МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ
ХАРЬКОВСКИЙ НАЦИОНАЛЬНЫЙ
УНИВЕРСИТЕТ РАДИОЭЛЕКТРОНИКИ

РАДИОТЕХНИКА

Всеукраинский межведомственный научно-технический сборник

ТЕМАТИЧЕСКИЙ ВЫПУСК «ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ»

Основан в 1965 г.

В Ы П У С К 1 9 8

Харків
Харківський національний
університет радіоелектроніки
2019

УДК 621.3

Сборник включен в список специальных изданий ВАК Украины по физико-математическим и техническим наукам.
Регистрационное свидетельство КВ № 12098-969 ПР от 14. 12. 2006.
Ответственность за содержание статей несут авторы.

Ответственный за выпуск: И.Д. Горбенко, д-р техн. наук, проф.
Технический секретарь Е.С. Полякова

Рекомендовано Ученым советом Харьковского национального университета радиоэлектроники, протокол № 9 от 24.10.2019.

Адрес редакционной коллегии: Харьковский национальный университет радиоэлектроники (ХНУРЭ), просп. Науки, 14, Харьков, 61166, тел. (0572) 7021-397.

Сборник «Радиотехника» включен в Каталог подписных изданий Украины, подписной индекс ***08391***

СОДЕРЖАНИЕ ЗМІСТ

ПЕРСПЕКТИВНЫЕ КРИПТОГРАФИЧЕСКИЕ ПРЕОБРАЗОВАНИЯ И ИХ ПРИМЕНЕНИЕ

ПЕРСПЕКТИВНІ КРИПТОГРАФІЧНІ ПЕРЕТВОРЕННЯ ТА ЇХ ЗАСТОСУВАННЯ

АНАЛИЗ И ИСПОЛЬЗОВАНИЕ КРИПТОГРАФИЧЕСКИХ МЕТОДОВ В ДЕЦЕНТРАЛИЗОВАННЫХ ТЕХНОЛОГИЯХ

АНАЛІЗ ТА ЗАСТОСУВАННЯ КРИПТОГРАФІЧНИХ МЕТОДІВ В ДЕЦЕНТРАЛІЗОВАНИХ ТЕХНОЛОГІЯХ

CONTENT

PERSPECTIVE CRYPTOGRAPHIC TRANSFORMATIONS AND THEIR APPLICATION

ANALYSIS AND USE OF CRYPTOGRAPHIC METHODS IN DECENTRALIZED TECHNOLOGIES

ПЕРСПЕКТИВНЫЕ КРИПТОГРАФИЧЕСКИЕ ПРЕОБРАЗОВАНИЯ И ИХ ПРИМЕНЕНИЕ

ПЕРСПЕКТИВНІ КРИПТОГРАФІЧНІ ПЕРЕТВОРЕННЯ ТА ЇХ ЗАСТОСУВАННЯ

UDC 004.056.55 **DOI:10.30837/rt.2019.3.198.01**

I.D. GORBENKO, Dr. Sc. (Technology), O.G.KACHKO, Cand. Sc. (Technology), A.N. ALEKSIYCHUK, Dr. Sc. (Technology), O.O. KUZNETSOV, Dr. Sc. (Technology), YU.I. GORBENKO, Cand. Sc. (Technology), V.V. ONOPRIENKO, Cand. Sc. (Technology), M.V. YESINA, Cand. Sc. (Technology), S.O. Candiy

ALGORITHMS OF ASYMMETRIC ENCRYPTION AND ENCAPSULATION OF KEYS OF POST-QUANTUM PERIOD OF 5 -7 LEVELS OF STABILITY AND THEIR APPLICATIONS

Introduction

At present, significant efforts are being made by the cryptographic community to create practical quantum-stable mechanisms of asymmetric encryption (ASE), key encapsulation protocols (KEP) and electronic signature (ES) [1 – 7]. The results of the implementation of the 1st stage of the international competition for creation of post-quantum ASE, KEP and ES [1], performed by NIST USA, as well is performed comparisons of alternatives [2, 3], make it possible to conclude about the prospects of application of cryptographic transformations in rings of polynomials (algebraic lattices) for their creation. Such transformations have stood the test of cryptographic stability in the form of the NTRU cryptosystem [1]. In general, NTRUEncrypt ANSI X9.98 [4], NTRU Prime [5], and NTRU Prime Ukraine [5] are examples of implementations for the ASE and KEPs of cryptographic transformations on algebraic lattices. Mechanisms for constructing the ASE and KEPs are proposed in [2], their application makes it possible to provide the 5^{th} level of cryptographic stability inclusive, i.e. 128 bits of quantum and 256 bits of classical crypto-stability. But in our view, the problem of providing encryption and encapsulation is important both from theoretical and practical point of view, including up to the 7^{th} level of stability, since the 5^{th} level of cryptographic stability is not enough for the quantum period [7 – 9]. That is, the current problem is the problem of creation and standardization of the ASE and KEP algorithms of 256 bit quantum and 512 bit classical crypto-stability [7, 5]. Moreover, in accordance with the requirements, the draft standard of the ASE and KEP, which is considered as the draft national standard of Ukraine officially [8], should provide different modes of operation: *asymmetric encryption; asymmetric encapsulation of keys; asymmetric encryption and encapsulation of keys; asymmetric encryption, encapsulation of keys and key generation for symmetric encryption, ensuring their crypto survivability in the form of "direct secrecy"* [11 – 13]. That is, nowadays, the problem of creating and standardizing post-quantum ASE and KEP algorithms of 128, 192 and 256 bits of quantum, as well as 256, 384 and 512 bits of classical crypto-stability [9 – 11] for valid and selected security models is important.

The purpose of this article is to present and review the constructed algorithms of asymmetric encryption and encapsulation of keys in polynomial rings (algebraic lattices), analysis of the essence of the cryptographic transformations of the used ASE and KEPs. The said purpose, in our view, is achieved by outlining the following problems:

- calculation of general and additional parameters for crypto transformations of the 5-7 levels of cryptographic stability;
- generation of asymmetric key pairs of encryption keys and encapsulation of keys for crypto transformations of the 5-7 levels of cryptographic stability;

- development of algorithms of asymmetric encryption (encryption and decryption) for crypto transformations of the 5-7 levels of cryptographic stability;

- development of algorithms of asymmetric encapsulation and decapsulation of keys for crypto-transformation of the 5-7 levels of cryptographic stability;

- development of proposals (algorithms) for calculating secure session keys for their use in symmetric encryption of data in communication channels;

- estimation of complexity of forward and reverse transformations at asymmetric transformations (encryption and encapsulation).

Questions of estimation of cryptographic stability and justification of the ASE and KEP mechanism parameters are given in [6 – 9].

In the following, we will consider the mechanisms of encryption and encapsulation with three sets of parameters that determine stability, marking them as follows: SKELYA-KEM 256/128; SKELYA-KEM 384/192 and SKELYA-KEM 512/256 [9].

1. Parameters of cryptographic transformations of keys encryption and encapsulation

According to [7, 8, 10], the parameters of cryptographic transformations are divided into basic (general), additional and mechanism parameters.

First and foremost, general parameters are defined to ensure the specified crypto stability as well as to ensure the success of operations (such as the absence of decryption errors), and to reduce computational complexity.

Only general parameters are used to calculate additional parameters. Re-calculating of additional parameters makes it possible to reduce the computational complexity of basic cryptographic transformation operations.

Algorithm parameters are parameters that need to be agreed to share encryption and encapsulation algorithms. Now it concerns identifiers of algorithms, constant messages and the like.

Detailed data on generating general, additional and mechanism parameters are given in [7].

Table 1 lists general parameters of SKELYA algorithms of keys encryption and encapsulation. The justification and calculation of the general parameters are given in [7, 9].

Table 1

General parameters of encryption and encapsulation of SKELYA algorithms keys

SKELYA-KEM 256/128			SKELYA-KEM 384/192			SKELYA-KEM 512/256		
n	*t*	*Q*	*n*	*t*	*q*	*n*	*t*	*q*
881	159	7673	1201	192	9221	1471	255	12251

The following notations are used in Table 1:

n – the degree of the polynomial;

t – the number of nonzero elements in *t* – small polynomial;

q – a large module, a simple number that is relatively simple with a polynomial

$x^n - x - 1$, and the value of *q* is determined by the condition of guaranteeing no decryption error;

p – a small module, $p = 3$.

Table 2 lists the additional parameters of SKELYA algorithms of keys encryption and encapsulation. The justification and calculation of the general parameters are given in [7, 10].

Table 2

Additional parameters of encryption and encapsulation algorithms of SKELYA algorithms keys

Sign	Purpose	Formula or value
qBits	The number of bits in *q* given as a binary string	$qBits = \lceil \log_2 q \rceil$
db	Length *b* (bit)	db= λ
bufferLenBytes	The length of the octet string for the conversion functions between the small polynomial and the octet string (code2to3, code3to2 functions)	*Ceil* (((n-1)/2)*3/8)
maxMsgLenBytes	Maximum message length for encryption (octets)	*(bufferLenBytes–λ/8)–1*
EncMsgLenBytes	The length of the encrypted message (octets)	*Ceil(qBits * n/8)*
cBits	The number of bits for specifying the degree of a polynomial as a binary string	cBits = Ceil($\log_2$n) + 1
Llen	The number of octets to specify the length of the encrypted message	1
pkLen	The number of bits *h* that are used during encryption	pkLen=db

Ceil function in Table 2 defines the smallest integer that is not less than the input argument.

The parameters of the encryption and encapsulation algorithms of SKELA algorithm algorithms are shown in Table 3. The justification and calculation of the general parameters are given in [7, 10].

Table 3

The parameters of the encryption and encapsulation algorithms of the keys and their values for the test version

Sign	Purpose	Formula or value
OID	Method identifier (3 octets)	For the test version OID[0]=0, OID[1]=0, OID[2]=1
m_kem	Permanent message used for encryption in the key encapsulation and decapsulation protocol	A row of octets is used for the test version { 'I' ⊕ 0xFF, 'I' ⊕ 0xFF, 'T' ⊕ 0xFF, ' ' ⊕ 0xFF, 'E' ⊕ 0xFF, 'n' ⊕ 0xFF, 'c' ⊕ 0xFF, 'a' ⊕ 0xFF, 'p' ⊕ 0xFF, 's' ⊕ 0xFF, 'u' ⊕ 0xFF, 'l' ⊕ 0xFF, 'a' ⊕ 0xFF, 't' ⊕ 0xFF, 'e' ⊕ 0xFF, 0 ⊕ 0xFF}
m_kemBytes	Length m_kem (octets)	*m_kemBytes* ≤ *maxMsgLenBytes* *m_kemBytes = 16 for m_kem*

2. Generation of key pairs of asymmetric cryptographic transformations

When generating a specific key pair – private key f and public key h the polynomials G, F are used, whose degree is *n,* and coefficients modulo *p (p = 3)*, that is, they take values 0, -1,1. The result is simultaneously calculated private key f and public key h (polynomials) and h (byte task h). Generation is carried out in such sequence.

Generation of Polynomial G. Polynomial G is a polynomial having (2n + 1) / 3 of nonzero coefficients.

Generation of polynomial F. Polynomial F is a polynomial having 2t nonzero coefficients.

Calculation of the private key f is carried out according to the formula [9]

$$f = pF + 1 \tag{1}$$

Calculation of the public key h is carried out to the formula [9]

$$h = p * G * f^{-1} \text{ в полі } (Z / q[x] / (x^n - x - 1)) \tag{2}$$

To calculate the public key, it is necessary to calculate f^{-1} in the field $(Z / q[x] / (x^n - x - 1))$ and multiply the polynomials in the same field.

Calculation of inverse element

To calculate the inverse element, an advanced Euclidean algorithm for polynomials is used, i.e., the equation:

$$ax + by = d \tag{3}$$

Moreover, in equation (3) it is necessary to set *a* as polynomial f, *b* – as polynomial $x^n - x - 1$. The value of the right-hand side of equation (3), i.e. (d), determines the greatest common divisor for a and b (GCD). All the calculations must be performed modulo *q*. Computed values *d, x, y* can be the result of using an advanced algorithm. If the degree of the polynomial *d* is 0, it is a guarantee for the presence of inverse element since *d* is an integer. In our case $||f||_1 \neq 0$, so the inverse element f^{-1} *mod* $(x^n - x - 1)$ exists, and the value of the variable *x* is the inverse element.

The number of steps that need to be done depends only on the degree of the polynomials, so the completion time of the inversion calculation operation is independent of the specific key

Multiplication of polynomials

The polynomial multiplication operation is defined mathematically as:

$$c(X) = a(X) * b(X), \tag{4}$$

where

$$c'_k = \sum_{\substack{i+j=k \\ i \in 0...n-1 \\ j \in 0...n-1}} (a_i * b_j) \bmod q; \quad c(X) = c'(X) \bmod (x^n - x - 1).$$

The specificity of polynomials used for multiplication is the presence of a large number of null elements in one of the polynomials (n/3 or even more if you use this operation when encrypting). In the implementation of this operation it is necessary to solve 2 problematic tasks:

- providing the least computational complexity;
- ensure the independence of the execution time of a particular key.

Many works are devoted to solving these problems [5 – 11].

3. Algorithms of asymmetric encryption and decryption

This section is devoted to discussion of the asymmetric encryption (encryption and decryption) Skelya algorithms.

When developing encryption algorithms, we have taken into account the NTRU encryption algorithm [4], which has been successfully used for almost 10 years. That is why most of the designations coincide with the designations adopted in [5]. But the algorithm [5] has a significant drawback associated with the possibility of decryption error, which was taken into account when developing a new Skelya algorithm [10].

In addition, the modern practice of using as a module the prime number q instead of the number 2^k and the polynomial $x^n - x - 1$ instead of the polynomial $x^n - 1$ is taken into account, which provides protection against known attacks [5, 10]. That is why the field $(Z / q[x] / (x^n - x - 1))$ is used as the field, as for NTRU Prime [5], but the parameters are

calculated taking into account the required levels of crypto-stability λ = {256, 384, 512} and ensuring no decryption errors [5, 10].

3.1. Encryption algorithm

Input:

- a string for encryption (m);
- the length of the string m (mLen);
- the recipient's public key h (polynomial R/q) and the corresponding octet string (h)
- Output:
- a sign of success *Success* (OK, ERROR);
- Encrypted string E in case of successful operation.

The length of the encryption line is limited by the fact that this line is built with additions that will provide semantic security, it can be specified by a polynomial of degree *n*. If the length of the string exceeds this value, the encryption operation returns an error. The maximum length of the valid message (octets), depending on the level of crypto-stability (cryptographic strength) λ (degree of polynomial n), is given in Table.4.

Table 4

The maximum length of the message, depending on crypto-stability (cryptographic strength) λ

λ	256 (n= 881)	384 (n=1201)	512 (n = 1471)
EncMsgLenBytes (octets)	132	176	210

Algorithm for encryption is an iterative algorithm, that continues until a small polynomial is formed, which is used to mask the encrypted message, until it satisfies the conditions:

- number (units) + number (minus) units not less than *2t*
- number (zeros) is not less than *t*.

The masking polynomial is formed on the basis of random components, so it is highly probable that the polynomial coefficients are equally probable. If *t* is significantly less than n/3 the probability of fulfilling this condition is high and the algorithm usually does not require a return for recalculations.

Iteration algorithm

1 An octet line of bufferLenBytes length is formed, where they write:

- random string of λ bits, denoted by *b* (provides semantic security);
- encryption string to which its length is transmitted;;
- zero octets (to complement the required length)

Denote this string *M*. This line depends on the random sreing and the incoming message and has a constant length that does not depend on the length of the message.

2 The octet string *M* is converted to R/3 by a polynomial, for which every 3 bits of the string are converted to two polynomial coefficients according to Table 5.

Table 5

Convertion of bit line into polynomial R/3 and vice versa

Bit string	Polynomial coefficients	Bit string	Polynomial coefficients
000	0, 0	100	1, 1
001	0, 1	101	1, -1
010	0, -1	110	-1, 0
011	1, 0	111	-1, 1

As a result of the transformation the corresponding polynomial *MTrin* is obtained. This polynomial, like string *M*, depends on random data and the encrypted string. It should be noted that *MTrin* is uniquely determined by string *M*, and conversely, string *M* can be restored by *MTrin*

3 An octet string is formed, where they write:

- method identifier (3 octets) that can be selected by agreement of the parties;
- message for encryption;
- random string b;
- a part of the public key h of the *pkLen* bits length

Let's mark this string as S. Let's note that string S has a variable length that depends on the length of the encryption string. String S depends not only on the string for encryption and the random string, but also on the identifier's algorithm and the recipient's public key.

4 Let us transform string S into a dazzling polynomial *r*. The dazzling polynomial is also a polynomial of degree *n* having *2t* nonzero coefficients, the other coefficients being 0. Let us determine the length of the bit string required to define the dazzling polynomial. We will define separately the signs of nonzero elements and their indices. All that should be defined is *2t* characters and *2t* indices. Let us apply *2t* bits to specify non-zero element characters and *2t* indices of these elements. To set each index it is enough to have *cBit* bits. Given the possibility of obtaining an index that has already been used, to set the indices, we form a string twice as long as necessary. Thus, the total length of the string, which should be formed is *2t + 4t * cBit*. In general, the transformation is carried out as follows:

- the pseudorandom data generator is initialized with string S;
- pseudo-random bytes of desired length are generated ();
- signs of non-zero elements are defined;
- indices of non-zero elements are defined;
- if not all indexes are formed, then pseudo-random bytes are generated for the remainder of the indices and we go to the previous step.

That is, the dazzling polynomial depends on the random data, the data being encrypted, and the portion of the public key of who the data are encrypted for.

5 The polynomial R = r * h is calculated in the field R/q

6 The polynomial R4 = R mod 4 is calculated

7 The polynomial R4 is converted to the string of oR4 octets, recording 4 coefficients into one octet. As a result, we get a 2n-bit octet string that depends on the dazzling ng polynomial and the recipient's public key. This line is then used to obtain the polynomial R/3.

8 We define the formation method, the required length of the string that must be used to form the R/3 polynomial with this method. We will use a byte to form five polynomial coefficients. To do this, we present a number in the ternary number system. The numbers 0, 1 ... 241, 242 can be set using 5 digits, each digit corresponds to the numbers 0, 1, 2, or 0, 1, -1. These values will be used to set the coefficients. The number of octets required in this case is Ceil (n/5). But octets with value 243 – 255 can not be used to set coefficients; the input line should be increased taking into account the probability of such octets. A string of a double length guarantees that the required number of coefficients is obtained.

The formation of the masking polynomial is as follows:

- perform initialization of the pseudo-random data generator with oR4 string;
- pseudo-random octets of 2 Ceil (n / 5) length are generated;
- the following 5 polynomial coefficients are calculated for each octet whose value is less than 243;
- the resulting polynomial is denoted as a mask;

9 The polynomial m'= (*MTrin* + maska) mod p is calculated

10 The success of the iteration is checked:

– The number of units (c1), minus units (c2) and zeros (c3) in the polynomial m ' is determined;
– if c1 + c2> 2t and c3> t the iteration is successful, otherwise go to step 1

The following steps are performed, if the iteration is successful in such a sequence:

11 $e = R + m' \pmod q$ polynomial is calculated

12 The polynomial e is converted to the octet string E

The result of the encryption algorithm is a string of octets *E*

The following mathematical transformations are performed in encryption:

1 M = b||mLen||m||0..0 (a string of octets)

2 mTrin = f1 (M) (mTrin – R/3 polynomial, f1 one-to-one function, that is, M can be restored by the value of mTrin)

3 S = oid||m||b|| part of h

4 r = f2 (S) (r – t – small polynomial, f2 – not one-to-one function)

5 R = r * h (in the field R/q)

6 oR4 = f3 (R mod 4) (oR4 – a string of octets, R cannot be restored)

7 maska = f4 (oR4) (maska – R/3 polynomial for which the conditions on the number of 1, -1, 0 are satisfied)

8 m' = mTrin + maska (mod 3)

9 e = R + m' (mod q) (e – polynomial in the field R/q)

10 E = f5 (e) (f5 is the inverse function, so the octet string e can be restored).

3.2 Decryption algorithm

When decrypting, the octet string E is fed to the input of the algorithm, and as a result, the output will receive an open message along with its length (in case of successful completion)

Entry:

E – string of bytes with an encrypted message;

f – recipient's private key;

recipient public key h (polynomial R / q) and corresponding octet string (h)

Output:

1 Sign of success *Success* (OK, ERROR).

2 Open message (in case of successful operation).

3 The length of the open message (in case of successful operation).

Algorithm

1 e polynomial is restored.

2 a = e*f in the field R/q polynomial is calculated

3 m' = a mod 3 is restored

4 m'requirements are checked

– the number of units (c1), minus units (c2) and zeros (c3) is determined in the polynomial m';
– if c1 + c2 > 2t and c3 > t then Success = OK, otherwise Success = ERROR and go (transition) to step 14.

Further steps are performed only in cases when Success = OK

5 R' = e – m' (mod q) is restoring

6 Polynomial R4 = R' mod 4 is calculated

7 R4 polynomial transformation into a oR4 octet string is carried out, recording 4 coefficients in one octet.

8 Calculation of the *maska* polynomial (see item 9 of the encryption algorithm)

9 mTrin = m' – maska (mod p) polynomial is calculated and M string is restored.

10 Determining the individual fields of M row:

– first λ bits – restored random sequence (*b*);
– next octet – length of encrypted data (restored mLen value);
– next mLen of octets – recovered message that was encrypted (m);

– next octets – is a restored string of addition.

11 Checking the success of the operation. The operation is considered successful if all conditions given below are met simultaneously:

– mLen restored value does not exceed the maximum admissible *EncMsgLenBytes* value;
– supplement (addition) string contains zeroes, the number of which is equal bufferLenBytes – db/8 – 1 – mLen

If at least one condition is not met, then Success = ERROR and go to step 14. Further steps are performed only in cases of successful operation.

12 The recovered values *b*, mLen, m are used to create a string S, calculate r, and calculate R = r * h in R/q field (steps 3-5 of the encryption algorithm).

13 Final check of the operation success: if R' recovered in step 5 matches the value of R obtained in step 12, the decryption operation ends successfully, the message m and its length mLen bytes is decrypted at the output.

14 If the operation fails (Success = ERROR), an empty message of length 0 is returned.

The following mathematical transformations are performed when decrypting

1 $e = f5^{-1}$ (E)

2 $a = e*f = (r*h + m') * f = (r * 3Gf^{-1} + m') * f = r * 3G + m'* f$ in the field R/q =

3 a mod 3 = m' (r * 3G mod 3 = 0; f mod 3 = 1)

4 R' = e – m' (mod q)

5 R4 = R' mod 4

6 oR4 = f3 (R4)

7 maska = f4 (oR4)

8 mTrin = m' – maska (mod 3)

9 $M = f1^{-1}$ (mTrin)

10 When using M = b || mLen || m || 0..0 (string of octets) separate fields are selected, as a result we get b, mLen, m.

4. Algorithms for encapsulation and decapsulation

For encapsulation algorithms, key data are used that are generated identically as for encryption. In fact, encapsulation algorithms use message encryption and decryption algorithms that are defined in advance and labeled *m_kem*. The length of the message *m_kemBytes*. Full description of the algorithms described above see in section 3.

An arbitrary allowed hash function is used as a hash function, providing a result length of 512 bits. We denote this function by Hash512.

Function input: octet string and its length.

Output: 512-bit octet string

4.1. The encapsulation algorithm

Input:

– length of the key for the symmetric encryption K_bytes (K_bytes $\leq \lambda/8$) ;
– the recipient's public key h (polynomial R/q) and the corresponding octet string (h).

Output:

– sign of success *(Success=OK, ERROR);*
– encapsulated key *Cc*;
– key SKey for symmetric encryption, the length of which is K_bytes (K_bytes).

As with the encryption algorithm, the first step is to check whether the operation can be performed:

– *Success= OK;*
– if *m_kemBytes* > *maxMsgLenBytes* then *Success= ERROR;*
– if K_bytes * 8 > λ then *Success= ERROR*

If *Success= ERROR* then the algorithm is not executed.

Next, the variant is considered in case of successful verification (*Success= OK;*)

1 We first perform steps 1-12 of the encryption algorithm for *m_kem* message of the length of *m_kemBytes*.

2 The dazzling polynomial is converted to the octet string *r* as follows:

- bit string bs1 is formed. The next bit of the string is 0 if the next polynomial coefficient is 0 and 1 if the next coefficient is 1 or -1. The length of the string is bs1 = n bits.
- bit string bs2 is formed. The next bit of the string is 0, if the next non-zero element is equal to1 and vice versa. The length of the string is bs2 = 2t bits.
- form a common bit string *bs = bs1 || bs2*
- denote r string of bytes, which corresponds to the bit string *bs*.

3 The value of the encapsulated key Cc and SKey is calculated.

c = *E*

The rest of the calculation depends on the value of λ.

If λ = 256, then:

H = Hash512 (r_);

C – are younger 32 octets of H;

SKey – are SKeyLen octets of senior 32 octets of H (SKeyLen ≤32);

Cc = C|| *c.*

If λ = 512, then

H1 = Hash512 (r_ || 1);

H2 = Hash512 (r_ || 2);

C – H1;

SKey – SKeyLen octets of H2 (SKeyLen ≤64);

Cc = C|| *c.*

4.2. Decapsulation algorithm

Input:

- length of the key for the symmetric encryption K_bytes (K_bytes ≤λ/8;
- encapsulated key *Cc*;
- recipient's private key *f*;
- recipient's public key h (R/q polynomial) and corresponding octet string (h).

Output:

- sign of success *(Success=OK, ERROR);*
- key SKey for symmetric encryption, the length of which is K_bytes (K_bytes).

As with the encryption algorithm, the first step is to check whether the operation can be performed:

- *Success= OK;*
- if *m_kemBytes* > *maxMsgLenBytes* then *Success= ERROR;*
- if K_bytes * 8 >*λ* then *Success= ERROR*

If *Success=ERROR* then the algorithm is not executed.

Next, the variant is considered in case of successful verification (*Success= OK;*)

1 Decoding *Cc* to *C* and *c.*

2 Decoding *c* is performed (steps 1-13 of the decryption algorithm)

3 If the decryption error *(Success = ERROR),* the algorithm returns an error.

4 If the length of the encrypted message does not match the message itself, or the messageы do not match, then Success = ERROR, the algorithm returns an error.

5 The recovered values *b,* mLen, m are used to create a string *S* and calculate *r.*

7 An octet string is formed for *r (r)* (see step 2 of the encapsulation algorithm)

6 The rest of the calculations depend on the value of λ.

If λ = 256, then:

H = Hash512 (r_);

C’ – are younger 32 octets of H;
SKey’ – are SKeyLen octets of senior 32 octets of H (SKeyLen ≤32).
If λ = 512, then:
H1 = Hash512 (r_ || 1);
H2 = Hash512 (r_ || 2);
C’ – H1;
SKey’ – are SKeyLen octets of H2 (SKeyLen ≤64).
7 If C’coincides with C then SKey = SKey’; sign of success *Success = OK*, otherwise *Success = Error*

5. Data encryption and authentication by a sender

During data encryption, consistent (agreed) functions, available to all subscribers, can be applied. They contain *MAC*(.) messages authentication and a symmetric encryption algorithm (Sym.Encrypt), for example [3, 14].

Subsequently, secret keys E_K and M_K, are produced using SKey, where E_K is the key for symmetric encryption and M_K is a suitable authentication key for the message of pre-encrypted data. (Algorithms for generating keys E_K and M_K from SKey can be determined by other standards).

5.1. Algorithm of data encryption and authentication by a sender

The algorithm is performed in the following sequence:

- messages (data, packet) Q are encrypted with the use of symmetric cipher on the key E_K, i.e.

$$C1 = Sym.Encrypt (Q, E_k);$$

- to authenticate C1ciphertext on M_K authentication key, calculate the C2 authentication code, i.e.

$$C2 = MAC (C1, M_k);$$

- finally, the sender sends *(Cc, C1, C2*) to the recipient.
-

5.2. Decryption and authentication of data by the recipient

The algorithms for decrypting and authenticating data by the recipient are executed in the following sequence:

- as a result of decapsulation, the recipient has the value *c',C',*SKey',*r';*
- if *r '*is *t*-small, *c' = c* and *C '= C*, then they output and apply the symmetric authentication and decryption key to *SKey'*. Otherwise, they determine the error and reject the secure message;
- the recipient calculates the message authentication code for C1 on the M_K key

$$C2' = MAC (C1, M_k)$$

- if *C2'* coincides with *C2* received from the sender, then the encrypted message *C1* is considered to be complete and authentic. Otherwise output "integrity violation" and stop executing the decryption algorithm;
- the recipient decrypts the cryptogram *C1* using the symmetric decryption algorithm agreed with the sender on E_K key, i.e.

$$Q = Sym.Decrypt (C1, E_k)$$

- the recipient receives the decrypted and authenticated *Q* data for further processing.

Note. If necessary, the order of encryption and authentication of Q data may be carried out in a different order, that is, first decryption and then authentication.

6. Analysis of cryptographic transformations complexity

The temporal characteristics of key data generation, encryption and decryption algorithms are shown in Table 6. Encapsulation and decapsulation algorithms actually use encryption and decryption primitives. Therefore, their temporal characteristics are not given. Processor (Intel (R) Core (TM) i5-3.1 GHz) clock [10, 11] is used to measure performance. For comparison, the data [5] were used for the ntruees787ep1 encryption algorithm, which is closest to the characteristics with crypto-stability λ=256.

Table 6

Temporal characteristics of key data generation, encryption and decryption algorithms

Stability	Generation	Encryption (59 bytes)	Decryption (59 bytes)
ntruees787ep1	16748110	111142	133926
λ=256 (n = 881)	4789644	87696	96604
λ=384	8065140	130556	143424
λ=512	11670676	160364	182724

The results of optimization of key data generation, encryption and decryption algorithms are presented in [6 – 10].

Key generation. The most laborious part of the algorithm is the inversion calculation and multiplication of polynomials in R/q field.

Inversion calculation. An extended Euclidean algorithm for polynomials is used to calculate the inversion. The computational complexity of Euclidean extended algorithm is determined by the number of division operations and the computational complexity of one operation.

The polynomials of degree *n* and *n-1* are used as data for the inversion calculation. Performing a single division operation reduces the degree of both polynomials by 1, i.e. the number of division operations is n.

The computational complexity of the division operation depends on the degrees of polynomials and the difference between them. It can be considered that the difference between degrees is one, then, the number of coefficients processed by the division operation equals a smaller degree and accordingly changes from n to 1. That is, labor intensity can be estimated as n-1 + n -2 +… + 1 = n (n-1)/2. Thus, the overall complexity of the investment stage is $O(n^3)$.

Methods for optimization of calculations.

1 To apply SIMD operations to work with the coefficients – there are limitations associated with variable block addresses.

2 Recalculation of inverse elements for coefficients.

Operation of polynomial multiplication

Computational complexity of multiplication of polynomials of general form $O(n^2)$.

Optimization through the use of polynomial F/3 features with compensation of dependence on the key format enables you to pass from the quadratic to the linear dependence on the degree of the polynomial.

The computational complexity of encryption and encapsulation operations

In fact, the encapsulation algorithm uses the encryption algorithm, and the decapsulation algorithm uses the decryption algorithm. The rest of the operations performed for encapsulation / decapsulation take relatively little time. That is why it is enough to consider the computational complexity for encryption and decryption algorithms.

Conclusions

1. The generalized results of the implementation of the 1st stage of the international competition for the creation of post-quantum ASEs, KEPs and ESs, proposed by NIST USA, as

well as comparisons of most alternatives, allow us to make a conclusion about the prospects of creating standards for these cryptographic transformations in polynomial rings (algebraic lattices) [1 – 7]. Another interesting area that also deserves attention and exploratory research is the construction of post-quantum cryptographic transformations using methods of the theory of fault-free coding [15 – 17].

2. Mechanisms for constructing ASE and KEP are proposed in [2], the application of which makes it possible to provide 128 bits of quantum and 256 classical crypto-stability, i.e. including the 5^{th} level of cryptographic stability. But in our opinion, the problem of ensuring encryption and encapsulation, including up to 7 levels of stability, is important both from theoretical and practical point of view, since the 5th level of cryptographic stability for the quantum period is insufficient.

3. Therefore, the current problem is the creation and standardization of ASE and KEP algorithms of 256 bit quantum and 512 classical cryptographic stability. Practically the problem of creation and standardization of post-quantum algorithms of ASE and KEP of 5 – 7 levels of cryptographic stability is solved in Ukraine based on cryptographic transformations using algebraic lattices (polynomial rings).

