implementation of the Niederreiter modified crypto-code system on truncated elliptic codes, Eastern-European Journal of Enterprise Technologies, 6/4(2018)96, 24 –31.
4. YEVSEIEV S., BAKIROVA L., SUSHCHENKO M.: Mathematical models of hybrid crypto code constructions on damaged codes, Сучасні інформаційні системи. 3(2019)3, 87 – 103.
5. KOROL O.G.: Investigation of properties of modular transformations and methods of information hashing on their basis, O. G. Korol, L. T. Parkhuts, S. P. Evseev, Systems of information processing, 4(2013)111, 106–110.