4. The common parameters for ASE and KEPs of cryptographic transformations should be calculated, provided that the required level of crypto-stability is ensured, as well as to ensure the success of operations (such as no decryption errors) and to reduce the computational complexity. A list of general parameters of the encryption algorithms and key encapsulation algorithms of SKELA algorithms is shown in Table 1.

5. Only general parameters are used to calculate additional parameters. Recalculating additional parameters makes it possible to reduce the computational complexity of basic cryptographic transformation operations.

6. The asymmetric key pairs – private key f and public key h – are used for asymmetric encryption and encapsulation of keys. They are calculated on the basis of the use of polynomials G, F, whose degree is *n*, and the coefficients modulo *p (p = 3)*, i.e. take the values 0, -1, 1. The result is as follows: the private key f and the public key h (polynomials) and h (byte task h) calculated simultaneously.

7. In the process of developing encryption algorithms, the NTRU encryption algorithm [4] has been taken into account, which has been successfully tested over time since it has been in use for almost 10 years. But it has a significant drawback that has to do with the possibility of a decryption error. This drawback is absent in the SKELYA algorithm [10].

8. In developing the SKELYA algorithm, the modern practice of using the prime q as a module instead of the number 2k is taken into account, and the polynomial xn – x – 1 instead of the polynomial xn – 1, which provides protection against known attacks. Therefore, field $(Z/q[x]/(x^n - x - 1))$ is used as a field for NTRU Prime [5], but the parameters are calculated taking into account the required levels of crypto-stability $\lambda = \{256, 384, 512\}$ and ensuring that no decryption errors exist.

9. In the encryption algorithm, the following data are used as input: string for encryption (m); line length m (mLen); the recipient's public key h (R / q polynomial) and the corresponding octet string (h). The following data are used is output: sign of Success (OK, ERROR); encrypted string E in case of successful operation.

10. The following data are used as input data in the decryption algorithm: E – a byte string with an encrypted message; f – private key of the recipient; h (R / q – the recipient's public key polynomial) and the corresponding octet string (h). The following data are used as output data: sign of Success (OK, ERROR); open message (in case of successful operation) and length of open message (in case of successful operation).

11. Key data, generated identically as for encryption, are used in the encapsulation algorithm. In fact, encapsulation algorithms use encryption and decryption algorithms of the message defined in advance and indicated by m_kem

12. An arbitrary allowed hash function, that provides a result length of 512 bits, is used as a hash function.

13. The input data in the encapsulation algorithm are the length of the key for symmetric encryption of K_bytes (K_bytes $\leq \lambda / 8$) and the recipient's public key h (polynomial R/q) and the corresponding octet string (h), and the output data are a sign of success (Success = OK, ERROR), Cc encapsulated key and a SKey key for symmetric encryption of K_bytes (K_bytes) length.

14. The input data in the decapsulation algorithm are as follows: symmetric encryption key of K_bytes (K_bytes $\leq \lambda / 8$ length; encapsulated Cc key; recipient private key f and recipient public key h (polynomial R/q) and corresponding octet string (h).

15. The joint use of encryption algorithms and encapsulation of keys makes it possible to produce (calculate) symmetric encryption and authentication keys on communication channels, which in turn makes it possible to exchange the protected information with high speed and ensuring its integrity, confidentiality and cryptographic integrity of such key data.

16. When developing algorithms for encryption and encapsulation of keys, optimization methods of cryptographic transformations based on multiplication of polynomials were applied. To measure the characteristics of time complexity, processor (Intel (R) Core (TM) i5-3.1 GHz) clock cycles were used [9,10]. To compare the obtained complexity data, we used the data [5] (ntruees787ep1 encryption algorithm, which is closer to the characteristics with cryptographic strength $\lambda = 256$).

17. Thus, this article presents the main results and data on encryption algorithms, key encapsulation and their use for encryption and authentication of data on communication channels. They are achieved by means of solving such particular problematic tasks as: calculating general and additional parameters; generating asymmetric pairs of encryption keys and encapsulation; development of asymmetric encryption algorithms (encryption and decryption); development of asymmetric encapsulation algorithms and key decapsulation; developing proposals (algorithms) for calculating the keys of secure communication sessions for their use in symmetric data encryption in communication channels, as well as optimizing and assessing the complexity of direct and inverse transformations during encryption and encapsulation.

18. The main advantages of asymmetric encryption and key encapsulation algorithms are: providing 5 – 7 levels of post-quantum and classical stability; security against special attacks, as well as providing symmetric data encryption on communication channels is the use of block and stream high-speed transformations

References:

1. Lily Chen Report on Post-Quatum Cryptography. NISTIR 8105 (DRAFT) / Lili Chen, Stephen Jordan, Yi-Kai-Liu, Dustin Moody, Rene Peralta, Ray Perlner, Daniel Smith-Tone // Electronic resource. Access mode: http://csrc.nist.gov/publications/drafts/nistir-8105/nistir_8105_draft.pdf.

2. Status Report on the First Round of the NIST Post-Quantum Cryptography Standardization Process / Gorjan Alagic and others // https://nvlpubs.nist.gov/nistpubs/ir/2019/NIST.IR.8240.pdf

3. Gorbenko Yu.I. Methods for construction and analysis of cryptographic systems. Kharkiv : Fort, 2015. 959 p. (In Ukr.).

4. American National Standard X9.98-2010. Lattice-based polynomial public key encryption algorithm, Part 1: key establishment, Part 2: data encryption. 2010.

5. Daniel J. Bernstein NTRU Prime / Daniel J. Bernstein, Chitchanok Chuengsatiansup, Tanja Lange, Christine van Vredendaal // Electronic resource. Access mode: https://ntruprime.cr.yp.to/ntruprime-20160511.pdf. https://bench.cr.yp.to/results-encrypt.html

6. Gorbenko I.D. General statements and analysis of the end-to-end encryption algorithm NTRU Prime IIT Ukraine / I.D. Gorbenko, E.G. Kachko, MV Esina // Radiotekhnika. Kharkov : KNURE, 2018. Is. 193. P. 5-16.

7. Gorbenko I. D. Calculation of general parameters for NTRU Prime Ukraine of 6-7 levels of stability / I. D. Gorbenko, A. N. Alekseychuk, O. G. Kachko, M. V. Yesina, I. V. Stelnik, S. O. Kandy, V. A. Bobukh, V. A. Ponomar // Telecommunications and Radio Engineering, 2019. Vol. 78, Is. 4. P.327-340. DOI: 10.1615/TelecomRadEng.v78.i4.40.

8. Gorbenko I.D. Methods of building general parameters and keys for NTRU Prime Ukraine of 5th–7th levels of stability. Product form / I.D. Gorbenko,O.G. Kachko, Yu.I. Gorbenko, I.V. Stelnik, S.O. Kandyi, M.V. Yesina //

Telecommunications and Radio Engineering, 2019. Vol. 78, Is. 7 P. 579-594. DOI: 10.1615/TelecomRadEng.v78.i7.30.98.

9. CALCULATION OF GENERAL PARAMETERS FOR NTRU PRIME UKRAINE OF 6-7 LEVELS OF STABILITY / I. D. Gorbenko, A. N. Alekseychuk, O. G. Kachko, M. V. Yesina, I. V. Stelnik, S. O. Kandy, V. A. Bobukh,. A. Ponomar . pages 327-340 DOI: 10.1615/TelecomRadEng.v78.i4.40. Vol. 78, 2019 Is. 4.

10. Kachko O., Gorbenko I., Yesina M., Kandiy S. POLYNOMIALS MULTIPLICATION FUNCTIONS FOR ORDINARY AND PRODUCT FORM OF ONE OF THE POLYNOMIALS REPRESENTATION: https://github.com/KandiyIIT/NTRU-POLYNOMIALS-MULTIPLICATION.

11. Ran Canetti, Hugo Krawczyk Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels. Electronic resource. Access mode: http://iacr.org/archive/eurocrypt2001/20450451.pdf.

12. Post-Quantum Cryptography. Electronic resource. Access mode: https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

13. EUF-CMA and SUF-CMA. Electronic resource. Access mode: https://blog.cryptographyengineering.com/euf-cma-and-suf-cma/.

14 DSTU ISO / IEC 18033-2: 2015 (ISO / IEC 18033-2: 2006, IDT) Information Technology. Methods of Protection. Encryption Algorithms. Part 2. Asymmetric Ciphers. (In Ukr.)

15. Kuznetsov A., Pushkar'ov A., Kiyan N. and Kuznetsova T. Code-based electronic digital signature // 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT), Kyiv, Ukraine, 2018, pp. 331-336. DOI: 10.1109/DESSERT.2018.8409154.

16. Kuznetsov A. A., Gorbenko Yu. I., Prokopovych-Tkachenko D. I., Lutsenko M. S., Pastukhov M. V. NIST PQC: Code-Based Cryptosystems // Telecommunications and Radio Engineering. 2019. Vol. 78. Is. 5, pp. 429-441. DOI: 10.1615/TelecomRadEng.v78.i5.50.

17. Gorbenko Y., Svatovskiy I. and Shevtsov O. Post-quantum message authentication cryptography based on error-correcting codes // 2016 Third International Scientific-Practical Conference Problems of Infocommunications Science and Technology (PIC S&T), Kharkiv, 2016. P. 51-54. DOI: 10.1109/INFOCOMMST.2016.7905333.

Kharkiv National V.N. Karazin University;
JSC "Institute of Information Technologies";

Received 09.08.2019

UDC 004.056.5 **DOI:10.30837/rt.2019.3.198.02**

I.I. BOBOK, PhD, A.A. KOBOZEVA, Doctor of Sciences

STEGANALYSIS METHOD EFFICIENT FOR THE HIDDEN COMMUNICATION CHANNEL WITH LOW CAPACITY

Introduction

The rapid development of digital steganography, the publication of a large number of scientific works in this area in the open sources and restriction and even the ban of the use of cryptography on the legislative level in many countries have led to an increase in the use of steganographic methods for transmitting and storing confidential information [1,2]. The main advantage of steganography compared to cryptography is the concealment of the very fact that the confidential information is present in a particular information content, which makes the use of the steganographic system the common solution for the organization of a hidden communication channel [3]. However, the goals of organizing such a channel may be different: from the harmless, concerning specific individuals, to those that threaten the stability and vitality of a group of people united by specific goals, or the society as a whole. In this connection, the relevance of organizing an effective steganalysis of informational content increases. Its main task is to identify the presence or absence of additional information embedded in the non-attracting content, or container [4].

The most commonly used containers when organizing a hidden communication channel are digital images (DI), which is why DI is considered in this paper.

Currently, all steganalysis methods according to [1] can be classified into 6 main categories:

- visual steganalysis (visual detection of differences between container and steganographic message;
- signature or specific steganalysis (these techniques search for signature patterns to determine the presence of a hidden message);
- statistical steganalysis (those techniques developed by analyzing the embedding procedure and determining certain statistics that get modified as a result of the embedding process);
- spread spectrum steganalysis (aimed at the detection of embedded data introduced by steganographic methods, which perform the frequency spectrum spreading of the signal-container, i.e. by SS-methods (Spread-Spectrum));
- transform domain steganalysis (in the process of steganalysis the transform domain of DI investigated, e.g. discrete cosine transform domain, discrete wavelet transform domain, singular decomposition domain, etc.);
- universal or blind steganalysis (these techniques tries to detect the embedded messages regardless the steganographic technique applied to cover image).

The methods of the last group seem to be the most attractive since they are not related to the features of specific steganographic algorithms. However, the practice shows that they cannot provide equally high efficiency in detection of the results of various steganographic transform methods.

The most widely used steganographic method of DI transform is the modification of the least significant bit (the LSB-method) in its various implementations. The LSB-method detection remains an actual task. There are a large number of different steganalysis methods aimed at detecting the results of the LSB-method [4 - 8], belonging to different categories listed above. Most often, they are focused on statistical analysis [2], so the features of the current use of the LSB-method (with the low capacity of the hidden communication channel (HCC) (embedding rate)) make the vast majority of existing steganalysis methods ineffective (with 0.1<HCC<0.25 bpp), and practically unsuitable (with HCC <= 0.1 bpp) for detecting the hidden information presence. The testing of many modern methods under HCC <0.1 bpp is not performed at all [2, 9, 10]. Although the research in this area is being conducted and the development of new approaches is in progress [11-14], the task of providing the high efficiency of steganalysis method for LSB-embedding detection, which does not depend on DI type (color or grayscale) under the low HCC is still not solved.

For example, in [11], a steganographic algorithm is proposed, which is positioned as capable under the conditions of low embedding rate (0.01 bpp), however, the accuracy for these conditions is only 52.28 %. Similar efficiency results obtained under the embedding rate of 0.01 bpp for the steganalysis algorithms developed in [14].

In [12], a steganalysis algorithm for grayscale images was developed. The results of testing the algorithm given in the paper were carried out on more than 9000 DIs and obtained under the HCC rate from 0.1 to 0.5 bpp with the error rate of 21.0 % for the smallest HCC value. Such a result obviously cannot be considered a satisfactory value.

The steganalysis algorithm developed in [13] for color DI deserves great attention. The algorithm is based on analyzing the features of changing the number of color triads in a matrix of unique image colors while embedding additional information: containers stored in a lossy format have a small number of consecutive triads of triplets, while as a result of steganographic transform, even with small values of HCC, there is a significant increase in the number of such the triads. This makes it possible to detect the presence/absence of additional information in the digital content analyzed. The HCC of 0.05 bpp was the smallest value considered when testing the developed algorithm, while the detection accuracy coefficient was *ACC* = 0.9865. However, the algorithm is not efficient for grayscale images, which is a significant drawback.

Due to the large amounts of information, which is stored, sent or processed nowadays and usually stored in lossy formats, it is reasonable to consider the DI in a lossy format (LF) as a container in this paper.

The *aim* of the work is to increase the efficiency of steganalysis by developing a new steganalysis method for detecting the presence of additional information embedded by the LSB-method into the DI-container under conditions of the low communication channel capacity.

The HCC <= 0.1 bpp is considered as low HCC values. The effectiveness of the steganalysis algorithm estimated by Type I and Type II errors, as well as by the detection accuracy coefficient, formally defined below.

Main Body

For distinctness and taking into account the widest spreading of the Jpeg it is considered as a lossy format for DI (with various quality factors $QF \in \{0,1,2,\ldots,99,100\}$) further in the paper, Tif is used as a lossless format (LLF), a single rectangular $m \times n$ – matrix is considered as a formal representation of an arbitrary DI.

Let the matrix of the original DI in the LLF be F_T, and the matrix of the corresponding DI, which was obtained from LF by means of its repeated saving is F_J. Denote an arbitrary 4×4 – block of F_T / F_J as B_T / B_J respectively. Let $\sigma_T = \left(\sigma_1\left(B_T\right), \sigma_2\left(B_T\right), \sigma_3\left(B_T\right), \sigma_4\left(B_T\right)\right)^T$, $\sigma_J = \left(\sigma_1\left(B_J\right), \sigma_2\left(B_J\right), \sigma_3\left(B_J\right), \sigma_4\left(B_J\right)\right)^T$ be the vectors, those elements are the singular numbers (SN) of B_T, B_J blocks and wherein:

$$\sigma_i\left(B_T\right) \geq \sigma_{i+1}\left(B_T\right),\ \ \sigma_i\left(B_J\right) \geq \sigma_{i+1}\left(B_J\right),\ i = 1,2,3,\ \sigma_4\left(B_T\right) \geq 0,\ \sigma_4\left(B_J\right) \geq 0. \qquad (1)$$

Let us normalize vectors σ_T and σ_J, the obtained result given below:

$$\bar{\sigma}_T = \frac{\sigma_T}{\|\sigma_T\|} = \left(\bar{\sigma}_1\left(B_T\right), \bar{\sigma}_2\left(B_T\right), \bar{\sigma}_3\left(B_T\right), \bar{\sigma}_4\left(B_T\right)\right)^T,\ \bar{\sigma}_J = \frac{\sigma_J}{\|\sigma_J\|} = \left(\bar{\sigma}_1\left(B_J\right), \bar{\sigma}_2\left(B_J\right), \bar{\sigma}_3\left(B_J\right), \bar{\sigma}_4\left(B_J\right)\right)^T, \quad (2)$$

where $\|\sigma_T\|$ and $\|\sigma_J\|$ are norms of σ_T and σ_J. The condition (1) obviously also holds for elements of vectors $\bar{\sigma}_T$ and $\bar{\sigma}_J$, which further will be called the normalized SN.

In [15], a general approach to solve the problem of identifying violations of the DI integrity was proposed. It is based on the perturbation theory and matrix analysis and underlies further

reasoning. The development of the approach was reflected in [16], where it was shown that for the majority of the corresponding blocks B_T and B_J and matrices F_T and F_J, obtained as a result of their standard splitting, the following equation takes place:

$$\angle\left[e_1,\bar{\sigma}_T\right] > \angle\left[e_1,\bar{\sigma}_J\right], \tag{3}$$

where $e_1 = (1,0,0,0)^T$ is a standard space basis vector R^4, $\angle\left[e_1,\bar{\sigma}_T\right]$, $\angle\left[e_1,\bar{\sigma}_J\right]$ are the magnitudes of the angles between the vectors $\bar{\sigma}_T$ and e_1, $\bar{\sigma}_J$ and e_1 respectively, and the lower the quality factor QF used to obtain F_J, the smaller will be the angle between the normalized SN vector in the block B_J and the vector e_1.

Consider the relation (3) in more detail. Taking into account that

$$\left(e_1,\bar{\sigma}_T\right) = \|e_1\|\left\|\bar{\sigma}_T\right\| cos\left(\angle\left[e_1,\bar{\sigma}_T\right]\right) = cos\left(\angle\left[e_1,\bar{\sigma}_T\right]\right),\ \left(e_1,\bar{\sigma}_J\right) = \|e_1\|\left\|\bar{\sigma}_J\right\| cos\left(\angle\left[e_1,\bar{\sigma}_J\right]\right) = cos\left(\angle\left[e_1,\bar{\sigma}_J\right]\right),$$

where $\left(e_1,\bar{\sigma}_T\right), \left(e_1,\bar{\sigma}_J\right)$ are the scalar products of corresponding vectors, the relation (3) is rewritten as follows:

$$\left(e_1,\bar{\sigma}_T\right) < \left(e_1,\bar{\sigma}_J\right),$$

where for the majority of the corresponding blocks of F_T and F_J matrices we obtain:

$$\bar{\sigma}_1\left(B_T\right) < \bar{\sigma}_1\left(B_J\right). \tag{4}$$

Consider the case, when the singular numbers of block B_T has been perturbed because of lossy compression of image with the matrix F_T, that led to perturbation of elements of vector $\bar{\sigma}_T$. Denote these perturbations as $\Delta\sigma_1, \ldots, \Delta\sigma_4$. Then for the corresponding block B_J we have:

$$\bar{\sigma}_i\left(B_J\right) = \bar{\sigma}_i\left(B_T\right) + \Delta\sigma_i, \quad i = \overline{1,4}. \tag{5}$$

Then, taking into account (4) for most blocks:

$$\Delta\sigma_1 > 0. \tag{6}$$

If we consider the Euclidean norm as a vector norm in (2), then taking into account the normalization of vectors $\bar{\sigma}_T$ and $\bar{\sigma}_J$ and relation (5), we have:

$$\sum_{i=1}^{4}\left(\bar{\sigma}_i\left(B_T\right)\right)^2 = \sum_{i=1}^{4}\left(\bar{\sigma}_i\left(B_J\right)\right)^2 = \sum_{i=1}^{4}\left(\bar{\sigma}_i\left(B_T\right) + \Delta\sigma_i\right)^2,$$

from where

$$\sum_{i=1}^{4}\left(2\Delta\sigma_i\bar{\sigma}_i\left(B_T\right) + \left(\Delta\sigma_i\right)^2\right) = 0. \tag{7}$$

Let us rewrite (7) in more detail:

$$2\Delta\sigma_1\bar{\sigma}_1\left(B_T\right) + \left(\Delta\sigma_1\right)^2 + \left(\Delta\sigma_2\right)^2 + \left(\Delta\sigma_3\right)^2 + \left(\Delta\sigma_4\right)^2 + 2\Delta\sigma_2\bar{\sigma}_2\left(B_T\right) + 2\Delta\sigma_3\bar{\sigma}_3\left(B_T\right) + 2\Delta\sigma_4\bar{\sigma}_4\left(B_T\right) = 0 \tag{8}$$

It is $\bar{\sigma}_1\left(B_T\right) > 0$ for almost all blocks of the original DI in the LLF. Then

$$2\Delta\sigma_1\bar{\sigma}_1\left(B_T\right) + \left(\Delta\sigma_1\right)^2 + \left(\Delta\sigma_2\right)^2 + \left(\Delta\sigma_3\right)^2 + \left(\Delta\sigma_4\right)^2 > 0,$$

and the equality to zero of the expression value on the left-hand side of (8) is possible only due to the fact that there are negative values among $\Delta\sigma_2$, $\Delta\sigma_3$, $\Delta\sigma_4$.

Let us demonstrate that it is $\Delta\sigma_2 \le 0$ for most DI blocks using proof by contradiction. Let us suppose that $\Delta\sigma_2 > 0$. Then consider the principal possibility that (8) equals zero due to $\Delta\sigma_3$, $\Delta\sigma_4$, i.e. is it possible, in principle, to provide the equality to zero of the expression in (8) only at the expense of $\Delta\sigma_3$, $\Delta\sigma_4$ negativity. To do this, assume that both of these values are negative: $\Delta\sigma_3 < 0$ and $\Delta\sigma_4 < 0$. Moreover, given the fact that singular numbers are always non-negative, the maximum possible modulo values of $\Delta\sigma_3$, $\Delta\sigma_4$ are as follows:

$$\Delta\sigma_3 = -\bar{\sigma}_3(B_T),\ \Delta\sigma_4 = -\bar{\sigma}_4(B_T). \tag{9}$$

Then

$$\bar{\sigma}_3(B_J) = \bar{\sigma}_4(B_J) = 0,$$

that means that the vector $\bar{\sigma}_J = (\bar{\sigma}_1(B_J), \bar{\sigma}_2(B_J), 0, 0)^T$ lies in a plane defined by two vectors of the standard space basis R^4: $e_1 = (1,0,0,0)^T$ and $e_2 = (0,1,0,0)^T$. The end of the vector $(\bar{\sigma}_1(B_J), \bar{\sigma}_2(B_J), 0, 0)^T$ due to its normalization lies on the unit circle, which is the intersection of the unit sphere of space R^4 and the mentioned plane, whose center coincides with the origin. This position of the vector $\bar{\sigma}_J$ is the result of a perturbation of the normalized vector $\bar{\sigma}_T$, i.e. the result of its rotation at a certain angle within the first coordinate orthant of space R^4. First two coordinates $\bar{\sigma}_1(B_T), \bar{\sigma}_2(B_T)$ of vector $\bar{\sigma}_T$ are the direction cosines of its projection onto the coordinate plane of space R^4, defined by the vectors of the standard basis e_1 and e_2. In this case, taking into account the conditions (1), a simultaneous increase in the two first coordinates of the vector could not occur as a result of the DI compression, and if the first coordinate exactly increased (see (6)), then the second one should decrease. Thus, with $\Delta\sigma_2 > 0$ it is impossible to provide (8) even under the condition (9), i.e. with the largest modulus of possible negative perturbations $\Delta\sigma_3$, $\Delta\sigma_4$. Thus:

$$\Delta\sigma_2 \le 0. \tag{10}$$

In [17] the concept of gap $svdgap(i,A)$ of the singular numbers $\sigma_i(A)$ of matrix A introduced:

$$svdgap(i,A) = \min_{i \ne j} |\sigma_j(A) - \sigma_i(A)|.$$

Let us introduce the similar concept for the normalized singular numbers $\bar{\sigma}_i(A)$ of matrix A. Let us define $svdgap_n(i,A)$ as normalized gap of singular number $\sigma_i(A)$ of matrix A, which can be determined as follows:

$$svdgap_n(i,A) = \min_{i \ne j} |\bar{\sigma}_j(A) - \bar{\sigma}_i(A)|.$$

Taking into account the conditions (1) for any block B of any DI obtain:

$$svdgap(1,B) = \sigma_1(B) - \sigma_2(B),$$

$$svdgap_n(1,B) = \bar{\sigma}_1(B) - \bar{\sigma}_2(B). \tag{11}$$

Taking into account (11), (4), (10):

$$svdgap_n(1,B_J) = \bar{\sigma}_1(B_J) - \bar{\sigma}_2(B_J) > \bar{\sigma}_1(B_T) - \bar{\sigma}_2(B_T) = svdgap_n(1,B_T).$$

Thus, when storing DI in lossy formats in most DI blocks (for which (3) holds) the normalized gap $\bar{\sigma}_1(B_J)$ is greater than the normalized gap $\bar{\sigma}_1(B_T)$, where B_T, B_J are the corresponding blocks F_T, F_J.

Since, with a decrease in the quality factor *QF* used to save DI in the Jpeg format, the number of blocks for which (3) takes place will increase [16], the increase will be observed in the number of DI blocks in which the normalized gap of the maximum SN will grow compared to its normalized gap in the corresponding blocks of the corresponding DI in a lossless format. Indeed, the lossy compression in a certain way reflects on the SN values of DI blocks: it reduces the contribution of the signal high-frequency component, blurs DI and leads to a decrease in the minimum singular number values of blocks [18]. However, in case of a compression with a high quality factor (relatively small values of the elements of the quantization matrix) the changes in the SN will be insignificant, i.e. the number of DI blocks, for which the relation (3) will not hold in the case of high *QF* will be greater than in the case of small *QF*. The number of such blocks does not increase monotonously with the quality factor decreasing (increasing elements of the quantization matrix), since with decreasing of quality factor the value $\angle[e_1, \bar{\sigma}_J]$ will decrease [16], differing more and more from the value $\angle[e_1, \bar{\sigma}_T]$ in accordance with (3). The obtained theoretical conclusion is illustrated in 2 DIs in lossless formats (Tif), where a monotonous decrease took place in the number of DI blocks obtained as a result of standard splitting, in which an increase in the normalized gap of the maximum singular number was observed, with an increase in *QF* (Fig. 1). This conclusion found the practical confirmation in the results of the computational experiment, described in detail below (see Fig. 3 (curve 1)).

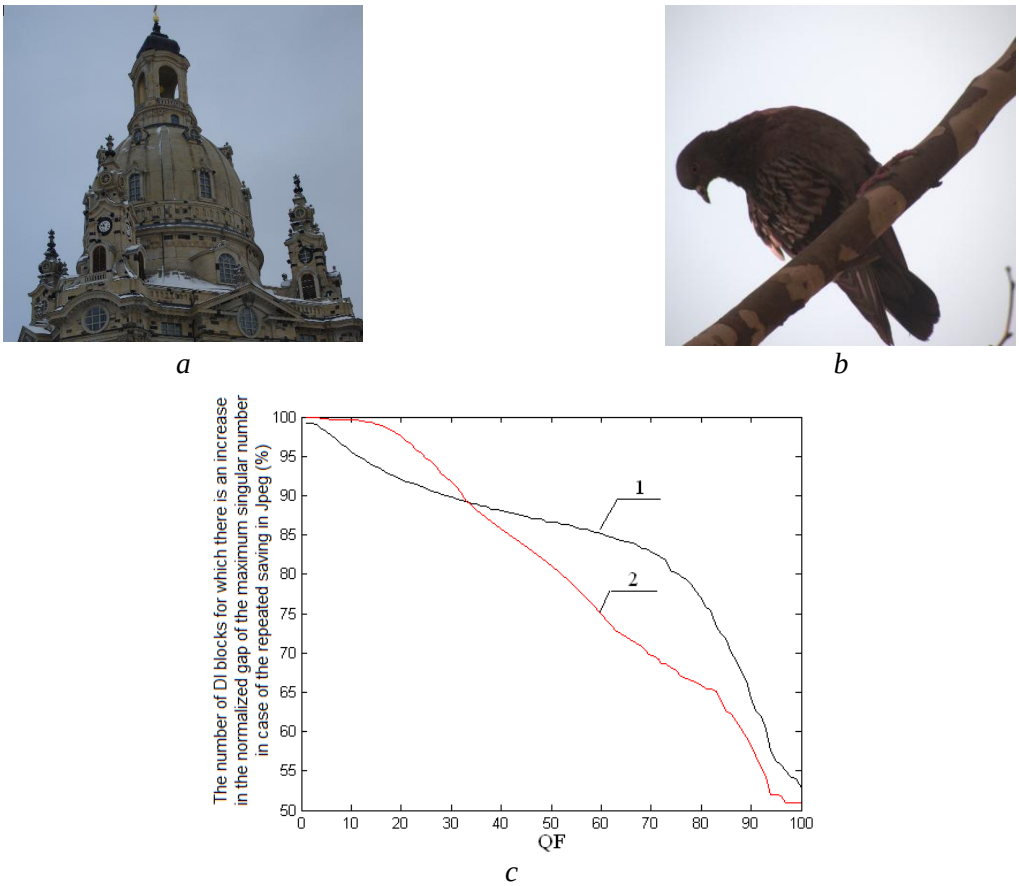


Fig.1. The results of the analysis of the maximum singular number normalized gap in the DI blocks with repeated saving in the Jpeg format with various quality factors *QF*: *a* – the original DI (Tif) from the img_Nikon_D70s base [19]; *b* – the original DI (Tif), obtained by a non-professional video camera; *c* – diagram that illustrates how the number of 4×4-blocks in DI, where the normalized gap of the maximum SN increased depends on *QF*: 1 – for DI from Fig.1, *a*, 2 – for DI from Fig.1, *b*

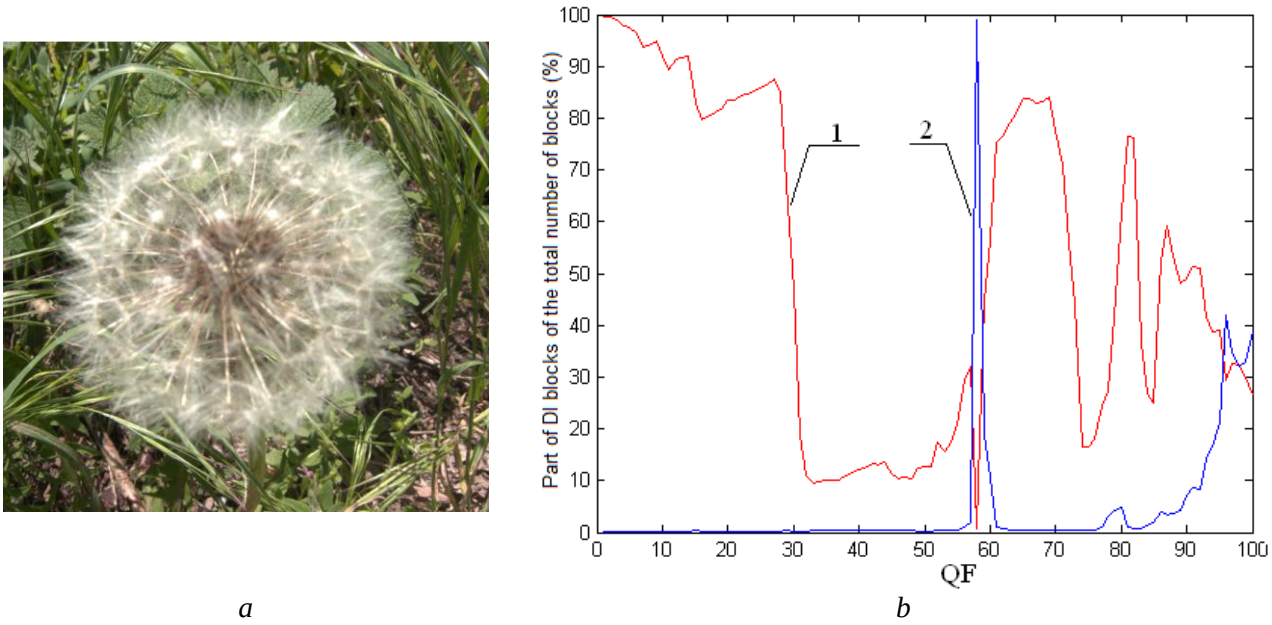
The monotonous increase in the number of blocks in the corresponding DI, for which the normalized gap of the maximum SN of block grows, will be violated with a decrease in the quality factor *QF* used to compress the original DI, if the image in lossy format (Jpeg) used as source. Indeed, if the original DI was stored with a quality factor QF_1, then its repeated saving with the same quality factor cannot significantly change the quantitative characteristics of the blocks, in particular, the singular number values (Table 1) (re-quantizing of the discrete cosine transform (DCT) coefficients in DI performed with the same quantization matrix as the primary one). Changes (if any) of the singular numbers will occur due to the presence of rounding in the process of compression [20], as well as rounding which causes the computational error when working in a floating point system. Therefore the number of blocks for which the normalized gap of the maximum SN will increase will be very small compared to the original DI (close to 0) (which is no longer typical for compression of image in the LLF with any quality factor), while the number of blocks, where the normalized gap of the maximum SN does not change, will be significant (the computational experiment shows, that the number of such blocks can exceed 90 % of the total number of blocks). However, the re-compression of the original DI with a quality factor different from QF_1 obviously breaks the above-mentioned monotony of changing the $4\times4-$blocks number, for which the normalized gap of the maximum SN increased, since the DI blocks have already been compressed (primary) with losses and the high-frequency component has already been reset to zero (taking into account the rounding that occur in the process of DI recovery after compression – it is close to zero). During the initial quantization, DCT coefficients corresponding to high (possibly medium) frequencies will be zeroed, resulting in a significant relative decrease in the lowest (possibly average) singular number values. Re-quantization will not significantly change the values of the DCT coefficients that became small after the initial quantization (and the smallest and possibly medium SN). For the maximum SN, the situation is different: as a result of quantization and subsequent rounding during the DI compression, the maximum singular number of block recovered after image quantization can both decrease and increase (depending on the rounding result after quantization), while three other singular numbers in block, as mentioned above, remain practically unchanged (this situation will occur for the same DI block when this image is saved for the second time with losses with different *QF*s (the results given in Table 1 for one randomly selected DI block stored in Jpeg format (QF = 85)) illustrate the above). This will lead to the fact that when DI is compressed with a quality factor $QF \neq QF_1$, the number of DI blocks for which the normalized gap of the maximum SN increases will be significantly different from zero, as the computational experiment shows, but the number of blocks where the normalized gap of the maximum SN will not change will decrease dramatically. These facts will lead to a loss of monotony for the number of blocks in which the growth of the normalized gap of the maximum SN occurred, with an increase in *QF*, which is an indication that the original DI was saved with losses.

Table 1

Result of repeated saving of original DI to Jpeg format with different quality factors QF, which was originally stored in Jpeg format (*QF*=85), for one $4\times4-$block

QF	Singular spectrum of the block				The normalized gap of the maximum SN
Original DI	**214.1468**	**4.8940**	**3.6310**	**0.1440**	**0.9768**
55	214.9262	3.0877	0.4180	0.1442	0.9855
60	210.4311	2.7594	0.9511	0.4780	0.9868
65	215.8175	2.8976	0.5545	0.3172	0.9865
70	210.2940	5.3903	1.7155	0.2854	0.9593
75	211.4048	5.6764	4.8425	0.0834	0.9677
80	213.0100	5.8375	3.5443	0.3043	0.9721
85	**214.1468**	**4.8940**	**3.6310**	**0.1440**	**0.9768**
90	214.0043	5.2478	4.1863	0.3131	0.9750
95	213.8960	4.8732	3.3984	0.4325	0.9768

The situation described above, in principle, can be obtained for DI, originally stored in Jpeg with any QF_1. Indeed, the original DI in the process of its analysis can be saved to the Jpeg format with each quality factor $QF \in \{0,1,2,\ldots,99,100\}$ (with step 1), which will give an opportunity to get a global minimum of the corresponding curve reflecting how the number of DI blocks in which the normalized gap of maximum SN increased as a result of repeated saving with losses depends on *QF*. The global minimum will be reached at $QF = QF_1$, and its value will be close to 0. The results of the analysis of a specific DI are shown in Fig. 2 to illustrate the truth of the above. Curves 1 and 2 (Fig. 2) have a global minimum and a maximum, respectively, with $QF = QF_1 = 58$, while the dependence of the number of DI blocks, in which there was an increase in the gap of the maximum SN, from *QF*, is not monotonic, which is in full compliance with the foregoing.



a *b*

Fig.2. The results of the analysis of the DI singular number blocks: *a* – the original DI (Jpeg format (QF = 58)); *b* – graphs which show relationship between the number of DI 4×4-blocks, given as a percentage of the total number of blocks, and the quality factor *QF* used in the repeated compression of DI: 1 – the number of blocks in which the normalized gap of the maximum SN increased; 2 – the number of blocks in which the normalized gap of the maximum SN did not change during the repeated saving

For practical confirmation of the findings, a computational experiment was carried out, which involved:

• 450 original DIs in a lossless format (Tif): 150 DIs from the 4cam_auth base [21] (size is 500×500 pixels) – set T_1; 200 DIs received by non-professional video cameras (size is 600×600 pixels) – set T_2, 100 DIs from the database img_Nikon_D70s [19] – set T_3 (size is 2000×2000 pixels);

• 4950 original DIs in Jpeg format, obtained by storing DIs from T_1, T_2, T_3 to Jpeg format with quality factors $QF \in M_1 = \{55,60,65,70,72,75,80,85,90,93,95\}$ (corresponding sets are denoted by $T_1^{(i)}$, $T_2^{(i)}$, $T_3^{(i)}$, $i \in M_1$).

In the experiment, each DI from the sets T_1, T_2, T_3, $T_1^{(i)}$, $T_2^{(i)}$, $T_3^{(i)}$, $i \in M_1$ was stored in the Jpeg format with values $QF \in M_2 = \{55,60,65,70,75,80,85,90,95\}$. Note that the elements of the sets M_1, M_2 were chosen as the most frequently used in practice, but at the same time M_1 contained values which were absent in M_2 by design, in order to consider such options when the quality factors of the secondary compression of DI do not coincide with the quality factor of the primary

one. In each of the 12 image groups $G_0, G_i, i \in M_1$ ($G_0 = T_1 \bigcup T_2 \bigcup T_3$, $G_i = T_1^{(i)} \bigcup T_2^{(i)} \bigcup T_3^{(i)}$, $i \in M_1$) for each quality factor $QF \in M_2$ used in repeated saving of the image, the average value of the number of DI 4×4-blocks obtained as a result of its standard splitting were calculated (as a percentage of the total block number) in which the normalized gap of the maximum SN increased. The results of the experiment, some of which are shown in Fig.3, fully confirm the theoretical conclusions obtained above.

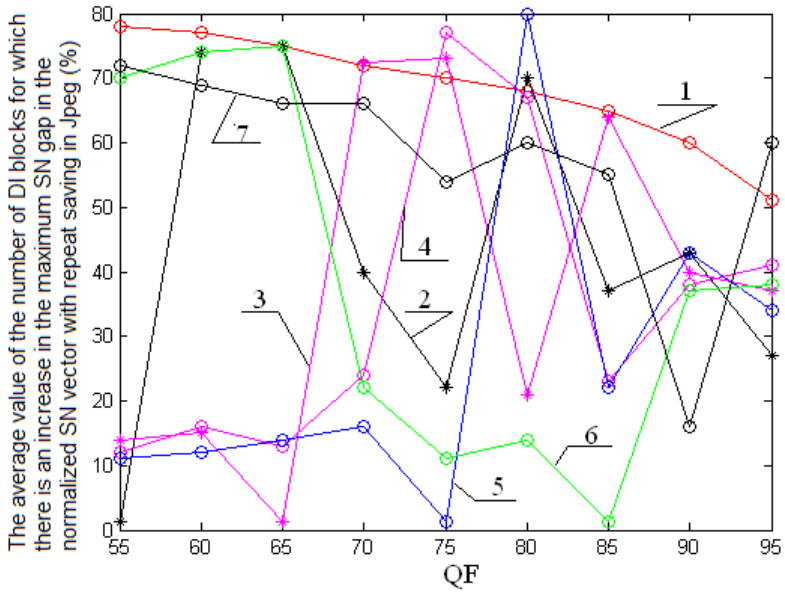


Fig. 3. Relationship between the average value of the DI block number, for which the normalized gap of the maximum SN increases as a result of DI repeated saving in Jpeg format, and the quality factor *QF* used for image repeated saving, when the original DIs were in the following formats: 1 – Tif; 2 – Jpeg with QF=55; 3 – Jpeg with QF=65; 4 – Jpeg with QF=72; 5 – Jpeg with QF=75; 6 – Jpeg with QF=85; 7 – Jpeg with QF=93

In the course of the experiment, it was revealed 6% of DIs from G_0, where a slight violation of the monotonous decrease in the block number occurred with an increased normalized gap of the maximum SN along with an increase in *QF* (a typical example is presented in Table 2). Obviously, this is a consequence of the features of machine arithmetic: calculations of SN blocks are performed in a set of floating-point numbers with an accumulation of computational error, which occurs here due to rounding. The possibility of such a situation is taken into account when developing the steganalysis method.

Table 2

Relationship between the relative number of blocks (% of the total number of DI 4×4-blocks obtained as a result of its standard splitting) with an increased normalized gap of the maximum SN and *QF* used for repeated saving, for a particular DI

QF								
55	60	65	70	75	80	85	90	95
67.66	67.06	66.45	**65.37**	**65.38**	65.00	63.29	58.40	42.76

The steganalysis method developed by the authors for detection of the steganographic transform results of any implementation of the LSB-method uses a DI in lossy format as a container, the expediency of which is justified above. After the embedding of an additional information, taking into account the well-known instability of the algorithmic implementations of the LSB method to attacks against the embedded message, the steganographic message is stored in a lossless format.

The main requirement for developed steganalysis method is the high efficiency in conditions of the low capacity of the hidden communication channel. The idea of this method is as follows. Due to the fact that in the conditions considered, the steganographic transform has little effect on the container, after the additional information embedding the properties of the DI steganographic message will not differ fundamentally from the properties of the used container. It means that for the matrix of the steganographic message, the dependence of the number of blocks, for which the normalized gap of the maximum singular number will increase, from the *QF* will not be monotonic.

In view of the above, the main steps of the proposed steganalysis method are as follows.

Let the DI with the matrix F_T be analyzed.

Step 1. Save the original DI in the lossy format – Jpeg with different quality factors $QF_i \in \{1,2,3,\ldots,100\}, i=\overline{1,t}$, $QF_i < QF_{i+1}, i=\overline{1,(t-1)}$. The result is the DI with matrices $F_i, i=\overline{1,t}$.

Step 2. For each pair of matrices F_T, $F_i, i=\overline{1,t}$, after the preliminary standard splitting them into non-intersecting $4\times 4-$blocks, determine the values $s_i, i=\overline{1,t}$, – is a number of blocks (a percentage of the total number of DI blocks) in $F_i, i=\overline{1,t}$, for which the normalized gap of maximum singular number increased compared to the corresponding blocks in F_T.

Step 3. *If*

$$\left((s_1 > s_2)\vee\left(0\le s_2 - s_1 \le P\right)\right)\wedge\ldots\wedge\left((s_{k-1} > s_k)\vee\left(0\le s_k - s_{k-1} \le P\right)\right)\wedge\ldots\wedge\left((s_{t-1} > s_t)\vee\left(0\le s_t - s_{t-1} \le P\right)\right),$$

(where P is the threshold value, which makes it possible to take into account the occurring monotony violations for the values $s_i, i=\overline{1,t}$, due to the peculiarities of machine arithmetic in the floating point system),

then

F_T – is an empty container

else

F_T – is steganographic message.

In the algorithmic implementation of the method, the following parameter values were used: $t=9$; $QF_i = 55+5(i-1), i=\overline{1,t}$; $P=1$.

To analyze the efficiency of the algorithmic implementation, a computational experiment conducted. During this experiment, the additional information was embedded by the LSB-method into DI-containers stored in Jpeg format. A randomly generated binary sequence was used as an additional information. The following values of the hidden communication channel capacity were used: 1, 0.1, 0.05, 0.01 bpp. As containers were used: 4950 images from the sets $T_1^{(i)}$, $T_2^{(i)}$, $T_3^{(i)}$, $i\in M_1$, and 350 DIs (size of 1000×1000 pixels) taken from the NRCS database [22] – the set T_J. In the course of the experiment, both the obtained steganographic messages and the original DIs from the sets T_1, T_2, T_3 were analyzed. The results of the computational experiment are given in Table 3 and Table 4 (the experiment-average value of Type II error is 2.7%). These results are illustrated on a specific example in Fig. 4, where 5 DI are presented in a lossless format, one of which is an original image, and the rest are the steganographic messages obtained by the LSB-method with different HCC values. The diagrams reflect how $s_i, i=\overline{1,9}$ depends on *QF*, used for repeated saving of the analyzed DI to Jpeg format, show that the monotony is present for the original DI (Fig. 4(f) curve 1), but there are monotony violations for DI steganographic messages (Fig. 4(f) curves 2-5).

For the convenience of comparing the efficiency of the method developed in the work with modern analogues, the obtained data were used to calculate the detection accuracy [13] (accuracy (ACC)) (Table 5):

$$ACC = (TP + TN)/(TP + FN + TN + FP), \quad (12)$$

where *TP* (*True Positive*) is the number of correctly identified steganographic messages (true positive result); *TN* (*True Negative*) is the number of correctly identified containers (true negative result); *FP* (*False Positive*) is the number of empty containers which were mistakenly qualified as steganographic messages (false positive (false alarm) or a Type II error); *FN* (*False Negative*) is the number of steganographic messages that are mistakenly identified as containers (false negative result Type I error).

Table 3

Type I errors of developed steganographic algorithm (%)

Container	**The capacity of a hidden communication channel (bpp)**			
	1	**0.1**	**0.05**	**0.01**
$T_1^{(95)}$	1.3	1.3	1.3	1.3
$T_1^{(93)}$	2	1.3	2	2
$T_1^{(90)}$	0.7	0	1.3	0.7
$T_1^{(85)}$	0.7	0.7	0.7	0.7
$T_1^{(80)}$	0.7	1.3	0.7	0
$T_1^{(75)}$	8	2	2	0.7
$T_1^{(72)}$	8	4	2	0.7
$T_1^{(70)}$	8	4	2	1.3
$T_1^{(65)}$	12	8	8.7	2
$T_1^{(60)}$	18.7	12	8.0	2
$T_1^{(55)}$	20	18.7	8.7	4
Image set average value ($T_1^{(i)}$, $i \in M_1$)	**7.3**	**4.8**	**3.4**	**1.4**
$T_2^{(95)}$	1.5	2	1	0.5
$T_2^{(93)}$	1.5	0	0.5	0
$T_2^{(90)}$	0	0	0	0
$T_2^{(85)}$	1	0	0.5	0.5
$T_2^{(80)}$	0	1	0.5	0
$T_2^{(75)}$	0	0	0	0
$T_2^{(72)}$	1	0	0	0
$T_2^{(70)}$	1.5	1	1	0
$T_2^{(65)}$	3.5	1	0.5	0.5
$T_2^{(60)}$	7	2	2	0.5
$T_2^{(55)}$	11.5	2.5	1	1
Image set average value ($T_2^{(i)}$, $i \in M_1$)	**2.6**	**0.9**	**0.6**	**0.3**
$T_3^{(95)}$	4	0	0	0
$T_3^{(93)}$	7	0	1	0
$T_3^{(90)}$	0	0	0	0
$T_3^{(85)}$	0	0	0	0

$T_3^{(80)}$	1	0	1	1
$T_3^{(75)}$	9	0	0	0
$T_3^{(72)}$	8	1	2	2
$T_3^{(70)}$	9	1	1	1
$T_3^{(65)}$	10	4	2	3
$T_3^{(60)}$	12	5	3	1
$T_3^{(55)}$	15	4	2	2
Image set average value ($T_3^{(i)}$, $i \in M_1$)	**6.8**	**1.4**	**1.1**	**0.9**
T_J	**0.9**	**0.6**	**0.6**	**0.9**
Experiment-average value	**4.8**	**2.2**	**1.5**	**0.8**

Table 4

Type II errors of developed steganographic algorithm (%)

Sets of original DIs		
T_1	T_2	T_3
0.7	0	11

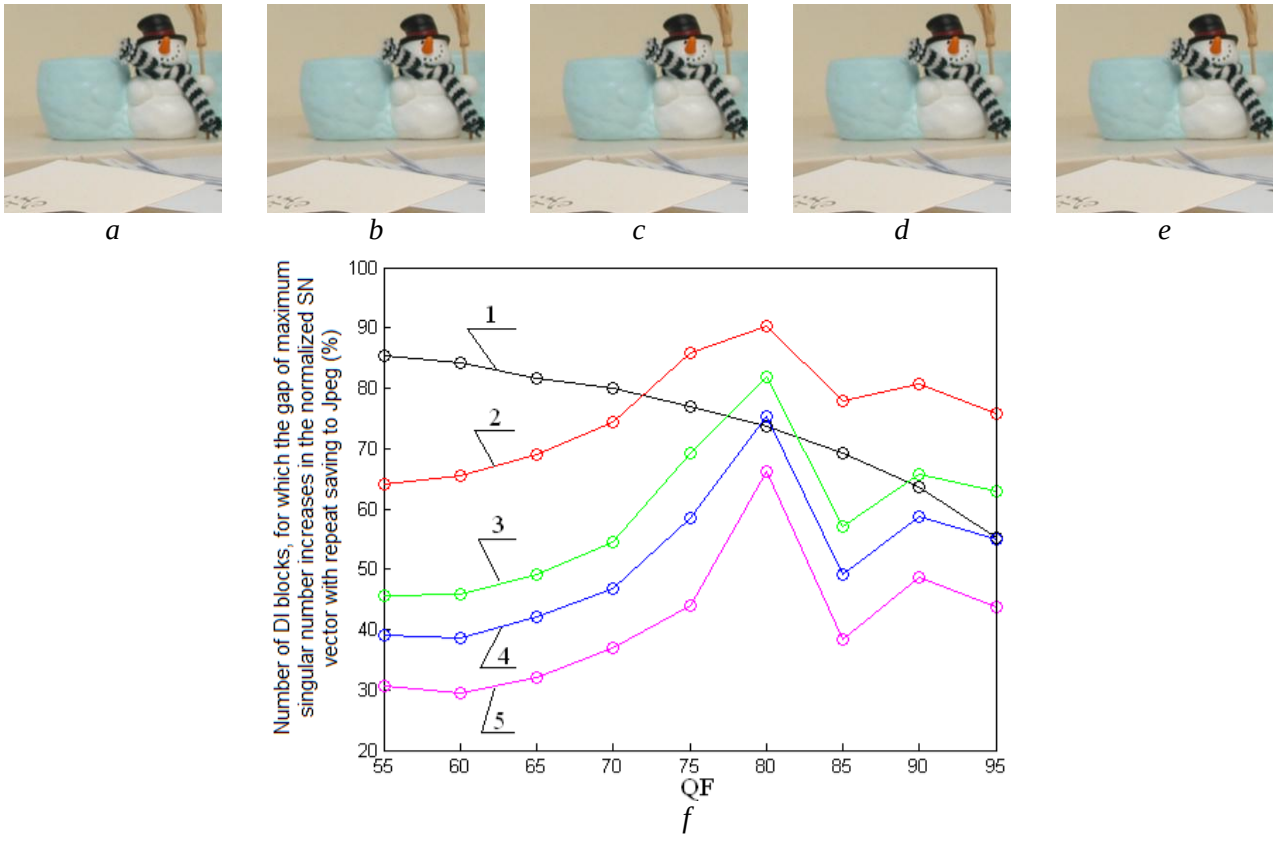


Fig. 4. Illustration of the steganalysis results obtained with developed algorithm for specific DIs: *a* – is an original DI; *b, c, d, e* – are steganographic messages, created by LSB-method with HCC 1, 0.1, 0.05, 0.01 bpp, respectively; *f* – diagrams that show how number of image blocks with an increased gap of maximum SN depends on *QF*, given that the image is converted for the second time to Jpeg format: 1 – for the original DI, 2,3,4,5 – for the steganographic messages, created by LSB-method with HCC 1, 0.1, 0.05, 0.01 bpp respectively

Table 5

The experiment-average (5750 DI) values of the detection accuracy coefficient for the developed algorithm, depending on the value of the HCC

The capacity of a hidden communication channel (bpp)			
1	0.1	0.05	0.01
0.954	0.978	0.987	0.991

The results of the computational experiment indicate a high efficiency of the algorithmic implementation of the developed method. The efficiency of the algorithm, as could be assumed on the strength of the theoretical basis of the developed method, increases with decreasing of HCC. Indeed, the smaller the HCC, the less perturbation is introduced into the DI-container by steganographic transform, the less DI steganographic message differs from the original container stored in the lossy format, the more effectively such steganographic message will be detected.

For a comparative analysis of the suggested algorithm efficiency, estimated by the ACC coefficient (12), modern analogous algorithms were chosen. These analogues are most efficient under the low HCC conditions and information about them is available from open sources: S1 (2006) [14], S2 (2006) [11], S3 (2008) [23], S4 (2009) [24], S5 (2010) [25], S6 (2015) [26], S7 (2015) [27], S8 (2016) [13], S9 (2016) [28]. The results are given in Table 6.

Table 6

Comparison of the developed algorithm efficiency, estimated using ACC, with modern analogues under the conditions of a low HCC

HCC, bpp	S1 (2006)	S2 (2006)	S3 (2008)	S4 (2009)	S5 (2010)	S6 (2015)	S7 (2015)	S8 (2016)	S9 (2016)	Our (2018)
0.1	0.9846	0.7727	0.9943	0.9937	0.9924	0.9971	0.988	0.9968	0.970	**0.978**
0.05	0.9769	0.6432	0.9283	0.9319	0.9404	0.9770	0.968	0.9865	0.941	**0.987**
0.01	0.5692	0.5094	-	-	-	-	-	-	-	**0.991**

Conclusions

As a result of the development of approach for detection of digital images integrity violations proposed by the authors earlier, the new steganalysis method has been created, as well as algorithm implementing it, which is effective in case of low hidden channel capacity, using LSB-method of embedding.

The computational complexity of the algorithm is determined by the number of non-intersecting 4×4-blocks into which the matrix of the analyzed DI is split, and in the case of its sizes of $n\times n$ pixels it is $\underline{O}\left(n^2\right)$ operations.

The developed algorithm significantly exceeds the existing analogues for HCC <0.1 bpp. With HCC = 0.01 bpp, this superiority over the best of its analogues (S1) is 74.1 %. In addition, it is workable for both color and grayscale DI, which often is not the case for analogues algorithms. In the case of an examination of color DI, 1, 2 or all color matrices (RGB scheme), brightness matrix (YUV scheme) will be analyzed.

References:

1. Karampidis K. A review of image steganalysis techniques for digital forensics / K. Karampidis, E. Kavallieratou, G. Papadourakis // Journal of Information Security and Applications. 2018. № 40. Pp. 217–235.
2. Saman Shojae Chaeikar. PSW statistical LSB image steganalysis / Saman Shojae Chaeikar, Mazdak Zamani, Azizah Bt Abdul Manaf, Akram M. Zeki // Multimedia Tools and Applications. 2018. Vol. 77, Iss. 1. Pp. 805–835.
3. Altaay A.A.J. An introduction to image steganography techniques / A.A.J. Altaay, S.B. Sahib, M.B. Zamani // Proceedings 2012 International Conference on Advanced Computer Science Applications and Technologies, ACSAT.
4. Li B. A survey on image steganography and steganalysis / B. Li, J. He, J. Huang, Y.Q. Shi // J. Inf. Hiding Multimedia Signal Process. 2011. No 2. Pp. 142–172.
5. Park T.H. Performance improvement of LSB-based steganalysis using bit-plane decomposition of images / T.H. Park, J.G. Han, Y.H. Moon, I.K. Eom // The Imaging Science Journal. 2016. No 64(5). Pp. 262–266.

6. Lerch-Hostalot D. LSB matching steganalysis based on patterns of pixel differences and random embedding / D.Lerch-Hostalot, D. Megias // Computers & Security. 2013. No 32. Pp.192–206.

7. Verma S. Relevance of steganalysis using DIH on LSB steganography / S. Verma, S. Sood, S.K. Ranade // International Journal of Advanced Research in Computer Science and Software Engineering. 2014. No 4(2). Pp. 835–838.

8. Xia Z. Steganalysis of LSB matching using differences between nonadjacent pixels / Z. Xia, X. Wang, X. Sun, Q.Liu, N.Xiong // Multimed Tools and Applications. 2016. No 75(4). Pp. 1947–62.

9. Lerch-Hostalot D. Unsupervised steganalysis based on artificial training sets / D. Lerch-Hostalot, D.Megías // Engineering Applications of Artifficial Intelligence. 2016. No 50. Pp. 45–59.

10. Juarez-Sandoval O. Compact image steganalysis for LSB-matching steganography / O. Juarez-Sandoval, M. Cedillo-Hernandez, G. Sanchez-Perez, K.Toscano-Medina, H. Perez-Meana, M.Nakano-Miyatake // In: Proceedings 2017 5th international workshop on biometrics and forensics (IWBF 2017). 2017. Pp. 1–6.

11. Zou D. Steganalysis based on Markov model of thresh-olded prediction-error image / D. Zou, Y.Q. Shi, W. Su, G.Xuan // 2006 IEEE international conference on multimedia and expo, ICME 2006 Proceedings, 2006. 2006. Pp. 1365–8.

12. Fridrich J. Steganalysis of content-adaptive steganography in spatial domain / J. Fridrich, J. Kodovský, V.Holub, M.Goljan // Lecture notes in computer science (including subseries on lecture notes in artificial intelligence lecture notes in bioinformatics (LNCS)). 2011. Vol. 6958. Pp. 102–17.

13. Ахмаметьева А.В. Стеганоанализ цифровых изображений, хранящихся в формате с потерями / А.В. Ахмаметьева // Захист інформації. 2016. Вип. 23. С.135-145.

14. Xiang-dong Chen. Detect LSB Steganography with Bit Plane Randomness Tests / Xiang-dong Chen, Feng Sun, Wei Sun // 6th World Congress on Intelligent Control and Automation, Dalian. 2006. Pp. 10306-10309.

15. Kobozeva A.A. General Principles of Integrity Checking of Digital Images and Application for Steganalysis / A.A. Kobozeva, I.I. Bobok, A.I. Garbuz // Transport and Telecommunication. 2016. Vol. 17, Iss. 2. Pp. 128-137.

16. Бобок И.И. Метод выявления изображений, пересохраненных в формат без потерь из формата с потерями // Математичне та комп'ютерне моделювання. 2017. Вип.16. С.5-14.

17. Деммель Д. Вычислительная линейная алгебра : теория и приложения / Пер. с англ. Х.Д. Икрамова. Москва : Мир, 2001. 430 с.

18. Кобозева А.А. Анализ информационной безопасности / А.А. Кобозева, В.А. Хорошко. Киев : ГУИКТ, 2009. 251 с.

19. Gloe T., Böhme R. The 'Dresden Image Database' for benchmarking digital image forensics. Proceedings of the 25th Symposium on Applied Computing (ACM SAC 2010). Sierre, 2010. Vol. 2, pp. 1585–1591.

20. Гонсалес Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс ; пер. с англ. П.А. Чочиа. Москва : Техносфера, 2006. 1070 с

21. Hsu Y.-F. Detecting image splicing using geometry invariants and camera characteristics consistency / Y.-F. Hsu , S.-F. Chang // Proceedings of the IEEE International Conference on Multimedia and Expo (ICME'06). Toronto, 2006, pp. 549-552.

22. NRCS Photo Gallery: [Електронний ресурс] // United States Department of Agriculture. Washington, USA. Electronic resource. Access mode: http://photogallery.nrcs.usda.gov (Дата обращения: 26.07.2012).

23. Chen C. JPEG Image Steganalysis Utilizing both Intrablock and Interblock Correlations / C. Chen, Y.Q. Shi // IEEE International Symposium on Circuits and Systems. Seattle, Washington, USA, 2008. Pp. 3029-3032.

24. Huang F. Calibration based universal JPEG steganalysis / F. Huang, J. Huang // Science in china series F: Information sciences. 2009. Vol. 52. No. 2. Pp. 260-268.

25. Pevny T. Steganalysis by subtractive pixel adjacency matrix / T. Pevny, P. Bas, J. Fridrich // IEEE Transactions on Information Forensics and Security. 2010. No. 2. Pp. 215-224.

26. Jinyang Su. Steganalysis using regional correlation and second-order Markov features / Su Jinyang, Zeng Xianting, Wang Lei // International Journal of Security and Its Applications. 2015. Vol. 9. № 1. Pp. 69-76.

27. B. Xue, X. Li, B. Li and Z. Guo. Steganalysis of LSB replacement for multivariate Gaussian covers // 2015 IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP), Chengdu, 2015, pp. 836-840. doi: 10.1109/ChinaSIP.2015.7230522.

28. Q. Lin, J. Liu and Z. Guo. Local ternary pattern based on path integral for steganalysis // 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, 2016, pp. 2737-2741.

Odesa National Polytechnic University *Received 21.04.2019*

UDC 681.3.06:519.248.681 **DOI:10.30837/rt.2019.3.198.03**

I.D. GORBENKO, Dr. of Sc, O.A. ZAMULA, Dr. of Sc, V.L. MOROZOV, S.V. RODIONOV, PhD

MATHEMATICAL MODEL OF ORTHOGONAL FREQUENCY SIGNALS DISTRIBUTION AND MULTIPLEXATION (OFDM)

Introduction

The need for new communication services is constantly increasing. At the same time the requirements for transmission speeds, noise-receiving of data, secrecy of functioning of the system, quality of services are provided. This leads to a worsening of the contradiction between the increasing requirements and the limited frequency resources, which in turn sets the task of increasing the spectral efficiency of the systems. It is known that a significant increase in pro-launch capability and reliability of communication can be achieved by using systems with multiple inputs and multiple outputs (MIMO), which uses multiple antennas on the transmitting side and multiple antennas on reception site. In combination with adaptive modulation and coding schemes, as well as the adaptive allocation of system resources, these methods can provide significant improvements in data rates and communication reliability [1]. Studies have shown that both MIMO technologies and other spectral efficiency enhancement techniques can be used in conjunction with multi-carrier transmission technologies, in particular, MIMO-OFDM (orthogonal frequency division multiplexing, hereinafter OFDM) and MIMO-OFDMA (orthogonal) technologies frequency division multiple access). Being able to provide subscribers with a wide range of applications with different capabilities in terms of tolerable delay, quality of service, bandwidth requires future systems of high resistance to interference and channel distortion, as well as more flexibility in radio resource management. Choosing the right radio interface is key to ensuring these properties of the communication system. Multiple Carrier Technology in orthogonal frequency division multiplexing is widely recognized as one of the most promising access schemes for use in advanced wireless communication systems

The main results of the research

The basic idea of OFDM is to split high-speed data flow into a number of sub-streams at lower speeds [1]. These sub-streams are then transmitted in parallel orthogonal subchannels, resulting in partial overlap of the spectrum. Compared to single-carrier transmission, this approach provides increased system resilience to narrowband interference and channel distortion. Moreover, this results in a high level of system flexibility, since modulation parameters such as constellation size, coding rate, manipulation sequence class, encoding method, character interleaving type, etc. can be independently selected for each subchannel. Problematic issues that limit, in some cases, the use of OFDM, include the significant magnitude of the peak of the emitted OFDM signal. As is known, the peak factor (PF) is defined as the ratio of the maximum (peak) instantaneous power of a signal to its average power. Increasing this parameter adversely affects the complexity of the high-frequency path design from amplifiers to the antenna, leading to a decrease in the efficiency of high-frequency equipment, to an increase in non-linear distortions. Synchronization, channel estimation, radio resource management are just some of the problems associated with multi-carrier data technology. In spite of the fact that OFDM provides high efficiency of spectrum utilization due to orthogonal frequency multiplexing [2], its out-of-band radiation may be unacceptable if the use of back-band is not foreseen. In particular, in 4G LTE about 10% of the dedicated bandwidth is reserved as a guard interval (also known as a cyclic prefix). This is quite a significant fee for a factor such as a spectrum resource. Frequency and time resources in OFDM are evenly divided into a number of elements of the same size for transmitting information [3]. In order to achieve orthogonality and to avoid inter-character or channel interference, it is necessary to ensure strict coordina-

tion of the operation (in time and frequency) of the elements of the transmitting and receiving sides of the system. The synchronization procedure results in intensive signal transmission to achieve perfect synchronization, especially in the case of uplink transmission. Improper synchronization can lead to suboptimal system performance. Fifth-generation (5G) systems offer many advantages over earlier systems, such as: high data rates, ultra-reliable low latency, high spectral efficiency, high subscriber connectivity, and enhanced energy efficiency. To take advantage of 5G, experts in the field and academia have proposed new and effective technologies based on OFDM modulation [4]: window-OFDM, Multiple Frequency Filter Group (FBMC); orthogonal frequency division encoded channel (C-OFDM) technology; Universal Filtered Multi-Carrier (UFMC); filtered-OFDM, FC-F-OFDM rapid-convolution system, etc. The filtration process is a proven and effective way of suppressing side lobes in OFDM. In the UFMC, filtering is applied to a block of sub-sequential subcarriers, which provides low out-of-band radiation. In multi-channel channels, UFMC is ineffective due to significant inter-character interference, which leads to suboptimal characteristics. In OFDM filtration systems, the available bandwidth is divided into many bands, enabling different sets of services to be implemented in different bands with signals filtered in the time domain accordingly. On the other hand, an approach based on filtering in the frequency domain was proposed, which has less computational complexity and increased flexibility compared to time domain filtering. The filter design is based on window optimization in the frequency domain, which balances the required minimum bandwidth attenuation, transition bandwidth, and error vector characteristics (EVM). OFDM is a type of frequency division multiplexing in which multiple subcarriers at adjacent frequencies are used in a single channel. The presence of multiple subcarriers in a single channel may create mutual interference, but due to the orthogonality of the subcarriers, this does not occur. For this reason, OFDM application maximizes the channel's spectral efficiency without interference. The spectrum of the OFDM system in the frequency domain is presented in Figure 1 [2].

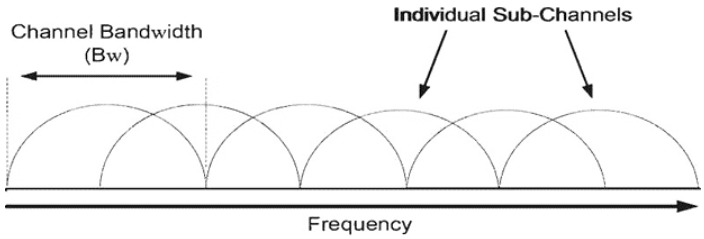


Fig. 1. OFDM frequency spectrum

Receiver / transmitter architecture in OFDM system

The classic OFDM transmitter and receiver model is shown in Fig. 2, *a* - OFDM transmitter model, Fig. 2, *b* - OFDM receiver. The transmitter converts the digital data that is to be transmitted to the corresponding amplitude and phase values, and then, using a backward Fourier transform (FFT), the digital data from the spectral representation is transformed into a time domain signal representation of the by adding a protective interval (CI). Thus, the received signal data is subjected to frequency multiplexing. The reverse operation is performed on the receiver side, as shown in Fig. 2, *b*. When the modulated OFDM signal arrives at the receiver, the radio frequency signal is summed up with the main carrier and the CO is deleted. The signal spectrum is then converted to the frequency domain using a Fourier transform (FFT). Then the subcarrier phase and amplitude are extracted and demodulated back into the digital data.

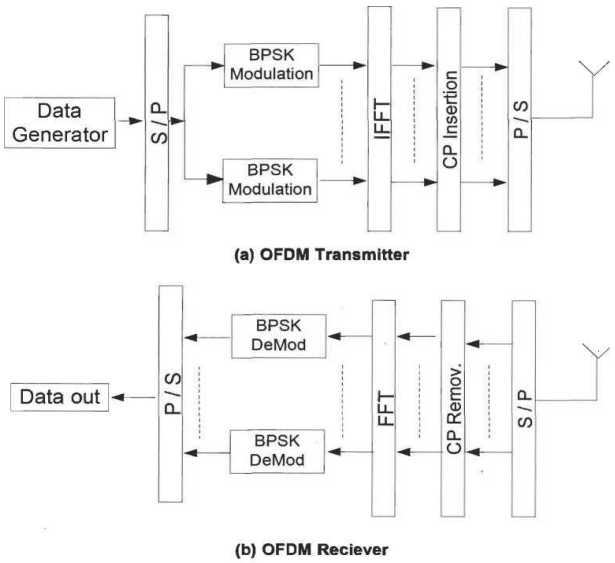


Fig. 2. Traditional OFDM system

In the transmitter, the output sequential stream of information bits is encoded by interference-tight code (according to LTE 3GPP TS 36.211 recommendation convolutional turbo codes are used with a base rate of 1/3), interspersed (P) and demultiplexed into N parallel sub-streams. Next, each of the streams is mapped to the symbol stream by phase pro-procedure (BPSK, QPSK, 8-PSK) or amplitude-phase quadrature modulation (QAM). When using BPSK modulation, a stream of binary numbers (1 and -1) is formed, with QPSK, 8-PSK, QAM a stream of complex numbers. In addition to the subcarriers on which the information is transmitted, service subcarriers are used. These include security intervals, pilots and additional service information to synchronize the receiver and transmitter and their operation modes. Pilots may have a fixed position on a subcarrier or a variable, with changes occurring from symbol to OFDM symbol in frames. Due to the insertion of inter-interconnections under channels of sufficient length of the protective interval, the possibility of spectral overlap is eliminated. In this case, the inter-channel interference (between bit interference, ICI) decreases, the probability of a bit error decreases, and thus the bandwidth of the wireless access system increases. The multiplication operation on a set exponent with the corresponding subchannel frequency and then summing all the sub channels to generate an OFDM signal is very similar to the Fourier Inverse Transform operation. In this regard, to form the required OFDM-symbol is used the device SHZPF, which greatly simplifies the implementation of modulators. Maintaining orthogonality is necessary in order for the receiver to correctly recognize the information on the subcarriers. To do this, you must complete the following conditions: the receiver and the transmitter must be precisely synchronized; the analog components of the transmitter and receiver must be of very high quality; the channel should not be multipath. In this case, multipath radiation is almost inevitable in radio communication systems, which leads to distortion of the received signal. To eliminate this kind of obstruction, you must select a protection interval that should be longer than the maximum propagation delay in the channel. Thus, it is possible to eliminate more than six types of interference between channels (ie interference between subcarriers) and between adjacent transmission units (ie between symbolic interference). To reduce the out-of-band emission of signals, a window processing of a temporal signal with the use of a window of the type "raised cosine" is used. Further, digital-to-analog converters (DACs) convert into an analog view separately the true and imaginary components. After passing through the low pass filter, the signal is fed to a quadrature mixer, which transfers the useful spectrum of the OFDM signal to the carrier frequency. These signals are further summed up, amplified and the OFDM signal is generated.

The use of a cycle prefix

In a wireless system, the radio signal in the transmission medium is reflected from different objects, causing multiple signals to be received at the receiver at different times. This phenomenon is known as multipath transmission. On the OFDM receiver side, the multipath propagation channel is represented as a distortion of the time at which the duration of each OFDM symbol increases [5]. As a result, the resulting symbols create obstacles to one another and form between symbolic obstacles [6]. Symbol rate for OFDM technology is much lower than for single carrier. For example, in a single-carrier system with BPSK modulation, the bit rate directly determines the transmission rate of sim-waves [7]. But in OFDM, the entire bandwidth is subdivided into N_f subcarriers, which leads to N_f - times a lower rate of symbol transmission than when transmitted from a single carrier. Thus, the effect of inter-character interference is reduced by multipath transmission from OFDM, which makes OFDM systems more resistant to ISI. The data transmission system can be improved by applying a buffer interval, which is a copy of a part of the transmitted signal of the OFDM symbol and this part is added to the beginning of the OFDM frame (Fig. 3). The use of a guard interval leads to an increase in the signal wavelength, but it significantly reduces the ISIs caused by multipath transmission [8-9].

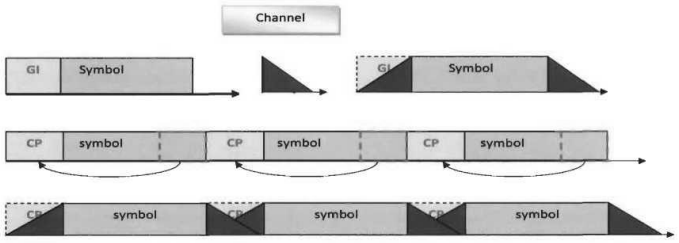


Fig. 3. Addition of CP to OFDM structure

With OFDM modulation, many subcarriers at adjacent frequencies are used in a single channel. This, in turn, can lead to mutual interference, but due to the orthogonality of the subcarriers, this does not happen [10]. Therefore, the application of OFDM to-maximize the spectral efficiency of the channel without interference. The spectrum of the OFDM system in the frequency domain is presented in Figure 3 [2].

Channel subcarriers orthogonality

As noted above, the subcarriers in OFDM systems are orthogonal. Thus, the sub-carriers are positioned as close as possible to one another, thereby increasing spectral efficiency. In other words, orthogonality provides simultaneous transmission to each subcarrier in the frequency space without interference [2] (Fig. 4). Thus, it is possible to detect signals on individual subcarriers in the receiving device. On the other hand, in a frequency-modulated conventional (FDM) system, such subcarrier overlap is impossible and, to avoid interference with the carriers, a protective band between the carriers is used. The OFDM multiuser version is OFDMA (Orthogonal Frequency Division Multiple Access) [5]. In this system, subsets of subcarriers are assigned to the individual user dynamically, using time or frequency division (Fig. 5), thus supporting simultaneous data transmission to multiple users. Using OFDMA, each user has his or her own unique set of subchannels (Fig. 5) and the base station can dynamically distribute subcarriers to users [4], for example, when a particular user has requested more resources. In essence, this means that this user may need higher radiation power, a large number of sub-channels, and appropriate modulation types.

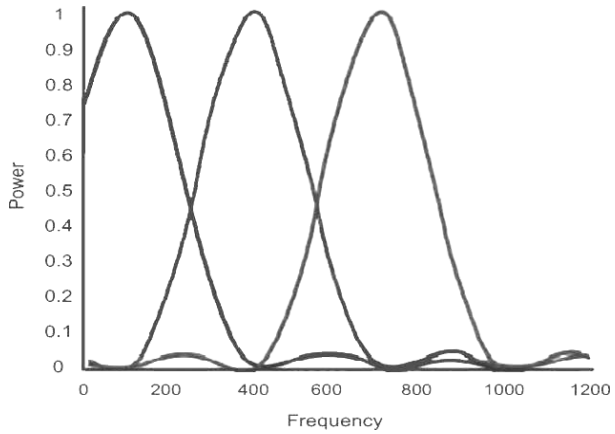


Fig. 4. Orthogonal intersection of the OFDM spectrum

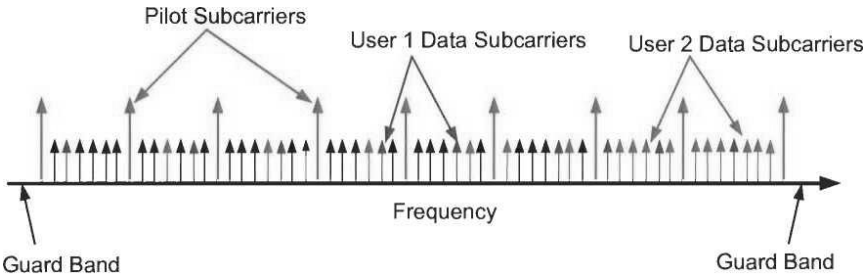


Fig. 5. Multiuser OFDM system

Thus, the main advantages of OFDMA include the following.

Resistance to fading.

Deployment flexibility in different frequency bands with small changes required for the radio interface.

Allows you to control channel or subchannel power.

By distributing carriers across the spectrum used, frequency diversity can be used.

Provides high signal quality when using a single carrier.

Mathematical models of OFDM and its versions

In the OFDM system in the converter (S/P) (Fig. 2) the serial information flow is transformed into N_f parallel flows, where N_f - the number of subcarriers before CP insertion. These parallel streams are then modulated using BPSK. With, every k symbol OFDM is given as [3]:

$$X_k(t) = \sum_{n=0}^{N_{f-1}} S_{k,n} p(t-kT) e^{j2\pi \frac{n}{T} t}, \quad (1)$$

where T - duration of OFDM symbol, and

$$S_k = \left[S_{k,0}, S_{k,1}, ..., S_{k,N_{f-1}} \right]^T, \quad (2)$$

Nf - parallel streams of data for one OFDM frame before inserting the CP,
T - the transposition operator, and p (t) is the impulse form used to form the symbols.

If we consider a rectangular pulse shape as a model:

$$p(t) = \begin{cases} 1, & 0 \le t \le T \\ 0, & otherwise \end{cases} \quad (3)$$

and provided that each subcarrier and OFDM symbol is selected N_f once per frame interval, the modulated signal (1) will look like:

$$X_k\left(\frac{mT}{N_f}\right)=\sum_{n=0}^{N_{f-1}} S_{k,n}e^{j2\pi nm/N_f}, m=0,1,2,...,N_f-1 \quad (4)$$

Further, IFFT is used to obtain the modulated signal:

$$X_k = N_f IFFT(S_k) \quad (5)$$

The thus obtained parallel stream of symbols after OFDM modulation is transformed into a sequential stream to which CP is added. The duration of IS to eliminate ISI should be greater than the channel delay. In the receiving device, the signal is converted from sequential to parallel, and the CP is removed, then FFT is used to demodulate and make decisions according to the type of modulation used.

MIMO-OFDM systems

In MIMO-OFDM systems, the wireless link is a transmitter and receiver system equipped with multiple antennas (Fig. 6). The main reason for the growing popularity of the MIMO-OFDM system is the ability to provide high quality signal and a high data rate. This is ensured by combining the signals on the transmit (T_X) antennas at one end and the receive (R_X) antennas at the other end. The resulting quality and performance improvements result in the communication lines being able to be used to significantly improve the quality of wireless network service.

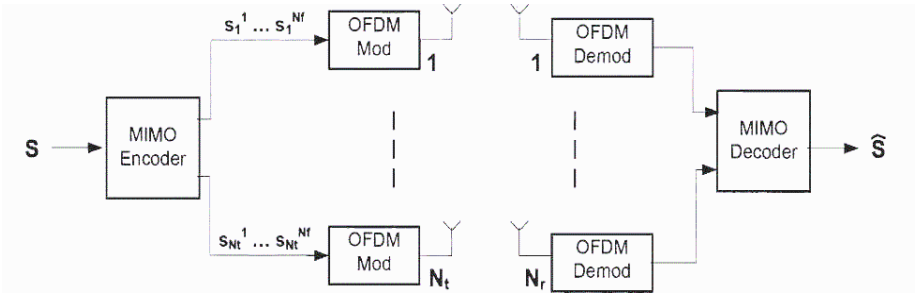


Fig. 6 A simplified block diagram of the MIMO-OFDM system

The MIMO-OFDM system involves the use of N_t transmitting antennas and N_r receiving antennas (Fig. 6). Initially, the input bitstream is displayed in several data characters using certain types of modulation, such as BPSK. Then block of N_s character data [$s_{1,}s_{2,}.,., s_{Ns}$] transforms into a codeword matrix S of size TXN_t, which then in T frames after implementation of OFDM modulation using N antennas will be transmitted to the communication channel. In this case, each frame consists of N_f subcarriers. Exactly, $S_{1j},S_{2j},\dots,S_{Tj}$ code words will be transmitted with each j_{th} transfer antennas in the form of 1,2,… T OFDM frames respectively. Code word S_{nj} means the vector of length N_f, for all j =1,2,…,N_t and n = 1,2,…,T. The codeword matrix S can be expressed as:

$$S=\begin{pmatrix} S_1^1 & \cdots & S_1^T \\ \vdots & \ddots & \vdots \\ S_{N_t}^1 & \cdots & S_{N_t}^T \end{pmatrix} \quad (6)$$

After adding a cyclic prefix in all OFDM frames, S_{nj} will be transmitted from the transmitting antenna at n_{th} OFDM frame. We define X as a subset of S, which represents an array of characters transmitted from all transmit antennas within a single OFDM frame, of dimension N_f X N_t:

$$X = \begin{pmatrix} S_1^1 & \cdots & S_1^{N_f} \\ \vdots & \ddots & \vdots \\ S_{N_t}^1 & \cdots & S_{N_t}^{N_f} \end{pmatrix} \quad (7)$$

Expression (7) can be written as follows:

$$X^{(k=1:N_f)} = \begin{bmatrix} X_1^{(k=1:N_f)} \\ X_2^{(k=1:N_f)} \\ \vdots \\ X_{N_t}^{(k=1:N_f)} \end{bmatrix}. \quad (8)$$

Here vector $X_j^{(k=1:N_f)}$ represents all characters transmitted from antenna j using N_f subcarriers. At the receiving station in the MIMO system, the received signals perform the procedures opposite to those implemented in the transmitting device. The protective terminal is removed and the FFT is applied and the data is then fed to the decoder. The resulting vector can be represented as:

$$Y^{(k=1:N_f)} = \begin{bmatrix} Y_1^{(k=1:N_f)} \\ Y_2^{(k=1:N_f)} \\ \vdots \\ Y_{N_r}^{(k=1:N_f)} \end{bmatrix} \quad . \quad (9)$$

Consider only the subcarrier (k = l). Then equation (9) can be represented as:

$$Y = HX + v, \quad (10)$$

where: Y - obtained vector, with dimension Nr;

H – Nr x Nt - complex signal propagation matrix, whose value is constant for the length of the transmitted frame (ie quasi-static channel) and is known in the receiver (for example, by transmitting test sequences). It is assumed that the statistics of the channel transmitter of the matrix H can be described by the attenuation statistics, namely: Rayleigh attenuation, Rice attenuation or AWGN. In addition, it is assumed that the elements H have a variance equal to one, or, in other words, the average gain of the channel P_c is normalized to one. Exposure to radio frequencies emitted by a transmitter by a counter-station may impair the characteristics of the receiving station's (AU) receiver. The level of interference power at the input of the AU receiver is usually set as [8] - [11]:

$$I = P_T - L_{FL.T} + G_T + G_R - L_{FL.R} - L_{POL} - L_P - L_{FDR}, \quad (11)$$

where P_T - transmitter output power in dB·W;

$L_{FL,T}$ - losses in the feeder line between the output of the transmitter and the input of the transmitting antenna;

G_T and G_R - gain of transmitting and receiving antennas;

$L_{FL,R}$ - losses in the feeder line between the output of the receiving antenna and the input of the receiver;

L_{POL} - loss due to mismatch of polarization of the receiving antenna;

L_P - propagation loss (including interference loss) between the transmitting and receiving antennas;

L_{FDR} - frequency-dependent deviation (FDR) losses.

FDR is a measure of the deviation caused by the selectivity curve of the receiver in the spectra of unwanted transmitter radiation, and can be represented as:

$$L_{FDR} = 10\log_{10}\left(\frac{\int_{-\infty}^{\infty}\Phi(f)df}{\int_{-\infty}^{\infty}\Phi(f)\Psi(f-\Delta f)df}\right), \quad (12)$$

where $\Phi(f)$ - power spectral density (PSD) of a complex equivalent representation of a fundamental frequency band (or complex envelope) of a true interference signal;

$\Psi(f)$ - normalized frequency response of the receiver;

Δf - frequency shift between the counter station transmitter and the AC receiver [11]. For signal signature $\Psi(f-\Delta f)$, if $\Delta f - \frac{W_v}{2} \le f \le \Delta f + \frac{W_v}{2}$, the expression for FDR is simplified to the form:

$$L_{FDR} = 10\log_{10}\left(\frac{\int_{-\infty}^{\infty}\Phi(f)df}{\int_{\Delta f-\frac{W_v}{2}}^{\Delta f+\frac{W_v}{2}}\Phi(f)df}\right), \quad (13)$$

where W_v - the bandwidth of the channel of the receiver AC.

The PSD signal area is equal to the signal power, and thus we obtain an expression to calculate the transmitter output power: $P_{T=10\log_{10}}\left(\int_{-\infty}^{\infty}\Phi(f)df\right)$.

As shown in [13], FDR mainly depends on the interference power spectral density (PSD).

CP-OFDM systems

The complex envelope signal transmitted in CP-OFDM technology [11] can be expressed as:

$$s(t) = \sum_{n=-\infty}^{\infty}\sum_{k=0}^{N-1} c_{n,k}\, p(t-n(T_s+T_g))e^{-j2\pi k\frac{1}{n}(T_s+T_g)}, \quad (14)$$

where $c_{n,k}$ - complex data symbol modulated on the k-th subcarrier of the n-th OFDM symbol,

p(t) - window of impulse formation,

$T_{tot} = T_s + T_g$ - total character length,

T_s and T_g - the length of the data symbol and the security interval, respectively.

Assuming that the complex signals on each subcarrier are statistically independent and mutually orthogonal, the expression for the power spectral density of the OFDM signal with arbitrary pulse formation is given as [11] - [13]:

$$\Phi_s(f) = \frac{P_s}{T_{tot}}\sum_{k=0}^{N-1}\left|P(f-\frac{k}{T_s})\right|^2, \quad (15)$$

where P_s - data symbols dispersion $c_{n,k}$, as well as the power of one of the OFDM subcarriers;

$1/T_s$ distance between subcarriers;

P(f) - Fourier transform impulse forming window.

Adding the representation of the rectangular momentum form to (14), we obtain:

$$p(t) = \Pi\left(\frac{t - \frac{T_{tot}}{2}}{T_{tot}}\right),$$

$$\partial e\,\Pi\left(\frac{t}{T_{tot}}\right) = \begin{cases} 0, if\ |t| > \frac{T_{tot}}{2} \\ \frac{1}{2}, if\ |t| = \frac{T_{tot}}{2} \\ 1, if\ |t| < \frac{T_{tot}}{2} \end{cases}. \qquad (16)$$

Using the shift time property of the Fourier transform, $|P(f)|^2$ is expressed as

$$\left|P(f)\right|^2 = \left|\Im\{p(t)\}\right|^2 = \left|\Im\left\{\Pi(\frac{t}{T_{tot}})\right\}e^{-j\pi_{tot}}\right|^2 = \left|\Im\left\{\Pi(\frac{t}{T_{tot}})\right\}\right|^2 = T_{tot}^2 \sin c^2(T_{tot} f), \qquad (17)$$

where sin *c* – a function that is defined as sin c (x)= sin (πx)/πx, if x ≠ 0, otherwise – equals one.

Using (15) and (17), the power spectral density (PSD) of CP-OFDM with a rectangular pulse shape is defined as:

$$\Phi_s^{(CP)}(f) = P_s T_{tot} \sum_{k=0}^{N-1}\left\{\sin c\left[\left(f - \frac{k}{T_s}\right)T_{tot}\right]\right\}^2. \qquad (18)$$

Let FDR be the measure of the deviation determined by the selectivity curve of the receiver on the spectrum of unwanted transmitter radiation. Applying (13) and (18), we obtain an expression to calculate the LFDR of CP-OFDM technology:

$$L_{FDR}^{(CP)} = P_T - 10\log_{10}\left(\frac{P_s}{\pi}\sum_{k=0}^{N-1}\left[\frac{\sin^2(f_k^-)}{f_k^-} - \frac{\sin^2(f_k^+)}{f_k^+} - Si(2f_k^-) + Si(2f_k^+)\right]\right), \qquad (19)$$

where

$$Si(x) = \int_0^x \frac{\sin t}{t}dt;$$

$$f_k^+ = \pi T_{tot}(\Delta f + \frac{W_v}{2} - \frac{k}{T_s});$$

$$f_k^- = \pi T_{tot}(\Delta f - \frac{W_v}{2} - \frac{k}{T_s}).$$

Windowed-OFDM systems

For the suppression of out-of-band radiation in the OFDM window system, window-time functions, for example, the function of the increased co-sine, are used to generate momentum. $w_{rc}(t)$, duration $T_w = T_{tot} + T_{tr}$ kind:

$$p(t) = w_{rc}\left(t - \frac{T_{tot}}{2}\right),$$

$$\partial e\, w_{rc}(t) = \begin{cases} 0, \ \dfrac{(T_{tot}+T_{tr})}{2} \le |t| \\ \dfrac{1}{2}\left(1+\cos\left(\dfrac{\pi\left(|t|-\dfrac{(T_{tot}+T_{tr})}{2}\right)}{T_{tr}}\right)\right), if\ |t| = \dfrac{T_{tot}}{2}, \\ 1, \ 0 \le |t| < \dfrac{T_{tot}-T_{tr}}{2} \end{cases} \quad (20)$$

T_{tr} - signal transmission time.

From (15) and the Fourier transform (20) the expression for the power spectral density of the OFDM window can be represented as:

$$\Phi_s^{(W)}(f) = P_s T_{tot} \sum_{k=0}^{N-1} \left\{ \sin c\left[\left(f \times \frac{k}{T_s}\right) T_{tot}\right] \times \frac{\cos\left(\pi T_{tr}\left(f - \dfrac{k}{T_s}\right)\right)}{1 - 4T_{tr}^{\,2}\left(f - \dfrac{k}{T_s}\right)^2} \right\}^2 . \quad (21)$$

It follows from (21) that the expression for the FDR of the OFDM window is defined as:

$$L_{FDR}^{(CP)} = P_T - 10\log_{10}\left[P_s T_{tot} \sum_{k=0}^{N-1} \int_{\Delta f - \frac{W_v}{2}}^{\Delta f + \frac{W_v}{2}} \left\{ \sin c\left(\left(f - \frac{k}{T_s}\right) T_{tot}\right)^2 \times \frac{\cos^2\left(\pi T_{tr}\left(f - \dfrac{k}{T_s}\right)\right)}{\left(1 - 4T_{tr}^{\,2}\left(f - \dfrac{k}{T_s}\right)^2\right)^2} \right\} df \right]. \quad (22)$$

The latter expression can be implemented using software for settlement calculations. As follows from expression (22), the FDR of the OFDM window is affected only by the parameters: T_{tr}, T_s, and T_{tot}.

Filtered OFDM systems

The filtered x(t) OFDM signal is formed by passing the s(t) signal CP-OFDM (6) through the spectrum forming filter. Thus, x(t) is given by the convolution s(t) and the impulse response of the filter h(t) in this way:

$$x(t) = s(t) \cdot h(t). \quad (23)$$

We apply a spectrum forming filter [9], which is based on the truncation of the basic filter. The truncation is performed by applying the window-time function w (t) to the impulse response of the main filter g (t). The impulse response of a truncated filter is defined as

$$h(t) = g(t) \cdot w(t). \quad (24)$$

Suppose a function sin c for $g(t) = W_g \sin c\left(W_g t\right)$, has a frequency response $W_g = \Pi\left(\dfrac{f}{W_g}\right)$. To suppress out-of-band radiation, the base filter truncates by using window-time functions such as the Henning, Hamming, and Blackman windows [11]. For example, a Henning window of T_w duration is defined as:

$$w(t)=\begin{cases}\frac{1}{2}+\frac{1}{2}\cos\left(\frac{2\pi|t|}{T_w}\right), & |t|\le\frac{T_w}{2}\\ 0, & |t|>\frac{T_w}{2}\end{cases}. \quad (25)$$

Filtered-OFDM PSD signal is given as

$$\Phi_x(f)=\frac{P_s T_{tot}}{16\pi^2}\left[2Si(\pi f_u)-2Si(\pi f_l)+Si(\pi-\pi f_l)-Si(\pi-\pi f_u)+Si(\pi+\pi f_u)-Si(\pi+\pi f_l)\right]^2 \times\sum_{k=0}^{N-1}\left\{\sin c\left[\left(f-\frac{k}{T_s}\right)T_{tot}\right]\right\}^2, \quad (26)$$

where $f_u = T_w(f + W_g/2)$, and $f_l = T_w(f - W_g/2)$.

It should be noted that the PSD of the filtered OFDM signal can be found in quasi-closed form, since $S_i(x)$ can be estimated as easily as the basic trigonometric function using numerical calculation software. Using (26), the FDR of filtered-OFDM is defined as:

$$L_{FDR}^{(F)}=P_T-10\log_{10}\left[\frac{P_S T_{tot}}{16\pi^2}\sum_{k=0}^{N-1}\int_{\Delta f-W_v}^{\Delta f+W_v}\sin c\left(\left(f-\frac{k}{T_s}\right)T_{tot}\right)^2\times\left\{\begin{matrix}2Si(\pi f_u)-2Si(\pi f_l)+Si(\pi-\pi f_l)\\ -Si(\pi-\pi f_l)+Si(\pi+\pi f_u)-Si(\pi+\pi f_u)\end{matrix}\right\}^2 df\right]. \quad (27)$$

This expression can be implemented programmatically for numerical calculations [11]. Analysis of the latter expression shows that the FDR of the filtered OFDM is affected by the parameters: T_w, Ts, T_{tot}, and W_g. This is important in terms of controlling inter-character and inter-channel interference in the development of OFDM system architecture.

Conclusions

The use of signals with orthogonal frequency division channels, allows to sub-increase not only the information capacity of the system in conditions of multipath propagation with a limited bandwidth, but also the speed of data transmission, bringing it closer to the bandwidth of the channel, increase the transmission secrecy and noise immunity, the effects of multipath propagation (provided the appropriate shielding interval is used), increase the spectral efficiency of the system. The article analyzes the systems of OFDM and a number of systems, which basically contain procedures of orthogonal time-division of channels and multiplexing. Mathematical models of OFDM signals are obtained, describing the basic stages of transformations that are performed to obtain such signals and estimating the properties of the signals. The obtained results can be used in the construction of secure information and communication systems, for which the primary tasks are to provide the necessary indicators of noise immunity (noise immunity of receiving signals, system operation), information security, resistance to interference, spectral and energy efficiency.

References:

1. Замула А.А., Морозов В.Л. Принципы применения технологии ортогонального частотного разделения каналов и мультиплексирования (OFDM) для построения современных беспроводных коммуникационных систем. Проблеми інформатизації // Матеріали шостої міжнародної науково-технічної конференції, 14 – 16 листопада 2018. – С. 23-24.

2. OFDM and Multi-Channel Communication Systems, National Instruments Measurement Fundamentals series, Publish Date: Feb 02, 2012.

3. Rodger Ziemer and William Tranter, Principles of Communications - Systems Modulation and Noise, fifth edition, John Wiley and Sons Ltd, NJ, 2002.

4. K. B. Letaief and Y. Zhang. Dynamic Multiuser Resource Allocation and Adaptation for Wireless Systems // IEEE Wireless Commun., vol. 13, no. 4, Aug. 2006, pp. 38-47.

5. S. Srikanth, V. Kumaran, C. Manikandan et al. Orthogonal Frequency Division Multiple Access: is it the multiple access system of the future. AU-KBC Research Center, Anna University, India.

6. A. D. S. Jayalath and C. Tellambura, ‘‘Reducing the out-of-band radiation of OFDM using an extended guard interval // IEEE 54th Veh. Technol. Conf. (VTC Fall), Atlantic City, NJ, USA, vol. 2, Oct. 2001, pp. 829–833.

7. C. Liu and F. Li. Spectrum modelling of OFDM signals for WLAN // Electron. Lett., vol. 40, no. 22, pp. 1431–1432, Oct. 2004.

8. Procedures for Determining the Potential for Interference Between Radars Operating in the Radiodetermination Service and Systems in Other Services, document ITU-R Rec. M.1461-2, Jan. 2018.

9. Frequency and Distance Separations, document ITU-R Rec. SM.337-6, Oct. 2008.

10. S. H. Raghavan and H. Chew. Frequency-dependent rejection, spectral separation coefficient, and interference analysis // IEEE Aerosp. Conf., Big Sky, MT, USA, Mar. 2018, pp. 1–10.

11. Park, Jaedon & Lee, Eunhyoung & Park, Sung-Ho & Sabogu-Sumah, Raymond & Pyo, Seongmin & Jo, Han-Shin. Modeling and Analysis on Radio Interference of OFDM Waveforms for Coexistence Study // IEEE Access. PP. 1-1. 10.1109/ACCESS.2019.2896280.

12. T. Levanen, J. Pirskanen, K. Pajukoski, M. Renfors, and M. Valkama. Transparent Tx and Rx waveform processing for 5G new radio mobile communications // IEEE Wireless Commun., vol. 26, no. 1, pp. 128–136, Feb. 2019. doi: 10.1109/MWC.2018.1800015.

13. Qualcomm. (2017). Spectrum for 4G and 5G. [Online]. Available: https://www.qualcomm.com/media/documents/files/spectrum-for-4ga.

Kharkiv National V.N. Karazin University;
JSC "Institute of Information Technologies"

Received 07.08.2019

УДК 004.056.5 **DOI:10.30837/rt.2019.3.198.04**

О.О. КУЗНЕЦОВ, д-р техн. наук, Ю.І. ГОРБЕНКО, канд. техн. наук,
В.В. ОНОПРІЄНКО, канд. техн. наук, І.В. СТЕЛЬНИК, Д.В. МЯЛКОВСЬКИЙ

АЛГОРИТМИ КРИПТОГРАФІЧНОГО ҐЕШУВАННЯ, ЯКІ ЗАСТОСОВУЮТЬСЯ В СУЧАСНИХ БЛОКЧЕЙН-СИСТЕМАХ

Вступ

Сучасні децентралізовані інформаційні системи та мережі, побудовані за новітньою технологією блокчейн, дедалі поширюються та застосовуються у різних додатках, наприклад при побудові криптовалют; для реалізації розподілених та захищених від несанкціонованої зміни реєстрів, кадастрів, списків, тощо; для побудови різних за призначенням та функціональними завданнями децентралізованих систем, які об'єднують, наприклад, центри сертифікації ключів, тощо; при побудові розподілених децентралізованих мереж електронної ідентифікації та електронного голосування; при розбудові інформаційних систем із підтримкою так званих смарт-контрактів, тощо. Отже, аналіз та дослідження всіх складових сучасних систем та мереж, які побудовано за технологією блокчейн, є актуальним та важливим науковим завданням.

За визначенням блокчейн являє собою вибудуваний за певними правилами безперервний послідовний ланцюжок блоків (або зв'язний список), що містить певну інформацію. Найчастіше копії ланцюжків блоків зберігаються на безлічі різних комп'ютерів незалежно один від одного [1]. Вперше цей термін з'явився як назва розподіленої бази даних, реалізованої в системі «біткойнів», через що блокчейн часто відносять до транзакцій в різних криптовалютах, проте технологія ланцюжків блоків може бути поширена на будь-які взаємопов'язані інформаційні блоки [1, 2]. Біткойн став лише першим застосуванням технології блокчейн в жовтні 2008 р. [2].

Для забезпечення захисту інформації від різних загроз безпеці в системах блокчейн застосовуються криптографічні методи, механізми та протоколи. Зокрема одним із головних криптопримітивів в кожній блокчейн-системі є алгоритми криптографічного ґешування [3], які призначені для перетворення вхідного масиву даних довільної довжини у вихідний бітовий рядок фіксованої довжини [4]. Такі перетворення також називаються ґеш-функціями, або функціями згортання, а їхні результати називають ґешем, ґеш-кодом, ґеш-сумою, або дайджестом повідомлення (англ. message digest) [5].

Криптографічні ґеш-функції мають наступні важливі властивості безпеки [6 – 8]:

1. Вони стійкі до знаходження прообразу. Це означає, що вони односторонні, тобто з математичної точки зору неможливо обчислити правильне вхідне значення при відомому вихідному значенні. Наприклад, якщо задане ґеш-значення *y*, тоді обчислювально важко знайти таке *x*, для якого hash(x) = y;

2. Стійкі до знаходження другого прообразу. Це означає, що ніхто не може знайти вхідне значення, яке ґешується у конкретний результат. Більш детально – криптографічні ґеш-функції створені таким чином, що при заданому конкретному вихідному значенні обчислювально неможливо знайти друге вхідне значення, яке дає таке ж вихідне значення. Наприклад, якщо задане *x*, обчислювально важко знайти таке *y*, для якого hash(x) = hash(y). Єдиний доступний підхід полягає у тому, щоб перебирати вихідні значення у всьому просторі, однак з обчислювальної точки зору немає жодного шансу на успіх;

3. Стійкі до колізій. Це означає, що неможливо знайти два вхідних значення, які б ґешувалися до однакового результату. Якщо розглядати більш детально, то з математичної точки зору обчислювально неможливо знайти два вхідні значення, які привели б до одного і того ж вихідного значення. Наприклад, обчислювально важко знайти такі x і y, де hash(x) = hash(y).

В багатьох реалізаціях блокчейну застосовується захищений ґеш-алгоритм (SHA) з розміром вихідного значення 256 біт (SHA-256) [6]. Багато комп'ютерів апаратно підтримують даний алгоритм, що прискорює його обчислення [6].

Втім, слід відмітити, що з появою ASIC стало можливим добувати криптовалюту (наприклад, Bitcoin) набагато швидше, ніж за допомогою відеокарт або десктопних обчислювальних систем [9 – 15]. ASIC – це інтегральна схема, спеціалізована для вирішення конкретного завдання. Ці схеми у багато разів вигідніше відеокарт, тому що при більшій потужності (швидкості розрахунку ґешу) вони споживають набагато менше енергії. Отже почалася, так би мовити, «гонка озброєнь»: розробники блокчейн-протоколів шукають способи протистояти ASIC-Майнінгу, а виробники майнінгового обладнання шукають можливість обійти хитрості розробників за допомогою застосування швидких ASIC-обчислювачів. Таким чином, інвестуючи в придбання ASIC, недобросовісні конкуренти можуть бути поставлені у свідомо більш вигідне становище порівняно з іншими гравцями [16 – 18]. Для захисту від ASIC-майнерів і забезпечення справедливого розподілу прибутку необхідно змінювати алгоритм ґешування або застосовувати принципово нові криптографічні схеми та протоколи консенсусу [15].

В статті проводиться аналіз функцій ґешування, які застосовуються або можуть застосовуватися в різних блокчейн-системах. Зокрема, розглядаються найбільш поширені національні та міжнародні стандарти, в яких наведено специфікацію всесвітньо відомих алгоритмів криптографічного ґешування, та досліджуються різні проекти з побудови децентралізованих блокчейн-систем, де ці функції можуть бути застосовані. В подальших статтях проводяться порівняльні дослідження функцій ґешування за швидкодією та статистичною безпекою.

Аналіз функцій ґешування, які застосовуються або можуть застосовуватися в різних блокчейн-системах

За визначенням, ґешування (або хешування, англ. hashing) є перетворенням вхідного масиву даних довільної довжини у вихідний бітовий рядок фіксованої довжини [5 – 8]. Такі перетворення також називаються ґеш-функціями, або функціями згортання, а їхні результати називають ґешем, ґеш-кодом, ґеш-сумою, або дайджестом повідомлення (англ. message digest) [6]. Отже функція ґешування – це функція, що перетворює вхідні дані будь-якого (як правило, великого) розміру в дані фіксованого розміру. За визначенням з ДСТУ 7564:2014 «Інформаційні технології. Криптографічний захист інформації. Функція ґешування» під функцією ґешування розуміється криптографічне перетворення повідомлення M довжини від 0 біт (порожній рядок) до $2^{96}-1$ біт у ґеш-значення (ґеш-вектор) $H(M)$, що є двійковим рядком фіксованої довжини n ($n = 8 \cdot s, s \in \{1,2,...,64\}$) [19].

Основним міжнародним нормативним документом, який визначає терміни, основні поняття, класифікацію та специфікацію певних алгоритмів криптографічного ґешування, є міжнародний стандарт ISO/IEC 10118 [20 – 23]:

- в першій частині стандарту ISO/IEC 10118-1:2016 «Information technology – Security techniques – Hash-functions – Part 1: General» наводяться основні поняття та визначення з ґешування інформації, зокрема загальна ітеративна модель ґеш-функції (перша частина стандарту гармонізована в Україні у вигляді ДСТУ ISO/IEC 10118-1:2018 (ISO/IEC 10118-1:2016, IDT) «Інформаційні технології. Методи захисту. Ґеш-функції. Частина 1. Загальні положення» [20]);
- в другій частині ISO/IEC 10118-2:2010/Cor.1:2011 «Information technology – Security techniques – Hash-functions – Part 2: Hash-functions using an n-bit block cipher» визначаються алгоритми ґешування, які застосовують блокові симетричні шифри (ця частина стандарту гармонізована в Україні у вигляді ДСТУ ISO/IEC 10118-2:2015 (ISO/IEC 10118-2:2010; Cor 1:2011, IDT) «Інформаційні технології. Методи захисту. ґеш-функції. Частина 2. ґеш-функції, що використовують n-бітний блоковий шифр» [21]);

- третю частину ISO/IEC 10118-3:2018 «IT Security techniques – Hash-functions – Part 3: Dedicated hash-functions» присвячено розгляду спеціалізованих функцій ґешування (гармонізовано в Україні у вигляді ДСТУ ISO/IEC 10118-3:2005 «Інформаційні технології. Методи захисту. ґеш-функції. Частина 3. Спеціалізовані ґеш-функції» [22]);

- четверта частина ISO/IEC 10118-4:1998/Cor.1:2014 «Information technology – Security techniques – Hash-functions – Part 4: Hash-functions using modular arithmetic» містить опис функцій ґешування, які засновано на модулярній арифметиці (гармонізовано в Україні у вигляді ДСТУ ISO/IEC 10118-4:2015 (ISO/IEC 10118-4:1998; Cor 1:2014; Amd 1:2014, IDT) «Інформаційні технології. Методи захисту. ґеш-функції. Частина 4. ґеш-функції, що використовують модульну арифметику» [23]).

Функції ґешування, які описані у низці стандартів ISO/IEC 10118, не використовують секретного ключа (тобто є безключовими ґеш-функціями), зокрема вони можуть бути використані для формування кодів виявлення маніпуляцій (КВМ) (від англ. manipulation detection code – MDC).

Слід зазначити, що окремі криптографічні функції ґешування можуть також використовувати секретний ключ (тобто бути т.з. ключовими ґеш-функціями). Такі функції ґешування призначені для формування кодів автентифікації повідомлень (КАП) (від англ. message autentifacation code – MAC). Для їхнього опису та стандартизації на міжнародному рівні застосовується інший нормативний документ, а саме ISO/IEC 9797-1:2011 Information technology – Security techniques – Message Authentication Codes (MACs) [24 – 26]:

- перша частина ISO/IEC 9797-1:2011 «Information technology – Security techniques – Message Authentication Codes (MACs) – Part 1: Mechanisms using a block cipher» встановлює алгоритми формування КАП із застосуванням блокових симетричних шифрів (гармонізовано в Україні у вигляді ДСТУ ISO/IEC 9797-1:2015 (ISO/IEC 9797-1:2011, IDT) «Інформаційні технології. Методи захисту. Коди автентифікації повідомлень (MACs). Частина 1. Механізми, що використовують блоковий шифр» [24]);

- друга частина ISO/IEC 9797-2:2011 «Information technology – Security techniques – Message Authentication Codes (MACs) – Part 2: Mechanisms using a dedicated hash-function» містить специфікацію КАП із застосуванням спеціалізованих функцій ґешування (гармонізовано в Україні у вигляді ДСТУ ISO/IEC 9797-2:2015 (ISO/IEC 9797-2:2011, IDT) «Інформаційні технології. Методи захисту. Коди автентифікації повідомлень (MACs). Частина 2. Механізми що використовують спеціалізовану ґеш-функцію» [25]);

- третю частину ISO/IEC 9797-3:2011 «Information technology – Security techniques – Message Authentication Codes (MACs) – Part 2: Mechanisms using a universal hash-function» присвячено КАП, які застосовують універсальне ґешування (цей стандарт гармонізовано в Україні у вигляді ДСТУ ISO/IEC 9797-2:2015 (ISO/IEC 9797-2:2011, IDT) «Інформаційні технології. Методи захисту. Коди автентифікації повідомлень (MACs). Частина 2. Механізми, що використовують універсальну ґеш-функцію» [26]).

Таким чином, відповідно до діючих та гармонізованих в Україні міжнародних нормативно-правових документів загальну класифікацію криптографічних функцій ґешування можна подати у вигляді схеми, яку наведено на рис. 1. На рисунку штриховкою відмічені алгоритми, які не включено до відповідних стандартів, але які можуть бути застосовані для формування ґеш-кодів за відповідною схемою. Наприклад, відповідно до ISO/IEC 10118-2 ґешування може бути реалізоване із застосуванням блокового симетричного шифру. У якості такого шифру може бути застосований і алгоритм Калина (англ. Kalyna) – національний стандарт блокового симетричного криптоперетворення України [27, 28].

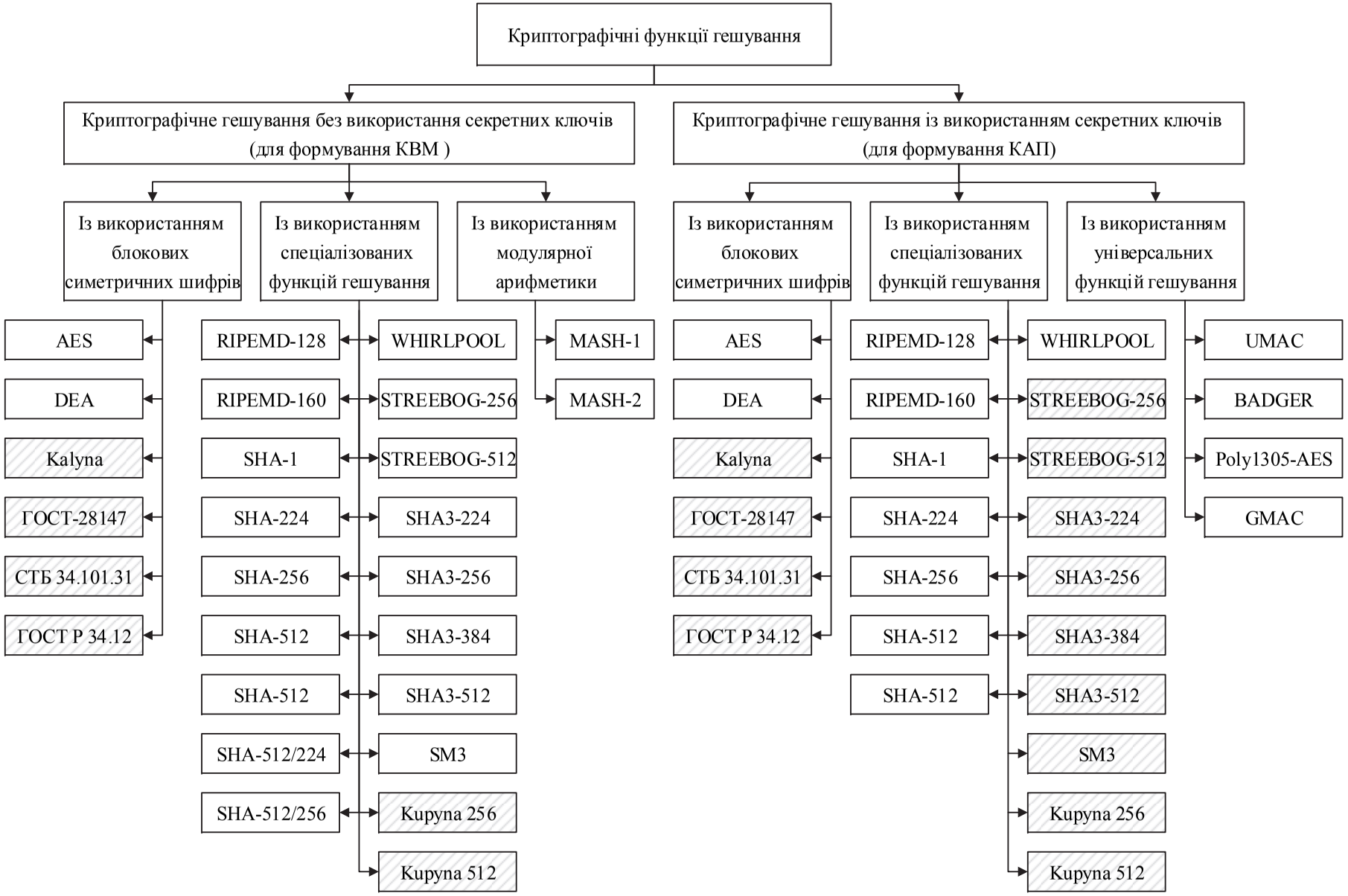


Рис. 1. Загальна класифікація криптографічних функцій гешування
(згідно з міжнародними стандартами ISO/IEC 10118 та ISO/IEC 9797)

Інший приклад – формування КАП. Відповідно до ISO/IEC 9797-1 КАП можуть формуватися із використанням алгоритмів блокового симетричного шифрування. І хоча за специфікацією ISO/IEC 9797-1 не передбачено використання алгоритму Калина, цей шифр може бути застосовано у відповідному режимі для формування КАП (за специфікацією ДСТУ 7624:2014 в алгоритмі Калина передбачена можливість формування КАП [27]).

До загальної класифікації, яку наведено на рис. 1, не входять чисельні алгоритми ґешування, які стандартизовано на національному рівні окремих країн, та алгоритми ґешування, які було подано та розглянуто на різних криптографічних конкурсах. Зокрема, на відкритому конкурсі «SHA-3», який проводився в 2007 – 2012 рр. Національним інститутом стандартів і технологій (NIST) на нову криптографічну ґеш-функцію, призначену для доповнення і заміни SHA-1 і SHA-2, було представлено велику кількість алгоритмів ґешування, з яких 51 алгоритм був допущений до проведення першого туру [29]. У табл. 1 представлені відомі учасники конкурсу «SHA-3» із зазначенням основних атрибутів ґеш-функцій і знайдених атак [30].

У табл. 1 застосовуються такі позначення [30]:

– FN (англ. A Feistel network) – мережа Фейстеля;

– WP (англ. Wide Pipe design) – метод побудови криптографічних ґеш-функцій, схожий на структуру Меркле – Дамгора;

– KEY (англ. Key schedule) – алгоритм, який одержує ключі для кожного раунду ґешування;

– MDS (англ. MDS Matrix) – розмір MDS-матриці;

– OUT (англ. Output Transformation) – криптографічна операція, яка здійснюється в останній вихідний ітерації;

– SBOX (англ. S-box) – S-блоки;

– FSR (англ. Feedback Shift Register) – регістр зсуву з лінійним зворотним зв’язком;

– ARX (англ. Addition Rotation XOR) – складання, циклічний зсув і XOR;

– BOOL (англ. Boolean operations) – булева алгебра;

– COL (англ. Collision Attack) – найкраща з відомих атак на пошук колізій, краще ніж атака «днів народження»;

– PRE (англ. Preimage Attack) – друга найкраща атака на пошук колізій, краще ніж атака подовженням повідомлення.

До переліку алгоритмів ґешування слід добавити спеціально розроблені функції ґешування для застосування в різних криптовалютах та інформаційних системах типу блокчейн. Зокрема, у табл. 2 наведено неповний перелік криптовалют із зазначенням року введення, спеціального позначення (тікера[1]) криптовалюти та алгоритму майнінгу[2].

[1] Тікер, Тікерна назва (англ. ticker symbol) – коротка назва котируваних інструментів (акцій, облігацій, індексів) в біржовій інформації. Є унікальним ідентифікатором в межах однієї біржі або інформаційної системи. Використовується для того, щоб постійно не друкувати в звітах та новинах повне найменування цінних паперів або інших об'єктів торгівлі.

[2] Майнінг, також видобування (від англ. mining – видобуток корисних копалин) – діяльність з підтримки розподіленої платформи і створення нових блоків з можливістю отримати винагороду в формі емітованої валюти і комісійних зборів у різних криптовалютах, зокрема в Біткоїнах. Обчислення потрібні для забезпечення захисту від повторного використання одних і тих же одиниць валюти, а зв'язок майнінгу з емісією стимулює людей витрачати свої обчислювальні потужності і підтримувати роботу мереж.

Таблиця 1

Відомі учасники конкурсу «SHA-3» із зазначенням основних атрибутів геш-функцій і знайдених атак [30]

Алгоритм гешування	FN	WP	KEY	MDS	OUT	SBOX	FSR	ARX	BOOL	COL	PRE
Abacus	-	X	-	4 x 4	X	8 x 8	X	-	-	2^{172}	2^{172}
ARIRANG	X	X	X	4 x 4, 8 x 8	-	8 x 8	-	-	-	-	-
AURORA	-	-	X	4 x 4	X	8 x 8	-	-	-	$2^{234,61}$ / $2^{229,6}$	2^{291} / $2^{31,6}$
BLAKE	X	-	X	-	-	-	-	X —	-	-	-
Blender	-	X	-	-	-	-	-	X	-	$10 \cdot 2^{\frac{n}{4}}$	$10 \cdot 2^{\frac{n}{4}}$
BMW	-	X	X	-	-	-	-	X	-	-	-
Boole	-	-	-	-	X	-	X	-	$\wedge$	2^{34}	$2^{\frac{9n}{r}}$
Cheetah	-	-	X	4 x 4, 8 x 8	-	8 x 8	-	-	-	-	-
Chi	X	X	X	-	-	4 x 3	-			-	-
CRUNCH	X	-	X	-	-	8 x 1016	-	-	-	-	-
CubeHash8/1	-	-	-	-	-	-	-	X	-	-	2^{509}
DHC	-	-	X	-	-	8 x 8	-	-	-	2^{9}	2^{9}
DynamicSHA	X	-	X	-	-	-	-			2^{114}	-
DynamicSHA2	X	-	X	-	-	-	-			-	-
ECHO	-	X	-	4 x 4	-	8 x 8	-	-	-	-	-
ECOH	-	-	X	-	-	-	-	-	-	-	-
Edon-R	-	X	X	-	-	-	-	X	-	-	$2^{\frac{2n}{3}}$
EnRUPT	-	X	-	-	-	-	-	X	-	-	2^{480} / 2^{480}
Essence	-	-	-	-	-	-	X	-	-	-	-
FSB	-	X	-	-	X	-	-	-	-	-	-
Fugue	-	X	-	4 x 4	X	8 x 8	-	-	-	-	-
Gr0stl	-	X	-	8 x 8	X	8 x 8	-	-	-	-	-
Hamsi	-	-	X	-	-	4 x 4	-	-	-	-	-
JH	X	X	-	1.5 x 1.5	-	4 x 4	-		-	-	$2^{510,3}$ / $2^{510,3}$
Keccak	-	X	-	-	-	-	-			-	-
Khichidi-1	-	-	X	-	-	-	X	-	-	1	$1/2^{33}$

Алгоритм гешування	FN	WP	KEY	MDS	OUT	SBOX	FSR	ARX	BOOL	COL	PRE
LANE	-	-	X	4 x 4	X	8 x 8	-	-	-	-	-
Lesamnta	X	-	X	2 x 2, 4 x 4	X	8 x 8	-	-	-	-	-
Luffa	-	-	-	-	X	4 x 4	-	-	-	-	-
Lux	-	X	-	4 x 4 , 8 x 8	X	8 x 8	-	-	-	-	-
MCSSHA-3	-	-	-	-	-	-	X	-	-	$2^{\frac{3n}{8}}$	$2^{\frac{3n}{4}}$
MD6	-	X	-	-	-	-	X			-	-
MeshHash	-	-	-	-	X	8 x 8	-	-	-	-	$2^{323,2} / 2^{\frac{n}{2}}$
NaSHA	X	-	-	-	-	8 x 8	X	-			-
SANDstorm	-	-	X	-	-	8 x 8	-			-	-
Sarmal	X	-	-	8 x 8	-	8 x 8	-	-	-	-	$2^{384} / 2^{128}$
Sgail	-	X	X	8 x 8, 16 x 16	-	8 x 8	-	X	-	-	-
Shabal	-	-	X	-	-	-	X			-	-
SHAMATA	X	X	X	4 x 4	-	8 x 8	-	-	-	$2^{40} / 2^{29}$	$2^{461,7} / 2^{462,7}$
SHAvite-3	X	-	X	4 x 4	-	8 x 8	X	-	-	-	-
SIMD	X	X	X	TRSC+	-	-	-			-	-
Skein	X	X	X	-	X	-	-	X	-	-	-
Spectral Hash	-	-	-	-	X	8 x 8	-	-	-	-	-
StreamHash	-	-	-	-	-	8 x 8	-	-	-	-	$n2^{\frac{n}{2}}$
SWIFFTX	-	-	-	-	-	8 x 8	-	-	-	-	-
Tangle	-	X	X	-	-	8 x 8	-				-
TIB3	U	-	X	-	-	3 x 3	-	-	-	-	-
Twister	-	X	-	8 x 8	X	8 x 8	-	-	-	2^{252}	$2^{448} / 2^{264}$
Vortex	-	-	-	4 x 4	X	8 x 8	-	-	-	$2^{125,5} / 2^{125,5}$	$2^3 / 2^{\frac{n}{4}}$
WAMM	-	X	-	-	X	8 x 8	-	-	-	-	-
Waterfall	-	X	-	-	X	8 x 8	X	-	-	2	-

Таблиця 2

Неповний перелік криптовалют та алгоритмів їхнього майнінгу [31]

Найменування	Рік	Тікер	Алгоритм майнінгу
Bitcoin	2009	BTC	SHA-256
Ethereum	2015	ETH	Dagger- Hashimoto
Steemit	2016	STEEM	SHA-256
Ripple	2013	XRP	ECDSA
DigiByte	2014	DGB	SHA256
Monero	2014	XMR	CryptoNight
Siacoin	2015	SC	blake2b
Litecoin	2011	LTC	Scrypt
EthereumClassic	2015	ETC	Dagger- Hashimoto
Dogecoin	2013	DOGE	Scrypt
NEM	2015	XEM	blockchain
Syscoin	2014	SYS	Scrypt
Augur	2015	REP	Smart contract
Dash	2014	DASH	X11
ByteCoin	2012	BCN	CryptoNight
BelaCoin	2014	BELA	Scrypt
lbryCoin	2016	LBC	LBRY
Radium	2015	RADS	Smartchain
Decred	2015	DCR	Blake256
Einsteinium	2014	EMC2	Scrypt
Gridcoin	2013	GRC	Scrypt
Primecoin	2013	XPM	1CC/2CC/TWN
NEO	2014	NEO	SHA-256 & RIPEMD160
MazaCoin	2014	MZC	SHA-256d
Titcoin	2014	TIT	SHA-256d
Verge	2014	XVG	Scrypt, x17, groestl, blake2s, and lyra2rev2
Stellar	2014	XLM	Stellar Consensus Protocol (SCP)
Tether	2015	USDT	Omnicore
Zcash	2016	ZEC	Equihash
Bitcoin Cash	2017	BCH	SHA-256d
EOS.IO	2017	EOS	–
VertCoin	2014	VTC	Lyra2RE
Dashcoin	2014	DSH	CryptoNight
Potcoin	2014	POT	Scrypt
Peercoin	2012	PPC	SHA-256
Namecoin	2011	NMC	SHA-256
Nxt	2013	NXT	SHA-256d
Nautiluscoin	2014	NAUT	NXT
Auroracoin	2014	AUR	Scrypt
Expanse	2015	EXP	Dagger- Hashimoto
PinkCoin	2014	PINK	X11
FoldingCoin	2014	FLDC	Stanford Folding
Navcoin	2014	NAV	X13
ViaCoin	2014	VIA	Scrypt
DNotes	2014	NOTE	Scrypt
Vcash	2014	XVC	Blake256

Висновки

Аналіз відомих розподілених технологій та блокчейн-мереж показує, що одним із їх головних криптографічних компонентів є ґешування. Саме на використанні криптографічних властивостей необоротності функцій ґешування будуються безперервні послідовні ланцюжки блоків (зв'язні списки), які дозволяють забезпечити надійне зберігання критично важливої інформації. Дійсно, при виконанні певних умов внесення несанкціонованих змін у захищену таким чином інформацію є практично неможливим, бо це порушить безперервність ланцюжка ґеш-значень і стане наявним для всіх користувачів блокчейн-системи. Отже, обрання функції ґешування для побудови зв’язаних списків, вивчення її властивостей та дослідження певних характеристик є дійсно важливою та актуальною науковою задачею.

З урахуванням можливого застосування ASIC-обчислювачів задача обрання надійної криптографічної функції ґешування ще більш ускладнюється. Дійсно, якщо певним гравцям блокчейн-мережі із визначеною криптографічною функцією вдастся першими ввести в дію значну кількість ASIC-майнерів, тоді вони отримують перевагу у формуванні наступних блоків мережі і, таким чином, зможуть нав’язувати свої рішення іншим учасникам системи. Отже в сучасних блокчейн-мережах необхідним є або заміна/модернізація протоколів консенсусу з метою зменшення можливого впливу ASIC-майнерів, або пошук таких алгоритмів ґешування, які б було складно відтворити у ASIC-обчислювачах. Саме тому на сьогодні спостерігається стрімкий зріст кількості різних алгоритмів ґешування, які застосовуються в різних блокчейн-системах.

В роботі розглянуто різні децентралізовані блокчейн-системи та проаналізовано застосовані в них алгоритми криптографічного ґешування. Слід зазначити, що перелік криптовалют та інформаційних систем за технологією блокчейн стрімко зростає. Наприклад, на середину 2018 р. неповний перелік криптовалют містив вже понад 1500 найменувань, і впродовж часу постійно оновлюється та розширюється. Практично неможливо відслідкувати всі діючі проекти з розробки технології блокчейн, але табл. 2 містить найбільш відомі та найпоширеніші криптовалюти світу.

Перспективним напрямком подальших досліджень є аналіз структури та особливостей застосування різних сімейств криптографічних функцій ґешування в блокчейн-системах, проведення порівняльних досліджень їх швидкодії та статистичної безпеки, що буде розглянуто у наступних роботах.

Список літератури:

1. Melanie Swan. Blockchain: Blueprint for a New Economy. O'Reilly Media, Inc., 2015. 152 p.
2. Marco Iansiti and Karim R. Lakhani (2017). The Truth About Blockchain // Harvard Business Review. January–February 2017 issue. Pp. 118-127.
3. Dylan Yaga, Peter Mell, Nik Roby, Karen Scarfone. NISTIR 8202 Blockchain Technology Overview // National Institute of Standards and Technology, Internal Report 8202, 66 pages (October 2018). https://doi.org/10.6028/NIST.IR.8202
4. ISO/IEC 10118-1:2016. Information technology – Security techniques – Hash-functions. Part 1: General. (2016-10), 12 p. https://www.iso.org/standard/64213.html
5. Handbook of Applied Cryptography / Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanstone. October 1996, 816 pages, Fifth Printing (August 2001). http://cacr.uwaterloo.ca/hac/
6. Dylan Yaga, Peter Mell, Nik Roby, Karen Scarfone. NISTIR 8202. Blockchain Technology Overview. [online] Available at: https://doi.org/10.6028/NIST.IR.8202
7. O. Potii, Y. Gorbenko and K. Isirova. Post quantum hash based digital signatures comparative analysis. Features of their implementation and using in public key infrastructure // 2017 4th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T), Kharkov, 2017. Pp. 105-109.
8. Yu. I. Gorbenko, T. V. Melnik, I. D. Gorbenko. Analysis of Potential Post-Quantum Schemes of Hash-Based Digital Signatur // Telecommunications and Radio Engineering. 2018. Volume 77, Issue 7. Pp. 603-626.
9. M. Khazraee, I. Magaki, L. Vega Gutierrez and M. Taylor. ASIC Clouds: Specializing the Datacenter // IEEE Micro.
10. S. Cheng and S. Lin. A Memory-Hard Blockchain Protocol // 2018 IEEE International Conference on Smart Energy Grid Engineering (SEGE), Oshawa, ON, 2018. Pp. 284-287.

11. M. Bedford Taylor. Bitcoin and the age of Bespoke Silicon // 2013 International Conference on Compilers, Architecture and Synthesis for Embedded Systems (CASES), Montreal, QC, 2013, pp. 1-10.

12. X. Zhang, R. WU, M. Wang and L. Wang. A High-Performance Parallel Computation Hardware Architecture in ASIC of SHA-256 Hash // 2019 21st International Conference on Advanced Communication Technology (ICACT), PyeongChang Kwangwoon_Do, Korea (South), 2019, pp. 52-55.

13. I. Magaki, M. Khazraee, L. V. Gutierrez and M. B. Taylor. ASIC Clouds: Specializing the Datacenter. 2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA), Seoul, 2016, pp. 178-190.

14. M. Bedford Taylor. The Evolution of Bitcoin Hardware // Computer, vol. 50, no. 9, pp. 58-66, 2017.

15. A. R. Zamanov, V. A. Erokhin and P. S. Fedotov. ASIC-resistant hash functions // 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), Moscow, 2018, pp. 394-396.

16. N. T. Courtois, P. Emirdag and Z. Wang. On detection of bitcoin mining redirection attacks // 2015 International Conference on Information Systems Security and Privacy (ICISSP), Angers, 2015, pp. 98-105.

17. M. Khazraee, L. V. Gutierrez, I. Magaki and M. B. Taylor. Specializing a Planet's Computation: ASIC Clouds // IEEE Micro, vol. 37, no. 3, pp. 62-69, 2017.

18. Y. Wang, J. Wu, S. Chen, M. C. Chao and C. Yang, "Micro-Architecture Optimization for Low-Power Bitcoin Mining ASICs," 2019 International Symposium on VLSI Design, Automation and Test (VLSI-DAT), Hsinchu, Taiwan, 2019, pp. 1-4.

19. ДСТУ 7564:2014 Інформаційні технології. Криптографічний захист інформації. Функція ґешування. З поправкою. 02.12.2014. Електронний ресурс. Режим доступу: http://online.budstandart.com/ua/catalog/doc-page.html?id_doc=66229

20. ДСТУ ISO/IEC 10118-1:2003. Інформаційні технології. Методи захисту. ґеш-функції. Частина 1. Загальні положення.

21. ДСТУ ISO/IEC 10118-3:2005 «Інформаційні технології. Методи захисту. ґеш-функції. Частина 3: Спеціалізовані ґеш-функції».

22. ДСТУ ISO/IEC 10118-2:2014. Інформаційні технології. Методи захисту. ґеш функції. Частина 2. ґеш-функції, що використовують n-бітовий блоковий алгоритм шифрування. На заміну ДСТУ ISO/IEC 10118-2:2003.

23. ДСТУ ISO/IEC 10118-4:2014. Інформаційні технології. Методи захисту. ґеш-функції Частина 4. ґеш-функції, що використовують модульну арифметику. Вперше.

24. ДСТУ ISO/IEC 9797-1:2015 (ISO/IEC 9797-1:2011, IDT). Інформаційні технології. Методи захисту. Коди автентифікації повідомлень (macs). Частина 1. Механізми, що використовують блоковий шифр. Електронний ресурс. Режим доступу: http://shop.uas.org.ua/ua/informacijni-tehnologii-metodi-zahistu-kodi-avtentifikacii-povidomlen-macs-chastina-1-mehanizmi-scho-vikoristovujut-blokovij-shifr.html

25. ДСТУ ISO/IEC 9797-2:2015 (ISO/IEC 9797-2:2011, IDT. Інформаційні технології. Методи захисту. Коди автентифікації повідомлень (macs). Частина 2. Механізми що використовують спеціалізовану ґеш-функцію. Електронний ресурс. Режим доступу: http://shop.uas.org.ua/ua/informacijni-tehnologii-metodi-zahistu-kodi-avtentifikacii-povidomlen-macs-chastina-2-mehanizmi-scho-vikoristovujut-specializovanu-gesh-funkciju-31543.html

26. ДСТУ ISO/IEC 9797-3:2015 (ISO/IEC 9797-3:2011, IDT. Інформаційні технології. Методи захисту. Коди автентифікації повідомлень (macs). Частина 3. Механізми, що використовують універсальну ґеш-функцію. Електронний ресурс. Режим доступу: http://shop.uas.org.ua/ua/informacijni-tehnologii-metodi-zahistu-kodi-avtentifikacii-povidomlen-macs-chastina-3-mehanizmi-scho-vikoristovujut-universal-nu-gesh-funkciju.html

27. ДСТУ 7624:2014. Інформаційні технології. Криптографічний захист інформації. Алгоритм симетричного блокового перетворення http://uas.org.ua/ua/services/standartizatsiya/109-2/

28. Roman Oliynykov, Ivan Gorbenko, Oleksandr Kazymyrov, Victor Ruzhentsev, Oleksandr Kuznetsov, Yurii Gorbenko, Oleksandr Dyrda, Viktor Dolgov, Andrii Pushkaryov, Ruslan Mordvinov, Dmytro Kaidalov. A New Encryption Standard of Ukraine: The Kalyna Block Cipher. Електронний ресурс. Режим доступу: https://eprint.iacr.org/2015/650.pdf

29. Hash Functions. Created January 04, 2017, Updated May 03, 2019. Електронний ресурс. Режим доступу: https://csrc.nist.gov/projects/hash-functions/sha-3-project

30. Classification of the SHA-3 Candidates. By Ewan Fleischmann, Christian Forler, and Michael Gorski. Version 0.90, April 19, 2009. Електронний ресурс. Режим доступу: https://eprint.iacr.org/2008/511.pdf

31. Алгоритмы майнинга криптовалют – таблица 2019 и краткое описание. Електронний ресурс. Режим доступу: https://mining-cryptocurrency.ru/algoritmy-kriptovalyut/.

Харківський національний
університет імені В.Н. Каразіна;
АТ «Інститут інформаційних технологій»;
Адміністрація Державної служби спеціального зв'язку
та захисту інформації України.

Надійшла до редколегії 02.09.2019

УДК 004.056.5 **DOI:10.30837/rt.2019.3.198.05**

О.О. КУЗНЕЦОВ, д-р техн. наук, Ю.І. ГОРБЕНКО, канд. техн. наук,
В.В. ОНОПРІЄНКО, канд. техн. наук, І.В. СТЕЛЬНИК, Д.В. МЯЛКОВСЬКИЙ

ДОСЛІДЖЕННЯ АЛГОРИТМІВ КРИПТОГРАФІЧНОГО ҐЕШУВАННЯ, ЯКІ ЗАСТОСОВУЮТЬСЯ В СУЧАСНИХ БЛОКЧЕЙН-СИСТЕМАХ

Вступ

Стаття є продовженням попередньої роботи «Алгоритми криптографічного ґешування, які застосовуються в сучасних блокчейн-системах».

В цій роботі досліджуються сучасні алгоритми ґешування, які застосовуються або можуть застосовуватися в різних блокчейн-системах. Зокрема, розглядаються найбільш поширені та застосовувані алгоритми криптографічного ґешування, які стандартизовані на міжнародному та національному рівнях, а також алгоритми, які хоча і не стандартизовані, але застосовуються у більшості сучасних децентралізованих системах, побудованих за технологією блокчейн. Зокрема, досліджено наступні функції криптографічного ґешування:

- сімейство криптографічних алгоритмів ARGON (ґеш-функції ARGON2D та ARGON2I);
- алгоритм ґешування BALLOON;
- сімейство криптографічних алгоритмів BLAKE (ґеш-функції BLAKE224, BLAKE256, BLAKE384, BLAKE512);
- сімейство криптографічних алгоритмів BMW (ґеш-функції BMW224; BMW256; BMW384; BMW512);
- сімейство криптографічних алгоритмів CUBEHASH (ґеш-функції CUBEHASH224; CUBEHASH256; CUBEHASH384; CUBEHASH512);
- алгоритм ґешування DJB-2;
- сімейство криптографічних алгоритмів ECHO (ґеш-функції ECHO224; ECHO256; ECHO384; ECHO512);
- алгоритм ґешування ED2K;
- сімейство криптографічних алгоритмів EDONR (ґеш-функції EDONR256; EDONR512);
- алгоритм ґешування DAGGER-HASHIMOTO та його подальший розвиток і вдосконалення – алгоритм ETHASH;
- сімейство криптографічних алгоритмів FUGUE (ґеш-функції FUGUE224; FUGUE256; FUGUE384; FUGUE512);
- алгоритм криптографічного ґешування GOST34.11-94-256;
- сімейство криптографічних алгоритмів GROESTL (ґеш-функції GROESTL224; GROESTL256; GROESTL384; GROESTL512);
- сімейство криптографічних алгоритмів HAMSI (ґеш-функції HAMSI224; HAMSI256; HAMSI384; HAMSI512);
- алгоритм ґешування Has160;
- сімейство криптографічних алгоритмів J-H (ґеш-функції J-H224; J-H256; J-H384; J-H512);
- сімейство криптографічних алгоритмів KECCAK (ґеш-функції KECCAK224; KECCAK256; KECCAK384; KECCAK512, які стандартизовані як SHA3);
- алгоритм ґешування Kupyna (ґеш-функції Kupyna256 та Kupyna512);
- алгоритм ґешування LOSELOSE;
- сімейство криптографічних алгоритмів LUFFA (ґеш-функції LUFFA224; LUFFA256; LUFFA384; LUFFA512);
- сімейство криптографічних алгоритмів LYRA (ґеш-функції LYRA2RE; LYRA2REV2);
- сімейство криптографічних алгоритмів MD (ґеш-функції MD4 та MD5);
- алгоритм криптографічного ґешування PANAMA256;
- алгоритм ґешування PROGPOW;
- алгоритм криптографічного ґешування RIPEMD160;

- алгоритм криптографічного ґешування SCRYPT;
- алгоритм криптографічного ґешування SHA1;
- сімейство криптографічних алгоритмів SHA2 (ґеш-функції SHA2-256 та SHA2-512);
- сімейство криптографічних алгоритмів SHABAL (ґеш-функції SHABAL256 та SHABAL512);
- сімейство криптографічних алгоритмів SHAVITE (ґеш-функції SHAVITE224; SHAVITE256; SHAVITE384; SHAVITE512);
- сімейство криптографічних алгоритмів SIMD (ґеш-функції SIMD224; SIMD256; SIMD384; SIMD512);
- сімейство криптографічних алгоритмів SKEIN (ґеш-функції SKEIN224; SKEIN256; SKEIN384; SKEIN512);
- алгоритм ґешування SNEFRU256;
- алгоритм ґешування STREEBOG (у варіантах STREEBOG256 та STREEBOG512);
- алгоритм ґешування TIGER;
- алгоритм ґешування WHIRLPOOL;
- алгоритми ґешування із сімейства «X» (алгоритми X11; X12; X13; X14; X15; X17).

Розглянуті алгоритми ґешування побудовано за різними схемами обробки інформаційних повідомлень та із застосовуванням різних математичних перетворень блоків даних для обчислення ґеш-кодів. Різна ідеологія побудови зазначених алгоритмів обумовлена міркуваннями авторів-дослідників та їх суб'єктивними уявленнями про раціональну побудову функції ґешування за критеріями стійкість/складність. Нашу увагу зосереджено, перш за все, на можливості застосування досліджених алгоритмів при побудові децентралізованих блокчейн-систем, зокрема, криптовалют, проектів розподілених технологій, смарт-контрактів, тощо.

Особливості побудови алгоритмів ґешування у сучасних блокчейн-системах

Алгоритм криптографічного ґешування ARGON2. Функція формування ключа Argon2 була розроблена Алексом Бірюковим, Даніелем Діну і Дмитром Ховратовичем з Університету Люксембургу в 2015 р. [1, 2].

Це сучасний та простий алгоритм, спрямований на високу швидкість заповнення пам'яті та ефективне використання декількох обчислювальних блоків. Алгоритм випущений під ліцензією Creative Commons.

У 2013 р. був оголошений конкурс Password Hashing Competition для створення нової функції ґешування паролів. До нового алгоритму висувалися вимоги щодо обсягу використовуваної пам'яті, кількості проходів по блоках пам'яті і по стійкості до криптоаналізу.

У 2015 р. Argon2 був оголошений переможцем конкурсу. З того часу алгоритм зазнав чотири серйозні зміни. Виправлені частина описів алгоритмів генерації деяких блоків і помилки, додані рекомендовані параметри.

Існують дві версії алгоритму:

– *Argon2d* – підходить для захисту цифрової валюти та інформаційних систем, що не піддаються атакам по стороннім каналам;
– *Argon2i* – забезпечує високий захист від trade-off атак, але працює повільніше версії d через кілька проходів по пам'яті.

Argon2 оптимізований під x86 – архітектуру і може бути реалізований на Linux, OS X, Windows.

Argon2d призначений для систем, де зловмисник не отримує регулярного доступу до системної пам'яті або процесору. Наприклад, для backend-серверів і кріптомайнерів. При використанні одного ядра на 2-Ghz CPU і 250 Mb оперативної пам'яті з Argon2d (p = 2) кріптомайнінг займає 0,1 с, а при застосуванні 4 ядер і 4 Gb пам'яті (p = 8) автентифікація на backend сервері проходить за 0,5 с.

Argon2i більше підходить для frontend-серверів і шифрування жорсткого диску. Формування ключа для шифрування на 2-Ghz CPU, використовуючи 2 ядра і 6 Gb оперативної пам'яті, з Argon2i (p = 4) займає 3 с, в той час як автентифікація на frontend-сервері, задіявши 2 ядра і 1 Gb пам'яті з Argon2i (p = 4), займає 0,5 с.

Алгоритм ARGON2 використовується в деяких криптовалютах, наприклад у MMXVI [3].

Алгоритм криптографічного ґешування BALLOON. Найбільш детальний опис алгоритму ґешування BALLOON наведено у [4].

Алгоритм Balloon було розроблено для ґешування паролів, він приймає чотири вхідних параметри: пароль, сіль, часовий параметр і параметр простору. Вихідні дані представляють собою бітові рядки фіксованої довжини (наприклад, 256 або 512 біт).

Параметр простору (Buer Size) вказує, скільки блоків робочого простору з фіксованим розміром буде потрібно ґеш-функції під час його обчислення. На високому рівні функція з жорсткими вимогами до пам'яті повинна бути «легкою» для обчислення і повинна бути «жорсткою» для обчислення з набагато меншим простором.

Параметр часу (кількість раундів) визначає кількість "обходів" обчислень. Чим більше параметр часу, тим довше буде обчислення ґешу. На платформах, обмежених пам'яттю, системний адміністратор може збільшити кількість раундів ґешування, щоб збільшити вартість обчислення функції без збільшення вимоги до пам'яті алгоритму.

Великий обсяг пам’яті, необхідний для швидкої роботи алгоритму, дозволяє зменшити переваги більш швидких обчислень ASIC, порівняно з CPU та GPU.

Алгоритм Balloon використовується у криптовалюті Deft.

Алгоритм криптографічного ґешування BLAKE. Ґеш-функція BLAKE запропонована Jean-Philippe Aumasson, Luca Henzen, Willi Meier, Raphael C.-W. Phan [5]. Зазначені ґеш-функції є "налаштованими" версіями, що подані на фінал конкурсу SHA-3. Оригінальні подані функції були названі BLAKE28, BLAKE-32, BLAKE-48 і BLAKE-64; налаштовані версії – BLAKE-224, BLAKE-256, BLAKE-384 і BLAKE-512.

BLAKE – це сім'я з чотирьох ґеш-функцій: BLAKE-224, BLAKE-256, BLAKE-384 і BLAKE-512. Як і алгоритм SHA-2, ґеш-функція BLAKE має 32-бітну версію (BLAKE-256) і 64-бітну версію (BLAKE-512), з якої інші екземпляри виводяться з використанням різних початкових значень, відмінного заповнення і скороченого виводу.

Реалізація BLAKE вимагає невеликих ресурсів і є швидкою як в програмному, так і в апаратному середовищі [5, 6]. У 180нм ASIC, BLAKE-256 може бути реалізований з близько 13500 gate, і може досягати пропускної здатності більше 4 Гбіт/с; BLAKE-512 може досягати пропускної здатності більше 6 Гбіт/с. На Intel Core 2 Duo, BLAKE-256 може ґешувати близько 15 циклів/байт, коли BLAKE-512 приблизно 10 циклів/байт.

Алгоритм BLAKE використовується в криптовалюті Blakecoin [7]. Крім того, цей алгоритм застосовується як складова в алгоритмах X11, X12, X13, X14, X15 та X17 для майнінгу різних криптовалют та розподілених децентралізованих систем [8].

Алгоритм криптографічного ґешування BMW. Криптографічна ґеш-функція BMW (англ. BMW – Blue Midnight Wish) розроблялася як набагато більш ефективна ґеш-функція, ніж SHA-2, в той же час із таким або кращим рівнем безпеки [9].

Алгоритм BMW працює з повідомленнями, розбиваючи їх на блоки. Блок, в свою чергу, ділиться на слова. Розміри блоків і слів залежать від конкретної реалізації алгоритму.

Алгоритм BMW був представлений у якості кандидата на криптографічний конкурс SHA-3. За результатами досліджень був відхилений конкурсною комісією національного інституту стандартів і технологій США: «Рівень безпеки є нижчим за очікуваний: BMW-256 знижується до 65 біт, BMW-512 – до 128 біт. Витрати пам'яті, необхідні для здійснення цих атак, є несуттєвими» [10].

На сьогодні алгоритм криптографічного ґешування знайшов своє практичне застосування в різних децентралізованих системах типу блокчейн. Зокрема, він застосовується як скла-

дова в алгоритмах X11, X12, X13, X14, X15 та X17 для майнінгу криптовалют [8]: DarkCoin (Dark); DeepOnion (ONION); Cloakcoin (CLOAK); MaruCoin (MARU); Hshare (HSR); Stealthcoin (XST); MarteXcoin (MXT) та інш.

Алгоритм криптографічного ґешування CUBEHASH. Сімейство криптографічних ґеш-функцій CubeHashr / b. CubeHash8 / 1 було запропоновано Деніелом Бернштейном [11] в якості нового стандарту в конкурсі ґеш-функцій SHA-3. Спочатку алгоритм вимагав близько 200 циклів на байт [12]. Після деяких уточнень автор змінив параметри на CubeHash16 / 32, який приблизно в 16 разів швидше, ніж CubeHash8 / 1 і легко наздоганяє SHA-256 і SHA-512 на різних платформах [13, 14].

Ґеш-функція CubeHash пройшла до другого раунду конкурсу SHA-3, але так і не потрапила в п'ятірку фіналістів [10].

Варто відзначити, що CubeHash не поступається в швидкості своїм опонентам. Стійкість цього алгоритму збільшується як при зменшенні b до 1, так і при збільшенні r. Тому CubeHash 8 / 1-512 міцніше (більш безпечніше) ніж CubeHash 1 / 1-512, а CubeHash 1 / 1-512 міцніше ніж CubeHash 1 / 2-512. Найслабший з можливих версій даного алгоритму – це CubeHash 1/128-h. Однак безпека залежить від часу, тобто більш безпечний варіант буде довше обчислювати ґеш-значення.

Ґеш-функція CubeHash застосовується як складова у складі алгоритмів майнінгу криптовалют X11,X12, X13, X14, X15, X17.

Алгоритм ґешування DJB-2. Це дуже проста та швидка ґеш-функція, яка була запропонована Даном Бернштейном багато років тому [15]. Одна з найкращих строкових ґеш-функцій, яка має гарне розподілення та швидкість для безлічі наборів ключів та розмірів таблиць, в порівнянні з іншими алгоритмами, наприклад PJW, LOSELOSE та інш. Існує інша версія алгоритму, яка також схвалена Бернштейном, що використовує операцію xor: hash(i) = hash(i-1)*33^str[i]. Пояснення, чому використані такі константи при обчисленнях, не надаються. Ця функція ґешування не є криптографічною, але вона все ж таки використовується у складі алгоритму майнінгу криптовалют X17 [8].

Алгоритм криптографічного ґешування ECHO. ECHO-256 є кандидатом другого туру конкурсу SHA-3 [10]. Це ґеш-функція на базі алгоритму AES, яка викликала великий інтерес і аналіз. ECHO – ітеративна ґеш-функція [16], функція стиснення ECHO оновлює внутрішній стан, описаний в матриці 16 × 16 елементів GF(2^8), яку також можна розглядати як матрицю 4 × 4 з 16 станів AES. Перетворення на такому великому 2048-бітному стані дуже схожі на AES, основна відмінність полягає в еквівалентному S-Box під назвою BigSubWord, який складається з двох раундів AES. В кінці перестановки операція BigFinal додає поточний стан до вихідного стану (подача вперед) і, в разі ECHO-256, додає чотири стовпці разом, щоб отримати нове значення ланцюжка. В обох раундах AES використовуються два ключа – внутрішній лічильник і сіль відповідно. Вони вводяться в основному для того, щоб зруйнувати існуючі симетрії AES-перестановки без ключа.

Ґеш-функція ECHO застосовується як складова алгоритмів майнінгу криптовалют X11, X12, X13, X14, X15, X17 [8].

Алгоритм криптографічного ґешування ED2K. Ґеш-функція ED2K заснована на алгоритмі MD4, але замість надання одного ґешу всього файлу, вона розбиває файл на частини по 9500 Кбайт і створює остаточний ґеш-код на основі сум MD4 [17].

Ґеш-функція ED2K знайшла застосування у файлообмінних мережах та програмах обміну файлами eDonkey2000 і Overnet [18].

Алгоритм криптографічного ґешування EDONR. Сімейство криптографічних ґеш-функцій EdonR запропоновано у роботі [19]. EdonR є класом ґеш-функцій з змінними довжинами виводу. Він визначається за допомогою квазігрупп і перетворень рядків квазігрупп.

У статті [20] детально описано змінену криптографічну ґеш-функцію Edon-R, яку було представлено в якості кандидата на ґеш-конкурс SHA-3, організований Національним інститутом стандартів і технологій (NIST). Різниця полягає в доданні зворотного зв'язку до вихід-

ної функції стиснення R. Зворотній зв'язок полягає в збереженні виведення функції R з попереднім значенням подвійної труби і значенням поточного блоку повідомлень. Введені зміни роблять недійсними криптоаналітичні зусилля з аналізу квазігрупових операцій, використовуваних в EDON-R, а також його функції R.

Швидкість оптимізованої 32-розрядної версії на певній еталонної платформі з Intel C ++ v 11.0.072 становить 6,71 циклів / байт для n = 224, 256 і 10,74 циклів / байт для n = 384, 512. Швидкість оптимізованої 64-бітної версії на певній еталонної платформі з Intel C ++ v 11.0.072 становить 4,90 циклів / байт для n = 224, 256 і 2,74 циклів [19, 20].

Алгоритм криптографічного ґешування ETHASH. Криптографічна функція Ethash застосовується для перевірки працездатності в заснованих на Ethereum криптовалютах [21]. Вона використовує ґеш-функцію Keccak, стандартизовану як SHA-3. Починаючи з версії 1.0 алгоритм Ethash був розроблений так, щоб бути стійким до ASIC завдяки жорсткості пам'яті (важче реалізувати в спеціальних чіпах ASIC) [21-23]. Він також використовує модифіковану версію більш ранніх ґеш-алгоритмів Dagger і Hashimoto для усунення накладних витрат на обчислення. Ця функція раніше називалася Dagger-Hashimoto [22].

Ethash використовує вихідний набір даних обсягом 1 ГБ, відомий як Ethash DAG, і кеш-пам'ять об'ємом 16 МБ для легких клієнтів. Вони відновлюються кожні 30000 блоків, відомих як епоха. Майнери беруть фрагменти DAG для створення змішаних ґеш, використовуючи дані транзакцій, а також криптографічний одноразовий номер для створення ґешу нижче динамічної цільової складності [21 – 23].

Алгоритм Dagger Hashimoto був попередньою дослідницькою реалізацією і специфікацією алгоритму майнінгу для Ethereum 1.0. Пізніше він був замінений на алгоритм Ethash.

Алгоритм криптографічного ґешування FUGUE. Алгоритм ґешування Fugue був розроблений Shai Halevi , William E. Hall і Charanjit S. Jutla з IBM для конкурсу ґеш-функцій Національного Інституту стандартів і технологій у 2009 р., де пройшов до другого раунду [10]. Однак алгоритм не пройшов до третього раунду конкурсу за недостатньою кількістю криптографічного аналізу та невпевненістю в криптостійкості [24].

Для вхідного повідомлення довжиною від 1 до 2^{64}-1 біт алгоритм генерує 224, 256, 384 або 512-бітове ґеш-значення. Функції для відповідних довжин вихідних даних алгоритм ґешування називається відповідно Fugue-224, Fugue-256, Fugue-384 і Fugue-512 [25]. Автори алгоритму також описали параметризовану версію алгоритму Fugue [25]. Слабозахищена версія Fugue-256, що працює в два рази швидше стандартної версії, також описується через параметризовану версію.

Fugue був спроектований таким чином, щоб зменшити вразливість перед атаками диференціального аналізу [25]. Також, за запевненнями авторів алгоритму, ґеш-функція конкурентоспроможна за ефективністю з SHA ґеш-функціями в програмному і апаратному вигляді, досягаючи продуктивності до 36,2 циклів в байт (CPB) на шостому сімействі процесорів Intel Xeon 5150, і до 25 циклів в байт (CPB) на процесорі Intel Core 2 T7700. На 45 нанометровому чіпі Intel Core 2 T9400 Fugue-256 досягає всього 16 циклів в байт (CPB), використовуючи інструкції SSE 4.1. На процесорах з архітектурою Westmere (32нм), типу Intel Core i5, ґеш для Fugue-256 розраховується зі швидкістю 14 циклів в байт (CPB).

Алгоритм Fugue застосовується як складова в алгоритмах X13, X14, X15 та X17 для майнінгу різних криптовалют [8].

Алгоритм криптографічного ґешування GOST34.11-94. Для обчислення криптографічної функції ґешування в Росії у 1994 р. було введено стандарт ГОСТ Р 34.11-94 [26] (який на цей час вже скасовано) [27]. Згодом цей стандарт було перевидано як міждержавний стандарт СНД ГОСТ 34.311-95 [28]. До 2013 р ЦБ РФ вимагав використовувати ГОСТ Р 34.11-94 для формування та перевірки електронного підпису. З 1 січня 2013 р. був замінений на ГОСТ Р 34.11-2012 «Стрибог» [27].

Розробником стандарту ГОСТ 34.311 є Головне управління безпеки зв'язку (ГУБЗ) Федерального агентства урядового зв'язку та інформації (ФАУЗІ) і Всеросійський науково-

дослідний інститут стандартизації. Цей стандарт є обов'язковим для застосування в якості алгоритму ґешування в державних організаціях РФ і ряді комерційних організацій.

Стандарт визначає алгоритм і процедуру обчислення ґеш-функції для послідовності символів. При обробці блоків використовуються перетворення за алгоритмом криптографічного перетворення ГОСТ 28147-89 [29]. Обробляється блок довжиною 256 біт, вихідне значення теж має довжину 256 біт. Алгоритм визначає контрольну суму, розраховану по всіх блоках вихідного повідомлення, яка є частиною фінального обчислення ґешу, що дещо ускладнює атаки на пошук колізій. Застосовуються також заходи боротьби проти пошуку колізій, що засновані на неповноті останнього блоку. Обробка блоків відбувається за алгоритмом шифрування ГОСТ 28147-89, який містить перетворення на S-блоках, що істотно ускладнює застосування методу диференціального криптоаналізу до пошуку колізій.

Алгоритм криптографічного ґешування GROESTL. Ітеративна криптографічний ґеш-функція GROESTL (правильне написання «Grøstl») [30, 31] є однією з п'яти фіналістів конкурсу SHA-3, організованого NIST [10].

Стискаюча функція Grøstl складається з двох фіксованих перестановок P і Q, структура яких запозичена у шифру AES. Зокрема, використовується такий же S-блок. Результат роботи ґеш-функції може мати довжину від 8 до 512 біт з кроком 8 біт. Варіант, який повертає n біт, називається Grøstl-n.

Алгоритм Grøstl спеціально розроблений для участі в конкурсі криптографічних функцій SHA-3 командою криптографов з Данського технічного університету [10]. Спочатку функція називалася Grøstl-0. Однак для участі в фіналі були збільшені структурні відмінності між перестановками. Були змінені значення ShiftBytes в перестановці Q. Також змінам піддалися раундові константи в P і Q. Оновлена ґеш-функція отримала назву Grøstl. Однак, показавши хорошу криптостійкість, по ряду показників вона поступалася іншим учасникам фінального раунду і не змогла стати переможцем.

Назва алгоритму походить від назви блюда Грёстль австрійської кухні. За рецептом воно дуже близько до страви, яке в США називають «Hash». Буква «ö» в назві функції була замінена на букву «ø» з данського алфавіту, яка має таку ж вимову.

За своєю структурою алгоритм Grøstl є байт-орієнтованою SP-мережею. За своєю будовою цей алгоритм значно відрізняється від алгоритмів сімейства SHA (MD-подібних конструкцій). Багато компонентів ґеш-функції Grøstl також запозичені з шифру AES.

Ця функція ґешування є досить ефективним та безпечним криптопримітивом. Знайшла застосування в деяких блокчейн-системах, зокрема у проекті криптовалюти Verge [32].

Алгоритм криптографічного ґешування HAMSI. Криптографічна ґеш-функція Hamsi була запропонована при проведенні конкурсу SHA-3 [10]. В її основу покладено алгоритми Grindahl [33] і Serpent [34]. Ця ґеш-функція не запатентована і є громадським надбанням. Існують два різновиди алгоритму: Hamsi-256 і Hamsi-512 [10].

В основі алгоритму лежать функція розкладання і циклічна трансформація. Циклічна трансформація працює з чотирма рядками матриці станів. Число стовпців цієї матриці дорівнює 4 для Hamsi-256, 8 для Hamsi-512. Елементами матриці є слова розміром 32 біта.

Ґеш-функція Hamsi була одним з учасників у відкритому конкурсі SHA-3 Національного інституту стандартів і технологій. Однак Hamsi не потрапила в число 5 кандидатів останнього туру, оголошених 10 грудня 2010 [10].

Алгоритм Hamsi застосовується як складова в алгоритмах X12, X13, X14, X15 та X17 для майнінгу різних криптовалют [8].

Алгоритм криптографічного ґешування Has160. HAS-160 – це ґеш-функція, яка призначена для використання з корейським стандартом цифрового підпису (який пов'язаний з DSA, але має деякі відмінності). Він дуже схожий на SHA-1, але має деякі зміни, які, ймовірно, збільшують стійкість алгоритму. Він описаний в корейському стандарті під назвою TTAS.KO-12.0011 / R1 [35], але немає опису англійською мовою.

З огляду на дуже широке застосування SHA-1 (а також і SHA-256 і SHA-512), цілком ймовірно, що алгоритму HAS-160 ніколи не буде приділено стільки ж уваги при аналізі або розгортанні, скільки він заслуговує. HAS-160, однак, значно швидше, ніж SHA-1, і (принаймні, на перший погляд) здається приблизно таким же безпечним. Єдина очевидна можлива атака, яку можливо застосувати до HAS-160, це диференціальна атака, яка зламала SHA-0. HAS-160 в основному можна розглядати як продукт переходу до дизайну, який почався з SHA-1 і закінчився як HAS-V. Він використовується в деяких корейських продуктах, а також в KeyTools Crypto v5.0.1 компанії Baltimore Technologies. Є кілька інших ґешів від тих же розробників, які схожі, але не ідентичні, включаючи PMD-V, PMD-128, PMD-160, PMD-192, PMD-224 і PMD-256 (всі з яких є різними алгоритмами). Ні у одного з них немає специфікацій англійською мовою, є дуже мало інформації про них. Алгоритм в значній мірі описується як різниця між ним і SHA-1.

У [36] наведено реалізацію HAS-160 в C ++ як частина крипто-бібліотеки Botan.

У описі алгоритму з [36] поняття «крок» відноситься до окремого застосування підфункції. Наприклад, SHA-1 (а також HAS-160) використовує 4 раунди по 20 кроків кожен, всього 80 кроків.

Алгоритм криптографічного ґешування J-H. Алгоритм JH – це сімейство з чотирьох криптографічних ґеш-функцій: JH-224, JH-256, JH-384 і JH-512 [37]. Кожна з цих ґеш-функцій відрізняється тільки значенням одного внутрішнього параметра – довжини (в бітах) вихідного значення.

Ґеш-функція JH входить в п'ятірку фіналістів другого туру SHA-3 [10]. В процесі цього конкурсу вона була покращена.

Алгоритм JH розроблений і представлений на конкурс SHA-3 в 2008 р. сінгапурським криптографом Wu Hongjun. Алгоритм базується на конструкції «криптографічна губка». Стиснення виконується бієктивною функцією (блочний шифр з постійним ключем). Прототипом для алгоритму шифрування став AES. AES побудований на основі SP-мережі, вхідні дані представляють собою двовимірний масив. У JH-алгоритмі використовується його модифікація: AES узагальнюється до використання багатовимірних масивів, що дозволило побудувати шифр з великою довжиною блоку на основі більш маленьких компонентів.

У 2011 р. запропонована модифікація алгоритму: кількість раундів блочного шифру збільшилася до 42 для підвищення ефективності реалізації на апаратних платформах і поліпшення характеристик безпеки. Незважаючи на це, в цьому ж році представлена атака на перестановку, яка поширюється на всі 42 раунди, а також атака на функцію стиснення, що дозволила отримати псевдоколізії аж до 37 раунду [38]. У 2013 р. з'явилися інші відомості про ряд атак на ґеш-функції, серед яких опинилася і JH.

Алгоритм JH застосовується як складова в алгоритмах X11, X12, X13, X14, X15 та X17 для майнінгу різних криптовалют [8].

Алгоритм криптографічного ґешування KECCAK. Алгоритм криптографічного ґешування Keccak розроблений групою авторів на чолі з Йоаном Дайменом, співавтором Rijndael, автором шифрів MMB, SHARK, Noekeon, SQUARE і BaseKing [39]. 2 жовтня 2012 р. Keccak став переможцем конкурсу криптографічних алгоритмів SHA-3, що проводився Національним інститутом стандартів і технологій США [10]. 5 серпня 2015 р. алгоритм затверджений і опублікований в якості стандарту FIPS 202 [40].

Стандарт FIPS 202 визначає сімейство криптографічних ґеш-функцій SHA-3:

- SHA3-224 – довжина ґеш-значення 224 бітів;
- SHA3-256 – довжина ґеш-значення 256 бітів;
- SHA3-384 – довжина ґеш-значення 384 бітів;
- SHA3-512 – довжина ґеш-значення 512 бітів,

та дві функції розширення (XOF, Extendable Output Functions) SHAKE128 і SHAKE256, для чого повідомлення необхідно доповнювати «суфіксом» з 2 або 4 біт, в залежності від типу функції:

- функція розширення SHAKE 128 (Secure Hash Algorithm Keccak), результат може бути довільної довжини, забезпечує рівень безпеки 128 бітів;
- функція розширення SHAKE256 (Secure Hash Algorithm Keccak), результат може бути довільної довжини, забезпечує рівень безпеки 256.

У програмній реалізації автори заявляють про 12,5 циклів на байт при реалізації на ПК з процесором Intel Core 2. Однак в апаратних реалізаціях Keccak виявився набагато швидше, ніж всі інші фіналісти [41].

На сьогодні алгоритм SHA-3 є однією із найпоширеніших криптографічних ґеш-функцій, яка застосовується у багатьох криптовалютах, наприклад у Nexus (NXS), SmartCash (SMART), X-Cash (XCASH), MaxCoin (MAX), SecureCoin (SRC), Bitcoin File (BIFI), CreativeCoin, Slothcoin (SLOTH), 365Coin (365), Galleon (GLN), Helix Coin (HXC), CryptoMeth (METH), BitcointalkCoin (TALK) та інш. [42, 43].

Алгоритм криптографічного ґешування Kupyna. Національний стандарт України ДСТУ 7564:2014 «Інформаційні технології. Криптографічний захист інформації. Функція ґешування» (набрав чинності з 1 квітня 2015 р.) описує ітеративну криптографічну ґеш-функцію «Купина». Цей стандарт прийнятий наказом Мінекономрозвитку від 2 грудня 2014 р. №1431 [44]. Текст стандарту є у вільному доступі [45].

Стандарт розроблено задля поступової заміни міждержавного стандарту ГОСТ 34.311-95 [28] та згідно чинних змін до наказу Держспецзв'язку від 20 серпня 2012 р. №1236/5/453 [46] після 1 січня 2022 р. разом з алгоритмом криптографічного перетворення ДСТУ 7624:2014 є обов'язковим для використання при накладанні та перевірці електронного цифрового підпису за ДСТУ 4145-2002 замість функції ґешування за ГОСТ 34.311-95.

Функція стиснення Купини складається з двох фіксованих перестановок, структура яких запозичена у шифра Калина (докладна специфікація англійською мовою наведена у статті [45]). Зокрема, використовуються чотири таких самих S-блоків. Результат роботи ґеш-функції може мати довжину від 8 до 512 біт. Варіант, який повертає n біт, позначається як Купина-n. Основними режимами роботи функції ґешування, що рекомендуються до застосування, є «Купина-256», «Купина-384» і «Купина-512».

За своєю структурою функція ґешування Купина є подібною алгоритму Groestl [30, 31]. Очікується застосування цієї ґеш-функції для побудови національних розподілених блокчейн систем.

Алгоритм ґешування LOSELOSE. Ґеш-функція запропонована в книзі Брайана Кернігана та Денніса Ритчи «Язык программирования Си» (англ. The C Programming Language. - 1978 також відома як K&R), причому останній з авторів – один з безпосередніх авторів та розробників мови програмування Сі [47].

За оптимістичними оцінками, це: «… *не самий кращій алгоритм, але він дуже простий*. *Алгоритм міг би бути набагато кращим, якщо не був би таким простим*…» [47]. Багато програмістів на Сі використовують цю функцію, фактично не перевіряючи її, наприклад, шляхом *сортування та пошуку* Кнута. В деяких розробках ця проста конструкція використовується як одна з компонент в схемах змішування та кодування.

Функція ґешування LOSELOSE не є криптографічною, але вона застосовується у складі алгоритму майнінгу криптовалют X17 [8].

Алгоритм криптографічного ґешування LUFFA. Криптографічний алгоритм (сімейство алгоритмів) ґешування Luffa [48] змінної розрядності, розроблений Даї Ватанабе (англ. Dai Watanabe), Хісайоші Сато (англ. Hisayoshi Sato) з Hitachi Yokohama Research Laboratory і Крістофом Де Канньере (нід. Christophe De Cannière) з дослідницької групи COSIC Льовенського католицького університету для участі в конкурсі Національного інституту стандартів і технологій США [10]. Luffa є варіантом функції губки, запропонованої Гвідо Бертоні (англ. Guido Bertoni) і співавторами, криптостійкість якої заснована тільки на випадковості перестановки. На відміну від оригінальної функції губки, Luffa використовує множину паралельних перестановок і функції інжекції повідомлень.

В ході другого туру конкурсу SHA-3 алгоритми Luffa-224 і Luffa-256 в первинному варіанті показали низьку криптостійкість, для успішної атаки потрібно лише 2^{216} повідомлень. Після чого, алгоритм був модифікований Даї Ватанабе і отримав назву Luffa v.2. Зміни Luffa v.2 є такими [10]:

- доданий порожній раунд функції завершення для всіх розмірів ґешу;
- змінений S-блок;
- збільшено кількість повторень крокової функції з 7 до 8.

Барт Пренель (Bart Preneel) представив успішну атаку [49] з пошуку колізій для 4 раундів крокової функції Luffa за 2^{90} операцій ґеширования і 2^{224} для 5-раундової схеми, показавши тим самим границю стійкості дизайну до диференціальних атак пошуку колізій.

У 2010 р. Томас Олівіера і Джуліо Лопез провели успішні дослідження [50] можливості збільшення продуктивності оригінальної реалізації Luffa. Оптимізована реалізація алгоритму має 20 % збільшення продуктивності обчислення ґешу Luffa-512 при виконанні в 1 потоці, для Luffa-256/384 приріст продуктивності однопоточної реалізації в різних тестах становить не більше 5 %.

Ґеш-функція Luffa застосовується як складова алгоритмів майнінгу криптовалют X11, X12, X13, X14, X15, X17.

Алгоритм криптографічного ґешування LYRA2RE. Найбільш докладний опис цього алгоритму ґешування наведено у [51].

Функція Lyra2RE – це алгоритм ґешування для майнінгу криптовалют, розроблений командою Vertcoin для заміни алгоритму Scrypt-N. Мета полягала в тому, щоб скоротити споживання електроенергії на майнінг токенов VTC і підтримувати монету недоступною для ASIC-майнінгу. Зміну алгоритму було анонсовано в липні 2014 р. і успішно здійснено розробниками [51, 52].

Lyra2RE – алгоритм зі змінними параметрами, які будуть корисні для зриву майбутніх загроз від ASIC пристроїв (літери RE в назві є абревіатурою «Reduced Efficiency» або «знижена працездатність»). За результатами тестів було визначено, що Lyra2RE споживає на ~ 30 % менше енергії в порівнянні, наприклад, з алгоритмом Scrypt-N і на ~ 17 % менше, ніж алгоритм X11. Lyra2RE дозволяє розробникам Vertcoin самостійно змінювати використання пам'яті і вартість часу [51, 52].

Алгоритм Lyra2RE був змінений розробниками Vertcoin на користь GPU-майнерів через одиничний ботнет на базі CPU, який контролював велику частину ґеш-потужності мережі. В результаті 10 серпня 2015 р. (блок 347000) Vertcoin був модифікований від Lyra2RE до Lyra2REv2 [51, 52].

Lyra2REv2 – це алгоритм ґешування для консенсусу Proof-of-work, створений командою криптовалюти Vertcoin (VTC) [51, 52]. Lyra2REv2 складається з ланцюжка різних ґеш-функцій – Blake, Keccak, Cubehash, Lyra2, Skein та BMW (Blue Midnight Wish). Lyra2REv2, заснований на першій версії Lyra2RE. Цей ланцюговий алгоритм безпечний, надійний і призначений для протистояння ASIC-майнінгу. Різниця між Lyra2RE і Lyra2REv2 полягає в більш пізній версії. Були введені 2 раунди Cubehash для зниження ефективності процесора і для того, щоб пом'якшити ефект від майнінгу ботнетів. Lyra2REv2 також споживає менше енергії, ніж попередня версія. Оскільки алгоритм залежить від пам'яті і через його ланцюгову структуру проектування ASIC буде досить складним. Після запуску Lyra2REv2 в Vertcoin (VTC) багато інших криптовалют стали також використовувати даний алгоритм доказу роботи. Деякі кріптовалютні проекти називають алгоритм ґешування Lyra2REv2, інші як Lyra2v2, однак це один і той же алгоритм. За рахунок того, що ґешування має принципову послідовність, так як кожен новий етап використовує результати попереднього, проводити паралельні потоки обчислень стає неможливим. Такий метод побудови є дуже надійним захистом алгоритму від розробки ASIC. І поки це вдається: на сьогоднішній день інтегральної схеми спеціального призначення для алгоритму Lyra2REv2 не поширені.

Паралелізація обчислень неможлива, а значить, для майнінгу потрібно устаткування, яке краще за інших вміє проводити багатомільйонні однотипні операції швидко і результативно, якими і є відеокарти [51, 52].

До популярних криптовалют, які працюють на основі алгоритму Lyra2v2 відносять наступні [51, 52]: Vertcoin (VTC); MonaCoin (MONA); Rupee (RUP); Straks (STAK); Verge (XVG); Shield (XSH); Galactrum (ORE).

Алгоритм криптографічного ґешування MD4. Криптографічна ґеш-функція MD4 (Message Digest 4) була розроблена професором Массачусетського університету Рональдом Ривестом в 1990 р., і вперше описана в RFC 1186 [53]. Для довільного вхідного повідомлення функція генерує 128-розрядне ґеш-значення. Цей алгоритм використовується в протоколі аутентифікації MS-CHAP, розробленому корпорацією Майкрософт для виконання процедур перевірки достовірності віддалених робочих станцій Windows. Є попередником MD5. Рівень безпеки, які закладаються в MD4, були розраховані на створення досить стійких гібридних систем електронного цифрового підпису, заснованих на MD4 і криптосистем з відкритим ключем. Рональд Ривест вважав, що алгоритм ґешування MD4 можна використовувати і для систем, які потребують сильної криптостойкости. Але в той же час він відзначав, що MD4 створювався насамперед як дуже швидкий алгоритм ґешування, тому він може бути поганий в змісті криптостійкості. Як показали подальші дослідження, він мав рацію, і для додатків, де важлива насамперед криптостійкість, став використовуватися більш надійний алгоритм MD5.

Алгоритм криптографічного ґешування MD5. MD5 (англ. Message Digest 5) це 128-бітний алгоритм ґешування, розроблений професором Рональдом Л. Ривестом в 1991 р. для заміни більш ранньої ґеш-функції MD4, був визначений в 1992 р. як RFC 1321 [54].

Широко застосовувався для перевірки цілісності інформації та зберігання ґешів-паролів. На сьогоднішній день безпека цього алгоритму також є недостатньою [55]. Навіть у 2019 р. MD5 продовжує широко використовуватися, незважаючи на добре документовані недоліки і відсутність підтримки з боку експертів з безпеки [56].

Алгоритм криптографічного ґешування PANAMA256. PANAMA – це криптографічний модуль, який можна використовувати як в якості криптографічної ґеш-функції, так і потокового шифру [57]. Він розроблений, щоб бути дуже ефективним в реалізації програмного забезпечення на 32-бітних архітектурах. Його основні операції виконуються над 32-бітними словами.

Алгоритм «Panama» заснований на машині з кінцевими станами, що складається з двох великих блоків: 544 біта станів і 8192-бітового буферу, який працює за принципом регістра зсуву зі зворотним зв'язком. Зворотній зв'язок забезпечує те, що вхідні біти після входу проходять через декілька ітерацій, що, в свою чергу, забезпечує побітову дифузію. Треба сказати, що подібний буфер застосовується в функції стиснення SHA. Об'єктом роботи «Panama» є 32-бітове слово і стан складається з 17 таких слів, в той час як буфер має 32 осередки, в кожній з яких лежить по 8 таких слів.

Ґеш-функція PANAMA піддавалася успішним атакам двічі. Вже в 2001 р. було показано, що дана ґеш-функція не є крипостійкою, так як були знайдені колізії за 2^{82} операцій. Більш того, можна знайти колізії вже за 2^{6} операцій, а для задоволення параметрами надійності необхідно, щоб колізії перебували хоча б за 2^{128} операцій [58].

Алгоритм криптографічного ґешування PROGPOW. ProgPoW – це алгоритм перевірки роботи, призначений для усунення недоліку ефективності, доступного спеціалізованим ASIC [59]. Він використовує практично всі частини GPU, і поставляється заздалегідь налаштованим для найпоширенішого обладнання, що використовується в мережі Ethereum.

З моменту випуску першого ASIC для майнінга біткоїну було створено багато нових алгоритмів Proof of Work, які були створені з метою бути "стійким до ASIC". Мета «стійкість до ASIC» полягає в тому, щоб протистояти централізації потужностуй майнінгу PoW таким чином, щоб цими монетами не могли так легко маніпулювати кілька гравців.

Дизайнерська мета ProgPoW полягає в тому, щоб вимоги алгоритму відповідали тому, що доступно на графічних процесорах: якщо алгоритм повинен бути реалізований на спеціальній ASIC, то має бути мало можливостей для підвищення ефективності порівняно з раніше застосованими GPU.

Новий алгоритм ProgPoW (Programmatic Proof-of-Work) повинен стати спадкоємцем алгоритму Ethash з посиленим захистом від використання ASIC-Майнер. Одна з найбільш популярних кріптовалют – Ethereum – планує перейти з алгоритму Ethash на ProgPoW, а це може вже незабаром призвести до появи нових монет на даному алгоритмі. Першою криптовалютою на алгоритмі ProgPoW стала Bitcoin Interest (BCI) [60].

Алгоритм криптографічного ґешування EQUIHASH. Цей алгоритм найбільш повно описаний у роботі [61].

Функція ґешування Equihash – це вимогливий до пам'яті алгоритм доказу роботи, запропонований Міжгалузевим центром безпеки, надійності та довіри Люксембургу (SnT) у 2016 р. на симпозіумі Network and Distributed System Security. Алгоритм базується на узагальненні проблеми «Дня народження», яка полягає у знаходженні коллізій ґешів. Він має серйозні часово-просторові компроміси, але уразливий до непередбачених паралельних оптимізацій. Алгоритм розроблений таким чином, щоб паралельні реалізації обмежувалися шириною пропускної здатності пам'яті, намагаючись збільшити витрати на розробку спеціальних реалізацій ASIC. Опір ASIC в Equihash базується на припущенні, що комерційний апарат вже має досить високу пропускну здатність пам'яті, тому вдосконалення, зроблене за допомогою спеціального обладнання, може не коштувати вартості розробки.

Алгоритм Equihash є дуже поширеним, його використовують такі криптовалюти: Zcash (ZEC); ZenCash (ZEN); ZClassic (ZCL); Bitcoin Gold BTG); Bitcoin Private (BTCP); MinexCoin (MNX); BitcoinZ (BTCZ); Komodo (KMD); Hush (HUSH).

Алгоритм криптографічного ґешування RANDOMX. RandomX – це алгоритм перевірки працездатності (PoW), оптимізований для процесорів загального призначення. RandomX використовує випадкове виконання коду (звідси і назва) разом з кількома методами, які займають багато пам'яті, щоб мінімізувати перевагу ефективності спеціалізованого обладнання [62].

RandomX поводиться як ключова ґеш-функція: він приймає ключ і довільний вхід і видає 256-бітний результат. У структурі RandomX використовується віртуальна машина, яка виконує програми в спеціальному наборі команд, який складається з комбінації математики цілих чисел, математики з плаваючою комою і гілок. Ці програми можуть бути перетворені в машинний код процесора «на льоту». Прикладом програми RandomX, перекладеної на збірку x86-64, є program.asm [62].

RandomX є доказом роботи (PoW) алгоритму, який був розроблений, щоб закрити розрив між процесорами загального призначення та спеціалізованим обладнанням. Ядром алгоритму є імітація віртуального процесора.

RandomX був спочатку розроблений як алгоритм PoW для проекту Monero, але на сьогодні застосовується і в інших розподілених системах.

Алгоритм криптографічного ґешування RIPEMD160. Криптографічна ґеш-функція RIPEMD-160 (від англ. RACE Integrity Primitives Evaluation Message Digest) була розроблена в Католицькому університеті Лувена Хансом Доббертіном (Hans Dobbertin), Антоном Босселарсом (Antoon Bosselaers) і Бартом Пренелом (Bart Preneel) [63]. Для довільного вхідного повідомлення функція генерує 160-розрядне ґеш-значення.

RIPEMD-160 є покращеною версією RIPEMD, яка, в свою чергу, використовувала принципи MD4 і за продуктивністю порівняння з більш популярною SHA-1.

Також існують 128, 256 і 320-бітові версії цього алгоритму, які, відповідно, називаються RIPEMD-128, RIPEMD-256 і RIPEMD-320. 128-бітна версія являє собою лише заміну оригінальної RIPEMD, яка також була 128-бітною і в якій були знайдені вразливості [64].

256 і 320-бітові версії відрізняються подвоєною довжиною дайджесту, що зменшує ймовірність колізій, але при цьому функції не є більш крипостійкі.

RIPEMD-160 була розроблена в відкритій академічній спільноті, на відміну від SHA-1 і SHA-2, які були створені NSA. З іншого боку, RIPEMD-160 на практиці використовується не так часто, ніж SHA-1. Використання RIPEMD-160 не обмежене будь-якими патентами.

RIPEMD-160 – це ітеративна ґеш-функція, що працює над 32-бітними словами. Раундова функція приймає на вхід 5-слівну зв'язувальну змінну та 16-слівний блок повідомлення та переводить це в нову зв'язувальну змінну. Усі операції визначено над 32-бітними словами. Заповнення таке саме, як у MD4 [53, 54].

Бітовий розмір результату ґешування та зв'язувальної змінної для RIPEMD-160 збільшено до 160 бітів (п'ять 32-бітних слів), кількість раундів збільшено з трьох до п'яти, між двома рядками зроблено більше відмінностей (змінено не тільки сталі величини, але також булеві функції та порядок слів повідомлення).

RIPEMD-128 і RIPEMD-160 уже мають два паралельні рядки, тому розширення з подвійною довжиною (до 256 та 320 бітів відповідно) можливо побудувати без потреби в двох паралельних екземплярах: достатньо відкинути поєднання двох рядків наприкінці кожного застосування функції стиснення.

Алгоритм RIPEMD є однією із найпоширеніших функцій ґешування, який застосовується у багатьох сучасних криптовалютах [65].

Алгоритм криптографічного ґешування SCRYPT. Найбільш подробно цей алгоритм описаний у роботі [66].

Scrypt (читається ес-крипт [66]) – адаптивна криптографічна функція формування ключа на основі пароля, створена офіцером безпеки FreeBSD Коліном Персивалем для системи зберігання резервних копій Tarsnap. Функція створена таким чином, щоб ускладнити атаку перебором за допомогою ПЛІС. Для її обчислення потрібен значний обсяг пам'яті з випадковим доступом. 17 вересня 2012 р. алгоритм scrypt був опублікований IETF у вигляді Internet Draft для подальшого внесення в RFC. Використовується, наприклад, в якості доказу виконаної роботи в криптовалюті Litecoin.

Scrypt використовується в багатьох криптовалютах як алгоритм перевірки роботи. Алгоритм вперше реалізований для Tenebrix (випущений у вересні 2011 р.) і став основою для Litecoin і Dogecoin. Також алгоритм scrypt використовують ProsperCoin, CashCoin, MonaCoin, Mooncoin та багато інших.

Scrypt важко реалізувати на спеціалізованому апаратному пристрої (ASIC), тому для майнінга зазвичай використовують CPU та GPU з програмною реалізацією алгоритму.

Алгоритм криптографічного ґешування SHA1. Secure Hash Algorithm 1 (SHA1) або алгоритм безпечного ґешування 1 – алгоритм криптографічного ґешування, який опубліковано в RFC 3174 [67]. Довжина вхідних повідомлень дозволена максимум 2^{64} -1 біт (що приблизно дорівнює 2 екзабайта). Алгоритм генерує 160-бітне (20 байт) ґеш-значення, що називається також дайджестом повідомлень, і яке зазвичай відображається як шестнадцятирічне число, довжиною в 40 цифр.

Цей алгоритм використовується у багатьох криптографічних додатках і протоколах. Також його рекомендовано в основному для державних установ в США. Принципи, закладені в основу SHA-1, аналогічні тим, які використовувалися Рональдом Ривестом при проектуванні MD4 [53].

Брюс Шнайер доходить такого висновку: «SHA-1 – це MD4 з додаванням розширювального перетворення, додаткового етапу і поліпшеним лавинним ефектом» [68].

Алгоритм криптографічного ґешування SHA2. Ґеш-функція SHA-2 розроблена Агентством національної безпеки (АНБ) США і опублікована Національним інститутом стандартів і технологій у федеральному стандарті обробки інформації FIPS PUB 180-2 в серпні 2002 р. [69]. В цей стандарт також увійшла ґеш-функція SHA-1, розроблена в 1995 р. У лютому 2004 р. в FIPS PUB 180-2 була додана SHA-224 [70]. У жовтні 2008 р. вийшла нова

редакція стандарту – FIPS PUB 180-3 [71]. У березні 2012 р. вийшла остання на даний момент редакція FIPS PUB 180-4, в якій були додані функції SHA-512/256 і SHA-512/224, засновані на SHA-512 (оскільки на 64-бітних архітектур SHA-512 працює швидше, ніж SHA-256) [72].

Таким чином, SHA-2 (англ. Secure Hash Algorithm Version 2 – безпечний алгоритм ґешування, версія 2) є сімейством криптографічних алгоритмів – односпрямованих ґеш-функцій, що включає в себе алгоритми SHA-224, SHA-256, SHA-384, SHA-512, SHA -512/256 і SHA-512/224.

Ґеш-функції сімейства SHA-2 побудовані на основі структури Меркле – Дамгарда. Повідомлення після доповнення розбивається на блоки, кожен блок містить на 8 слів. Алгоритм пропускає кожен блок повідомлення через цикл з 64-ма або 80-ма ітераціями (раундами). На кожній ітерації 2 слова з восьми перетворюються, функцію перетворення задають інші слова. Результати обробки кожного блоку складаються, сума є значенням ґеш-функції.

Алгоритми ґешування SHA-224, SHA-256, SHA-384 і SHA-512 урядом США допускаються до використання в деяких урядових програмах, включаючи використання в рамках інших криптографічних алгоритмів і протоколів, для захисту інформації, що не мають грифу секретності. Стандарт також допускає використання SHA-2 приватними і комерційними організаціями. Отже алгоритми сімейства SHA-2 є чи не найпоширенішими функціями ґешування, які застосовуються у тому числі в розподілених децентралізованих системах типу блокчейн та величезної кількості криптовалют [73].

Алгоритм криптографічного ґешування SHABAL. Алгоритм криптографічного ґешування SHABAL представлений у роботах [74, 75]. Був одним з учасників конкурсу SHA-3, який проводився Національним інститутом стандартів і технологій (NIST) в 2012 р., не зміг вийти у другий раунд. Він був оцінений як найшвидша функція в конкурсі. Цей алгоритм може бути швидким, але йому не вистачає безпеки.

Був представлений на конкурс дослідницьким проектом «Сапфір», спонсором якого є Французьке дослідницьке агентство (ANR), а головною організацією – France Telecom.

Автори алгоритму: Еммануель Брессон, Анна Кантеоут, Беноїт Шевальє-Мамес, Крістоф Клавьере, Томас Фухр, Аліна Гоуджет, Томас Ікарт, Жен-Франсуа Місарскі, Марія Ная-Пласенкія, Паскаль Пайлер, Томас Порніні, Жан-Рене Рейнхард, Селіна Тьюлльет, Маріон Відеау.

Окремі варіації алгоритму SHABAL називаються SHABAL-512, SHABAL-384, SHABAL-256, SHABAL-224, SHABAL-192 в залежності від довжини одержуваного ґеша, відповідно рівного 512, 384, 256, 224, 192 біт.

Після того як на вхід алгоритму приходить бітова послідовність, вона розбивається на блоки по 512 біт незалежно від використовуваної варіації SHABAL (SHABAL-512, SHABAL-384 і т.д.). Розмір блоку кратний 32. До останнього блоку, якщо його бітова довжина не дорівнює 512 бітам, приписується одна бітова одиниця і необхідне число нулів для досягнення заданого розміру блоку.

Алгоритм SHABAL є чимось середнім між схемою Меркле – Дамгаарда і функцією губки (sponge function). Для стійкості цих схем необхідна криптостойкость перетворення P. Розробники SHABAL прагнули до цієї мети. На жаль P виявилося нестійким, але криптоаналитики прийшли до висновку, що безпека SHABAL від цього не постраждала.

Ґеш-функція застосовується як складова алгоритмів майнінгу криптовалют X14, X15, X17.

Алгоритм криптографічного ґешування SHAVITE. Криптографічна ґеш-функція SHAvite розроблена ізраїльськими криптографами Елі Біхамом (англ. Eli Biham) і Ором Дункельманом (англ. Orr Dunkelman) [76]. Одна з чотирнадцяти учасників другого раунду конкурсу SHA-3, організованого NIST [10].

SHAvite-3 заснована на поєднанні компонентів AES с фреймворком HAIFA. Дана ґеш-функція використовує такі криптографічні примітиви, як мережа Фейстеля і конструкція Девіса-Мейєра.

Сімейство ґеш-функцій SHAvite-3 включає в себе два алгоритму – SHAvite-256 і SHAvite-512 [76].

Функція дійшла до другого раунду конкурсу криптографічних функцій SHA, але до фіналу не була допущена за недостатню захищеність ініціалізації S-блоків, що лежать в основі блочного шифру, що призводило до відносно низького рівня безпеки 512-розрядної версії [10].

SHAvite-3 добре підходить для різних платформ і машин, як і AES. Завдяки байт орієнтованії структурі і будівельним блокам AES, SHAvite-3 стає «нативним» для 8-бітних, 32-бітових, 64-бітних машин і фактично для будь-якої машини, яка вже постачає або використовує AES.

Алгоритм SHAVITE застосовується як складова в алгоритмах X11, X12, X13, X14, X15 та X17 для майнінгу різних криптовалют [8].

Алгоритм криптографічного ґешування SIMD. Ітеративна криптографічна ґеш-функція SIMD була розроблена Gaëtan Leurent, Charles Bouillaguet, Pierre-Alain Fouque. Була висунута як кандидат на конкурс стандарту SHA-3, де пройшла до другого раунда [10].

Існують два варіанти ґеш-функції: SIMD-256 і SIMD-512, що перетворюють повідомлення довільної довжини в 256 або 512-бітове ґеш-значення, зване також дайджестом повідомлення. Крім того можливо визначити ґеш-функції SIMD-n як усічення функцій SIMD-256 і SIMD-512 для n <256 і 256 <n <512 відповідно [10].

Як стверджують розробники алгоритму, головною особливістю ґеш-функції є значне розширення повідомлення, яке дозволяє захиститися від диференціального криптоаналізу [10].

Сімейство ґеш-функцій SIMD засноване на двох функціях SIMD-256 і SIMD-512. Крім того специфікація визначає SIMD-n з $n \le 256$ як усічення SIMD-256 і SIMD-n з $256 < n \le 512$ як усічення SIMD-512.

Кожна функція SIMD-n приймає, як вхідне, повідомлення довільного розміру, і виводить дайджест (ґеш-код) з n бітів.

Ґеш-функція SIMD була відібрана в якості фіналіста конкурсу SHA-3. Експерти конкурсу відзначили, що, хоча ґеш-функція SIMD багато в чому повторює алгоритми сімейств MD / SHA, але поліпшення, зроблені авторами, дійсно дозволили захистити SIMD від багатьох типів атак (наприклад, колізійна атака). Крім того, зміни, проведені для другого раунду, змогли захистити ґеш-функцію SIMD від атаки на основі диференціального криптоаналізу [10]. Однак високі вимоги до RAM і наявності SIMD інструкцій для гарної продуктивності роблять ґеш-функцію поганим кандидатом для реалізації на FPGA [10]. Головним чином з цієї причини ґеш-функція SIMD не потрапила у фінальну стадію конкурсу.

Алгоритм SIMD застосовується як складова в алгоритмах X11, X12, X13, X14, X15 та X17 для майнінгу різних криптовалют [8].

Алгоритм криптографічного ґешування SKEIN. Алгоритм ґешування Skein (англ. Skein) змінної розрядності було розроблено групою авторів на чолі з Брюсом Шнайєром [77].

Ґеш-функція Skein була створена в 2008 р. і увійшла до п'ятірки фіналістів конкурсу SHA-3, однак в 2012 р. у фіналі переможцем був обраний алгоритм Keccak, найбільш продуктивний і нечутливий до вразливостей SHA-2 [78]. Назва ґеш-функції Skein означає «моток пряжі».

Ґеш-функція Skein виконана як універсальний криптографічний примітив, на основі блочного шифру Threefish, що працює в режимі UBI-ґешування. Основною концепцією розробки була оптимізація під мінімальне використання пам'яті, криптографічно безпечне ґешування невеликих повідомлень, стійкість до всіх відомих атак на ґеш-функції, оптимізація під 64-розрядні процесори й активне використання звернень до таблиць.

Skein підтримує розміри внутрішнього стану 256, 512 і 1024 біт і розмір вихідного блоку до 2 64 −1 біт. Автори заявляють про 6.1 такт на байт для будь-якого розміру вихідного блоку на ПК з процесором Intel Core 2 Duo. З числа кандидатів на SHA-3 Skein входить в п'ятірку найшвидших, проте є лідером лише в 64-розрядному варіанті, який перевершує за швидкісними характеристиками 32-розрядний у більш ніж чотири рази. Це пояснюється тим, що автори спочатку орієнтувалися на оптимізацію під 64-розрядні процесори [77].

Skein-512 може бути реалізований з використанням всього 200 байт пам'яті, а спрощена версія Skein-256 – із використанням 100 байт, що оптимально для апаратної реалізації алгоритму в смарт-картах [78].

Як заявляють автори, ґеш-функція Skein на поширених процесорах працює в середньому два рази швидше SHA-512, Threefish в два рази швидше AES.

Skein захищена від нових видів атак на ґеш-функцій – підбору подовжених повідомлень і псевдоколізій.

Threefish, що лежить в основі Skein має дуже просту структуру і може бути використаний для заміни алгоритмів блочного шифрування, будучи швидким і гнучким шифром, що працює в довільному режимі шифрування. Сам Threefish не використовує S-блоки, натомість покладається на комбінації інструкцій XOR, складання і циклічного зсуву.

Область застосування Skein досить широка. Використовуючи повідомлення і ключ в якості відповідних входів, можна обчислити КАП. За допомогою аргументу Nonce використовувати Skein в режимі потокового шифру. Також можливе застосування в якості генератора псевдовипадкових чисел, наприклад в алгоритмах Fortuna і Yarrow, як Key Derivation Function і Password-Based Key Derivation Function (використовуючи аргументи Key і Key Derivation Identifer), в якості ґеш-функції для обчислення електронного підпису (мається на увазі використання аргументу Public Key).

Ґеш-функція Skein застосовується як складова у складі алгоритмів майнінгу криптовалют X11, X12, X13, X14, X15, X17. Одна з відомих криптовалют, яка видобувається на алгоритмі Skein через майнінг, є DigiByte (DGB).

Алгоритм криптографічного ґешування SNEFRU. Криптографічна ґеш-функція Snefru була запропонована Ральфом Меркле (сама назва Snefru, продовжуючи традиції блокових шифрів Khufu і Khafre, також розроблених Ральфом Меркле, являє собою ім'я єгипетського фараона) [79, 80]. Функція Snefru перетворює повідомлення довільної довжини в ґеш довжини m (зазвичай m = 128 або m = 256).

Snefru – це ітеративна ґеш-функція, яка спирається на будову Меркла – Дамгара. Її було спроектовано як таку, яка ґешує повідомлення довільної довжини в 128-бітові значення (також був представлений 256-бітний варіант за тією самою будовою). Snefru використовує схему заповнення, яка завжди додає додатковий блок заповнення з довжиною повідомлення (на відміну від компактнішої схеми заповнення MD4 , яка додає ще один блок тільки за потреби).

Використовуючи засоби диференційного аналізу, Елі Біхам і Аді Шамір показали, що двопрохідні функція Snefru не є стійкою до колізій 1-го роду і 2-го роду [81].

Алгоритм криптографічного ґешування STREEBOG. Чинний російський криптографічний стандарт ГОСТ Р 34.11-2012 «Информационная технология. Криптографическая защита информации. Функция ґеширования» визначає алгоритм і процедуру обчислення ґеш-функції СТРИБОГ. Алгоритм було розроблено Центром захисту інформації та спеціального зв'язку ФСБ Росії за участю ВАТ «ІнфоТеКС» [82] і введений в дію 1 січня 2013 р. [83].

Стандарт визначає алгоритм і процедуру обчислення ґеш-функції для послідовності символів. Цей стандарт розроблений і введений в якості заміни застарілого стандарту ГОСТ Р 34.11-94.

Основними параметрами алгоритму є: розмір ґешу – 256 або 512 біт; розмір блоку вхідних даних – 512 біт.

На конференції Crypto-2015 Алекс Бірюков, Лео Перрін і Олексій Удовенко представили доповідь, в якій говориться про те, що значення S-блоку блокового симетричного шифру Кузнечик і ґеш-функції Стрибог не є (псевдо) випадковими числами, а згенеровані на основі прихованого алгоритму, який доповідачам вдалося відновити методами зворотного проектування [84]. 29 січня 2019 р. було опубліковано дослідження «Partitions in the S-Box of Streebog and Kuznyechik» [85], яке спростовує заяву авторів про випадковий виборі параметрів таблиць замін в алгоритмах Стрибог і Кузнечик [86].

Платформа блокчейна izzz.io з відкритим вихідним кодом і інтелектуальними контрактами з застосовує криптографічні бібліотеки із реалізованим алгоритмом STREEBOG. Цю платформу застосовують компанії BigNet, BitCoen, Buzcoin, Baikalika, NWP Solution, SBS Platform, NS Platform [87].

Алгоритм криптографічного ґешування TIGER. Криптографічна ґеш-функція Tiger була розроблена Росом Андерсоном і Елі Біхамом в 1995 р. [88].

Tiger був призначений для особливо швидкого виконання на 64-розрядних комп'ютерах [89]. Tiger не має патентних обмежень, може використовуватися вільно як з еталонною реалізацією, так і з її модифікаціями.

Розмір значення ґешу – 192 біта (Tiger / 192), хоча є також більш короткі версії для сумісності з SHA-1 (Tiger / 160) і з MD4, MD5, RIPEMD, Snefru (Tiger / 128). Швидкість роботи – 132 Мбіт / с (перевірено на одному процесорі Alpha 7000, модель 660). На сучасних процесорах значно швидше (навіть при тесті на 32-бітному AMD Sempron 3000+ швидкість близько 225 Мбіт / с) [90].

Tiger2 – версія Tiger, яка відрізняється від основної тільки іншим алгоритмом додавання бітів, подібним з MD5 / SHA-1.

Tiger використовується в технології TTH, де ґеш обчислюється в деревовидній формі. TTH, в свою чергу, застосовується в протоколах файлового обміну Gnutella, Gnutella2, Direct Connect, а також в файлообмінниках Phex, BearShare, LimeWire, Shareaza, DC ++ і Valknut.

Алгоритм криптографічного ґешування WHIRLPOOL. Криптографічна ґеш-функція Whirlpool була розроблена Вінсентом Рейменом і Пауло Баррето [91]. Опублікована в листопаді 2000 р. Ґешує вхідне повідомлення з довжиною до 2^{256} бітів. Вихідне значення ґеш-функції Whirlpool становить 512 бітів.

Ґеш-функція Whirlpool названа в честь Галактики Вир (M51) в сузір'ї Гончі Пси – першої відкритої галактики з спіральною структурою.

З моменту створення в 2000 р. Whirlpool двічі модифікувалася.

Перша версія Whirlpool-0 була представлена як кандидат в проекті NESSIE (англ. New European Schemes for Signatures, Integrity and Encryption, Нові Європейські Проекти з Цифрового Підпису, Цілісності та Шифрування).

Модифікація Whirlpool-0, названа Whirlpool-T, в 2003 р. додана до переліку рекомендованих до використання криптографічних функцій NESSIE. Зміни стосувалися блоку підстановки (S-box) Whirlpool: в першій версії структура S-box була описана, і він генерувався довільно, що створювало певні проблеми при апаратній реалізації Whirlpool. У версії Whirlpool-T S-box «придбав» чітку структуру.

Дефект в дифузних матрицях Whirlpool-T, виявлений Тайдзо Сірано і Кёдзі Сібутані, згодом виправлений, і кінцева (третя) версія, названа для стислості просто Whirlpool, прийнята ISO в стандарті ISO / IEC 10118-3 : 2004 у 2004 р.

Ґеш-функція WHIRLPOOL застосовується як складова алгоритмів майнінгу криптовалют X14, X15, X17.

Алгоритми криптографічного ґешування сімейства «X». У 2014 р. був представлений новий тип криптографічного методу ґешування – алгоритм X11, а трохи пізніше його більш досконалі версії X12, X13, X14, X15 і X17. Перша криптовалюта, блокчейн якої побудований на X11, була DarkCoin [92].

Число в назві алгоритму позначає кількість раундів ґешування і видів функцій, які використовує даний алгоритм. Наприклад, алгоритм X13 використовує 13 ґеш-циклів з 13 різними криптографічними функціями, що робить його одним з найбільш надійних в сучасному світі криптовалют [8].

В табл. 1 наведено перелік використовуваних функцій в кожній алгоритмічної версії.

Таблиця 1

Перелік функцій в кожній версії алгоритму майнінга сімейства «X»

Ґеш-функція	X11	X12	X13	X14	X15	X17
BLAKE	+	+	+	+	+	+
BMW	+	+	+	+	+	+
GROESTL	+	+	+	+	+	+
J-H	+	+	+	+	+	+
KECCAK	+	+	+	+	+	+
SKEIN	+	+	+	+	+	+
LUFFA	+	+	+	+	+	+
CUBEHASH	+	+	+	+	+	+
SHAVITE	+	+	+	+	+	+
SIMD	+	+	+	+	+	+
ECHO	+	+	+	+	+	+
HAMSI		+	+	+	+	+
FUGUE			+	+	+	+
SHABAL				+	+	+
WHIRLPOOL					+	+
LOSELOSE						+
DJB-2						+

Початкове завдання X11 – запобігання проблем з централізацією системи Dash [93]. Надмірна простота SHA-256 могла стати причиною різкого цінового обвалу криптовалюти, оскільки велика ймовірність того, що велика частина цифрової валюти буде зосереджена в руках кількох впливових пулів.

X-алгоритми були створені спеціально для роботи на графічних процесорах, де вони забезпечують високу рентабельність і низьке енергоспоживання. Кожен результат підфункції потім передається в наступний під-алгоритм і так відбувається X раз. Таким чином, створення ASIC-обчислювачів для такого методу буде ускладнене, так як апаратне забезпечення повинне буде оптимізоване під кожен алгоритм, що сильно збільшує складність виробництва і вартість обчислювального обладнання. Можливо згодом виробники ASIC-обчислювачів зможуть розробити моделі для алгоритмів серії X (наприклад, для X11 таке обладнання вже є), але доцільність його використання знаходиться під великим питанням. Щоб зламати, наприклад, X13, потрібно знайти вразливість у всіх 13 ґешах, що набагато складніше, ніж для одного алгоритму, наприклад, SHA-256.

На сьогодні алгоритми ґешування сімейства «X» широко застосовуються в сучасних блокчейн-системах, зокрема криптовалютах. Наприклад, алгоритм X11 застосовуються для майнінгу наступних криптовалют: DeepOnion (ONION); Cloakcoin (CLOAK); MaruCoin (MARU); Hshare (HSR); Stealthcoin (XST); MarteXcoin (MXT). Алгоритм X16 використовується як основний засіб майнінгу у криптовалютах: Stone Coin; Ravencoin; Proton Coin; Gravium; HTHCoin; Motion. І цей перелік постійно зростає, отже поширюється практичне застосування алгоритмів криптографічного ґешування сімейства «X».

Проведений аналіз показує, що різні за своєю побудовою алгоритми застосовують різні математичні перетворення та окремі функціональні схеми. Розглянуті алгоритми ґешування застосовуються (або можуть застосовуватися) як основний криптографічний елемент технології блокчейн, тобто вони застосовуються у понад 90 % існуючих проектів децентралізованих систем [8, 93, 94].

Висновки

В роботі розглянуто переважну більшість відомих та широко розповсюджених алгоритмів ґешування, які застосовуються або можуть бути застосовані найближчим часом у протоколах консенсусу сучасних блокчейн-мереж. Проаналізовано як стандартизовані на міжнародному та національному рівні алгоритми ґешування, так і ґеш-функції, які були представлені на різних криптографічних конкурсах на науково-пошукових проектах. Зокрема встановлено, що більшість проектів розподілених мереж використовує надійні та перевірені часом алгоритми криптографічного ґешування (наприклад, алгоритми KECCAK, SHA2, RIPEMD160, тощо). Але останніми роками для захисту від ASIC-майнерів почали застосовуватися і інші ґеш-функції, які хоча і можуть бути навіть швидшими за KECCAK, SHA2 або RIPEMD160, але володіють певними вразливостями стосовно властивостей необоротності. Як приклад можна навести застосування криптографічних алгоритмів MD4, EDONR-256, EDONR-512, ED2K (надійність та безпечність яких на сьогоднішній день є незадовільною), або навіть найпростійших функцій DJB-2 та LOSELOSE (ці алгоритми не є криптографічними, а являють собою, по суті, звичайну контрольну суму). Через простоту обчислення певних показників не можна нехтувати порушенням властивостей необоротності, особливо якщо саме на них базуються основні переваги блокчейн-мереж. Отже перспективним є дослідження як швидкодії криптографічного ґешування, так і безпеки відповідних алгоритмів, що буде розглянуто у наступних роботах.

Список літератури:

1. The password hash Argon2, winner of PHC. Електронний ресурс. Режим доступу: https://github.com/P-H-C/phc-winner-argon2

2. Argon2. By Dmitry Khovratovich. 30 March 2015. Електронний ресурс. Режим доступу: https://www.cryptolux.org/index.php/Argon2

3. Bitcoin Forum. Електронний ресурс. Режим доступу: https://bitcointalk.org/index.php?topic=1318683.0

4. Dan Boneh, Henry Corrigan-Gibbs, and Stuart Schechter. Balloon Hashing: A Memory-Hard Function Providing Provable Protection Against Sequential Attacks. 12.05.2017. Електронний ресурс. Режим доступу: https://eprint.iacr.org/2016/027.pdf

5. SHA-3 proposal BLAKE. Електронний ресурс. Режим доступу: https://131002.net/blake/

6. BLAKE2 — fast secure hashing. Електронний ресурс. Режим доступу: https://blake2.net/

7. About Blakecoin. Електронний ресурс. Режим доступу: https://blakecoin.org/about-blakecoin/

8. Алгоритм X13 для майнинга на графических процессорах. Александр Марков. 28 мая 2018. Електронний ресурс. Режим доступу: https://miningbitcoinguide.com/mining/sposoby/x13

9. Danilo Gligoroski, Vlastimil Klima, Svein Johan Knapskog, Mohamed El-Hadedy, Jørn Amundsen, Stig Frode Mjølsnes. Blue Midnight Wish. Trondheim, Norway: Norwegian University of Science and Technology, 2008. P. 71.

10. NIST. Announcing request for candidate algorithm nominations for a new cryptographic hash algorithm (SHA-3) family. Federal Register, 72(112), November 2007. Електронний ресурс. Режим доступу: http://csrc.nist.gov/groups/ST/hash/documents/FR_Notice_Nov07.pdf

11. CubeHash specification (2.B.1) Daniel J. Bernstein. Електронний ресурс. Режим доступу: http://cubehash.cr.yp.to/submission2/spec.pdf

12. CubeHash efficiency estimates (2.B.2). By Daniel J. Bernstein. Електронний ресурс. Режим доступу: http://cubehash.cr.yp.to/submission/estimates.pdf

13. CubeHash parameter tweak: 16 times faster.By Daniel J. Bernstein. Електронний ресурс. Режим доступу: http://cubehash.cr.yp.to/submission/tweak.pdf

14. Single Block Attacks and Statistical Tests on CubeHash. By Benjamin Bloom, Alan Kaminsky. August 21, 2009. Електронний ресурс. Режим доступу: http://scholarworks.rit.edu/cgi/viewcontent.cgi?article=1986&context=article

15. Bernstein hash djb2. Електронний ресурс. Режим доступу: https://riot-os.org/api/group__sys__hashes__djb2.html

16. ECHO hash function. Електронний ресурс. Режим доступу: https://crypto.orange-labs.fr/echo/

17. Ed2k-hash. 7 May 2005. Електронний ресурс. Режим доступу: https://wiki.anidb.info/w/Ed2k-hash

18. ed2k-tools. Tools for eDonkey2000 and Overnet. Електронний ресурс. Режим доступу:http://ed2k-tools.sourceforge.net/index.shtml

19. Edon–R, An Infinite Family of Cryptographic Hash Functions. Danilo Gligoroski, Smile Markovski and Ljupco Kocarev. May 2009. Електронний ресурс. Режим доступу: https://pdfs.semanticscholar.org/e901/492cbb9d1f8a4365397676da808a9d9415cc.pdf

20. D. Gligoroski et al. Cryptographic hash function Edon-R′ // 2009 Proceedings of the 1st International Workshop on Security and Communication Networks, Trondheim, 2009, pp. 1-9.

21. The Ethereum Wiki Електронний ресурс. Режим доступу: https://github.com/ethereum/wiki

22. Dagger Hashimoto. Електронний ресурс. Режим доступу: Https://github.com/ethereum/wiki/wiki/Dagger-Hashimoto

23. Ethash Design Rationale. Електронний ресурс. Режим доступу: Https://github.com/ethereum/wiki/wiki/Ethash-Design-Rationale

24. NISTIR 7764. Status Report on the Second Round of the SHA-3 Cryptographic Hash Algorithm Competition. Електронний ресурс. Режим доступу: https://csrc.nist.gov/publications/detail/nistir/7764/final

25. Hash Function Fugue. Електронний ресурс. Режим доступу: https://researcher.watson.ibm.com/researcher/view_group.php?id=3302

26. Криптографическая защита информации функция хэширования. ГОСТ Р 34.11-94. Електронний ресурс. Режим доступу: https://pdf.standartgost.ru/catalog/Data2/1/4294824/4294824580.pdf

27. Министерство промышленности и торговли Российской Федерации. Федеральное агентство по техническому регулированию и метрологии. Об утверждении национального стандарта. ПРИКАЗ от 7 августа 2012 года N 216-ст. Електронний ресурс. Режим доступу: http://docs.cntd.ru/document/902368268

28. ГОСТ 34.311-95. Информационная технология. Криптографическая защита информации. Функция хэширования. Дата введения 1995-01-01. Електронний ресурс. Режим доступу: http://docs.cntd.ru/document/gost-34-311-95

29. Системы обработки информации. Защита криптографическая. ГОСТ 28147-89. Дата введения 01.07.90. Електронний ресурс. Режим доступу: https://files.stroyinf.ru/Data2/1/4294826/4294826631.pdf

30. Grøstl a SHA-3 candidate. By Praveen Gauravaram, Lars R. Knudsen, Krystian Matusiewicz, Florian Mendel, Christian Rechberger, Martin Schlaffer, and Søren S. Thomsen. March 2, 2011. Електронний ресурс. Режим доступу: http://www.groestl.info/Groestl.pdf

31. Grøstl a SHA-3 candidate. Електронний ресурс. Режим доступу: http://www.groestl.info/team.html

32. Verge Currency. Електронний ресурс. Режим доступу: https://vergecurrency.com/

33. Grindahl a family of hash functions. Lars R. Knudsen, Christian Rechberger and Søren S. Thomsen Електронний ресурс. Режим доступу: https://web.archive.org/web/20120915162204/http://www2.mat.dtu.dk/people/Lars.R.Knudsen/grindahl/grindahl.pdf

34. SERPENT. A Candidate Block Cipher for the Advanced Encryption Standard. Електронний ресурс. Режим доступу: https://www.cl.cam.ac.uk/~rja14/serpent.html

35. Telecommunications Technology Association. Hash Function Standard Part 2: Hash Function Algorithm Standard (HAS-160). TTAS.KO-12.0011/R1, December 2000. Електронний ресурс. Режим доступу: https://www.tta.or.kr/include/Download.jsp?filename=stnfile/TTA-0072.pdf

36. A Description of HAS-160. 2002-10-01. Електронний ресурс. Режим доступу: https://www.randombit.net/has160.html

37. JH. Електронний ресурс. Режим доступу: https://ehash.iaik.tugraz.at/wiki/JH

38. María Naya-Plasencia, Deniz Toz, Kerem Varici. Rebound Attack on JH42 // Advances in Cryptology ASIACRYPT 2011, Vol. 7073 of Lecture Notes in Computer Science, 2011, pp. 252-269, Springer, 2011. Електронний ресурс. Режим доступу: https://link.springer.com/chapter/10.1007/978-3-642-25385-0_14

39. The sponge and duplex constructions. By Team Keccak: Guido Bertoni, Joan Daemen, Seth Hoffert, Michaël Peeters, Gilles Van Assche and Ronny Van Keer. Електронний ресурс. Режим доступу: https://keccak.team/sponge_duplex.html

40. NIST Releases SHA-3 Cryptographic Hash Standard. August 05, 2015. Електронний ресурс. Режим доступу: https://www.nist.gov/news-events/news/2015/08/nist-releases-sha-3-cryptographic-hash-standard

41. NISTIR 7896 Third-Round Report of the SHA-3 Cryptographic Hash Algorithm Competition. Електронний ресурс. Режим доступу: https://nvlpubs.nist.gov/nistpubs/ir/2012/NIST.IR.7896.pdf

42. SHA-3 Coins. Електронний ресурс. Режим доступу: https://cryptorival.com/algorithms/sha3/

43. Keccak hashing algorithm (SHA-3) Keccak Coins and miner for Keccak. Електронний ресурс. Режим доступу: https://coinguides.org/keccak-algorithm-miner-coins/

44. Наказ «Про прийняття національних стандартів України, гармонізованих з європейськими стандартами, міжнародних стандартів як національних стандартів України, затвердження національних стандартів України, скасування міждержавних стандартів в Україні та внесення зміни до наказу Державного комітету стандартизації, метрології та сертифікації України від 12.06.2002 № 357» Електронний ресурс. Режим доступу: https://zakon.rada.gov.ua/rada/show/v1431731-14

45. A New Standard of Ukraine: The Kupyna Hash Function. Roman Oliynykov1, Ivan Gorbenko, Oleksandr Kazymyrov,Victor Ruzhentsev, Oleksandr Kuznetsov, Yurii Gorbenko, Artem Boiko, Oleksandr Dyrda, Viktor Dolgov, Andrii Pushkaryov. Електронний ресурс. Режим доступу: https://eprint.iacr.org/2015/885.pdf

46. Наказ «Про затвердження вимог до форматів, структури та протоколів, що реалізуються у надійних засобах електронного цифрового підпису». 20 серпня 2012 р. Електронний ресурс. Режим доступу: https://zakon.rada.gov.ua/laws/show/z1398-12

47. The C Programming Language by Brian W. Kernighan (1978-02-22) Paperback, Prentice Hall, 178 p.

48. The Hash Function Family Luffa (Round 2 Archive). Електронний ресурс. Режим доступу: http://www.hitachi.com/rd/yrl/crypto/luffa/index.html

49. Finding Collisions for Reduced Luffa-256 v2. By Bart Preneel, Hirotaka Yoshida, and Dai Watanabe. Електронний ресурс. Режим доступу: http://www.hitachi.com/rd/yrl/crypto/luffa/FindingCollisionsForReducedLuffa-256v2_20101108.pdf

50. mproving the performance of Luffa Hash Algorithm. Thomaz Oliveira1, Julio Lopez. August 19, 2010. Електронний ресурс. Режим доступу: https://eprint.iacr.org/2010/457.pdf

51. Lyra2RE A new PoW algorithm for an ASIC-free future. By Vertcoin Developers. Електронний ресурс. Режим доступу: https://cryptorating.eu/whitepapers/Vertcoin/Vertcoin_Lyra2RE_Paper_11292014.pdf

52. Lyra2REv2. Електронний ресурс. Режим доступу: https://en.bitcoinwiki.org/wiki/Lyra2REv2

53. MD4 Message Digest Algorithm. RFC 1186. Last updated 2013-03-02. Електронний ресурс. Режим доступу: https://datatracker.ietf.org/doc/rfc1186/

54. The MD5 Message-Digest Algorithm. Електронний ресурс. Режим доступу: https://www.ietf.org/rfc/rfc1321.txt

55. MD5 vulnerable to collision attacks. Електронний ресурс. Режим доступу: https://www.kb.cert.org/vuls/id/836068/

56. A quarter of major CMSs use outdated MD5 as the default password hashing scheme. Електронний ресурс. Режим доступу: https://www.zdnet.com/article/a-quarter-of-major-cmss-use-outdated-md5-as-the-default-password-hashing-scheme/

57. The Panama Cryptographic Function. By Joan Daemen and Craig Clapp, December 01, 1998. Електронний ресурс. Режим доступу: http://www.drdobbs.com/security/the-panama-cryptographic-function/184410745

58. Joan Daemen and Craig Clapp. Fast Hashing and Stream Encryption with Panama. Електронний ресурс. Режим доступу: https://link.springer.com/content/pdf/10.1007/3-540-69710-1_5.pdf

59. Обзор «асикоустойчивого» алгоритма ProgPOW для GPU-майнинга / Александр Марков. 10 октября 2018. Електронний ресурс. Режим доступу: https://miningbitcoinguide.com/mining/sposoby/progpow

60. Company Coinmarket. Електронний ресурс. Режим доступу: https://coinmarket.news/2019/01/20/progpow-obzor-svezhih-majnerov-dlya-novogo-algoritma/

61. Equihash: Asymmetric Proof-of-Work Based on the Generalized Birthday Problem / Dmitry Khovratovich, Alex Biryukov. Електронний ресурс. Режим доступу: http://orbilu.uni.lu/bitstream/10993/22277/2/946.pdf

62. Proof of work algorithm based on random code execution. RandomX. Електронний ресурс. Режим доступу: https://github.com/tevador/RandomX

63. The hash function RIPEMD-160. Електронний ресурс. Режим доступу: http://homes.esat.kuleuven.be/~bosselae/ripemd160.html

64. Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD. By Xiaoyun Wang , Dengguo Feng, Xuejia Lai3, Hongbo Yu. August 17, 2004. Електронний ресурс. Режим доступу: http://eprint.iacr.org/2004/199.pdf

65. Cryptography behind top 20 cryptocurrencies. Електронний ресурс. Режим доступу: https://www.susanka.eu/coins-crypto/

66. Colin Percival. Stronger key derivation via sequential memory-hard functions. 2009. Електронний ресурс. Режим доступу: https://en.bitcoinwiki.org/wiki/Scrypt http://www.tarsnap.com/scrypt/scrypt.pdf

67. US Secure Hash Algorithm 1 (SHA1). By P. Jones. September 2001. Електронний ресурс. Режим доступу: https://www.ietf.org/rfc/rfc3174.txt

68. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си = Applied Cryptography. Protocols, Algorithms and Source Code in C. Москва : Триумф, 2002. 816 с.

69. Secure Hash Standard. Federal Information. Processing Standards Publication 180-2. 2002 August 1. (FIPS PUB 180-2) Електронний ресурс. Режим доступу: http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf

70. FIPS Publication 180-2 (with Change Notice 1). Електронний ресурс. Режим доступу: https://csrc.nist.gov/csrc/media/publications/fips/180/2/archive/2002-08-01/documents/fips180-2withchangenotice.pdf

71. Secure Hash Standard (SHS). FIPS PUB 180-3. October 2008. Електронний ресурс. Режим доступу: http://csrc.nist.gov/publications/fips/fips180-3/fips180-3_final.pdf

72. Secure Hash Standar. FIPS PUB 180-4. August 2015. Електронний ресурс. Режим доступу: http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf

73. SHA-256 Coins. Електронний ресурс. Режим доступу: https://cryptorival.com/algorithms/sha256/

74. Shabal, a Submission to NIST's Cryptographic Hash Algorithm Competition. Initiated by the Saphir project. 28.10.2008 Електронний ресурс. Режим доступу: https://www.cs.rit.edu/~ark/20090927/Round2Candidates/Shabal.pdf

75. Status Report on the Second Round of the SHA-3 Cryptographic Hash Algorithm Competition / Meltem Sönmez Turan, Ray Perlner, Lawrence E. Bassham, William Burr, Donghoon Chang, Shu-jen Chang, Morris J.

Dworkin, John M. Kelsey, Souradyuti Paul, Rene Peralta. 12.2011 Електронний ресурс. Режим доступу: https://nvlpubs.nist.gov/nistpubs/Legacy/IR/nistir7764.pdf

76. The SHAvite-3 Hash Function. By Eli Biham and Orr Dunkelman. Електронний ресурс. Режим доступу: http://www.cs.technion.ac.il/~orrd/SHAvite-3/Spec.31.10.08.pdf

77. The Skein Hash Function Family Version 1.3 1 Oct 2010. By Niels Ferguson, Stefan Lucks, Bruce Schneier, Doug Whiting, Mihir Bellare, Tadayoshi Kohno, Jon Callas, Jesse Walker. Електронний ресурс. Режим доступу: http://www.skein-hash.info/sites/default/files/skein1.3.pdf

78. NIST Selects Winner of Secure Hash Algorithm (SHA-3) Competition. Created October 02, 2012, Updated December 11, 2018 Електронний ресурс. Режим доступу: https://www.nist.gov/news-events/news/2012/10/nist-selects-winner-secure-hash-algorithm-sha-3-competition

79. Ralph C. Merkle. A fast software one-way hash function // Journal of Cryptology. 1990. 3 (1): 43–58.

80. Cryptohash: snefru256. Електронний ресурс. Режим доступу: https://snefru256.cryptohash.net/

81. Eli Biham, Adi Shamir. Differential cryptanalysis of Snefru, Khafre, REDOC-II, LOKI and Lucifer (Extended Abstract)

82. Company Infotecs. Електронний ресурс. Режим доступу: http://www.infotecs.ru/

83. ГОСТ Р 34.11-2012. Информационная технология. Криптографическая защита информации. Функция хэширования. Дата введения 2013-01-01. Електронний ресурс. Режим доступу: http://docs.cntd.ru/document/gost-r-34-11-2012

84. Конкурс «Streebog». Открытый конкурс научно-исследовательских работ, посвященных анализу криптографических качеств хэш-функции ГОСТ Р 34.11-2012. Електронний ресурс. Режим доступу: http://www.streebog.info/

85. Cryptology ePrint Archive: Report 2019/092 Partitions in the S-Box of Streebog and Kuznyechik. By Léo Perrin. 29 Jan 2019. Електронний ресурс. Режим доступу: https://eprint.iacr.org/2019/092

86. Очередные странности в алгоритмах ГОСТ Кузнечик и Стрибог. 11 февраля 2019. Електронний ресурс. Режим доступу: https://habr.com/ru/company/virgilsecurity/blog/439788/

87. IZZZIO. Електронний ресурс. Режим доступу: https://en.bitcoinwiki.org/wiki/IZZZIO

88. A Tiger Hash Implementation for C#. 10 Mar 2012. Електронний ресурс. Режим доступу: https://www.codeproject.com/Articles/149061/A-Tiger-Hash-Implementation-for-C

89. Електронний ресурс. Режим доступу: http://th.informatik.uni-mannheim.de/People/Lucks/papers/Tiger_FSE_v10.pdf

90. Cryptanalysis of the Tiger Hash Function. Електронний ресурс. Режим доступу: https://online.tugraz.ac.at/tug_online/voe_main2.getvolltext?pDocumentNr=81263

91. LARC Laboratório de Arquitetura e Redes de Computadores. Електронний ресурс. Режим доступу: http://www.larc.usp.br/~pbarreto/WhirlpoolPage.html

92. Company Dash. Електронний ресурс. Режим доступу: http://dash.org/

93. X11 алгоритм добычи криптовалюты с 11 раундами хэширования / Александр Марков. 23 мая 2018 г. Електронний ресурс. Режим доступу: https://miningbitcoinguide.com/mining/sposoby/x11

94. Алгоритмы майнинга криптовалют – таблица 2019 и краткое описание. Електронний ресурс. Режим доступу: https://mining-cryptocurrency.ru/algoritmy-kriptovalyut/.

Харківський національний
університет імені В.Н. Каразіна;
АТ «Інститут інформаційних технологій»;
Адміністрація Державної служби спеціального зв'язку
та захисту інформації України.

Надійшла до редколегії 02.09.2019

УДК 004.056.5 **DOI:10.30837/rt.2019.3.198.06**

О.О. КУЗНЕЦОВ, д-р техн. наук, В.А. ТИМЧЕНКО, К.Є. ЛИСИЦЬКИЙ,
М.Ю. РОДІНКО, М.С. ЛУЦЕНКО, К.Ю. ШЕХАНІН, А.О. КОЛГАТІН

ДОСЛІДЖЕННЯ ШВИДКОДІЇ ТА СТАТИСТИЧНОЇ БЕЗПЕКИ АЛГОРИТМІВ КРИПТОГРАФІЧНОГО ҐЕШУВАННЯ

Вступ

Стаття є продовженням попередніх робіт «Алгоритми криптографічного ґешування, які застосовуються в сучасних блокчейн-системах» та «Дослідження алгоритмів криптографічного ґешування, які застосовуються в сучасних блокчейн-системах».

В цій роботі проводяться порівняльні дослідження алгоритмів криптографічного ґешування, які застосовуються (або можуть застосовуватися) в сучасних децентралізованих блокчейн-системах. Зокрема досліджується швидкодія ґешування на різних десктопних системах, оцінюється кількість тактів обчислювальної системи на один байт (Cycles/byte), обсяг ґешованого повідомлення за одну секунду (MB/s) та кількість сформованих ґеш-кодів за секунду (KHash/s). Додатково проводяться дослідження швидкодії окремих криптографічних функцій ґешування на графічних обчислювачах. Для оцінки статистичної безпеки проводяться дослідження вихідних послідовностей криптографічних функцій ґешування при обробці ними надмірних вхідних даних (які сформовано за допомогою звичайного лічильника). Для порівняльних досліджень показників статистичної безпеки використовується методика NIST STS (Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications), яку рекомендовано Національним інститутом стандартів і технологій США для дослідження генераторів випадкових і псевдовипадкових чисел для криптографічних застосувань.

Порівняльні дослідження на обчислювальних десктоп-системах

Для проведення порівняльних досліджень застосовувалися еталонні та (за наявності) оптимізовані програмні реалізації алгоритмів ґешування інформації.

Дослідження проводилися на обчислювальних десктоп-системах:

- 64-розрядної обчислювальної платформи із застосуванням AMD Ryzen Threadripper 2970WX 24-Core Processor 3.0 GHz.;
- 64-розрядної обчислювальної платформи із застосуванням Intel Core i9-7980 2.60 GHz.

Дослідження швидкодії проводилися за трьома критеріями:

- кількість тактів обчислювальної системи на один байт (Cycles/byte);
- обсяг повідомлення за секунду, MB/s;
- кількість сформованих ґеш-кодів за секунду, KHash/s.

Зазначені критерії характеризують як абсолютні (другий та третій показники), так і питомі показники швидкодії (перший показник). Отримані дані містять результати тестування швидкості ґешування для різних довжин вхідних текстів. Зокрема на вхід кожного алгоритму подавалися повідомлення (послідовні значення лічильника) завдовжки від 2^0 до 2^{20} байтів.

У наступних таблицях наводяться зведені результати порівняльного аналізу швидкодії зазначених алгоритмів за кожною із застосованих обчислювальних десктоп-системах. Зокрема, у табл. 1 наведено зведені результати порівняльних досліджень швидкодії алгоритмів ґешування на 64-розрядної обчислювальної платформи із застосуванням AMD Ryzen Threadripper 2970WX 24-Core Processor 3.0 GHz. При цьому у якості вхідних даних подавався $2^0 = 1$, 2^{10} та 2^{20} байт даних. У табл. 2 наведено результати тестування швидкодії досліджуваних алгоритмів для 64-розрядної обчислювальної платформи Intel Core i9-7980 2.60 GHz (із довжинами вхідних даних 1, 1024 та 1048576) байтів.

Таблиця 1

Результати тестування швидкодії алгоритмів для вхідного блоку даних розміром 2^0 байт
для 64-розрядної обчислювальної платформи (AMD Ryzen Threadripper 2970WX 24-Core Processor 3.0 GHz)

Назва алгоритму	Вхідний текст довжини 2^0 байт			Вхідний текст довжини 2^{10} байт			Вхідний текст довжини 2^{20} байт		
	Cycles/byte	**MB/s**	**KHash/s**	**Cycles/byte**	**MB/s**	**KHash/s**	**Cycles/byte**	**MB/s**	**KHash/s**
ARGON2D	2625530499,00	0,000001	0,001128	2605872,83	0,001149	0,001122	2531,66	1,182493	0,001128
ARGON2I	2271644943,00	0,000001	0,001315	2232922,54	0,001335	0,001304	2180,86	1,367934	0,001305
BLAKE-224	769,39	3,89	3892,12	11,67	256,69	250,67	10,89	274,93	0,26
BLAKE-256	776,22	3,84	3842,20	11,77	254,39	248,42	10,98	272,71	0,26
BLAKE-384	1010,83	2,96	2963,25	8,18	366,63	358,04	7,15	418,59	0,40
BLAKE-512	1034,26	2,93	2931,19	8,21	364,85	356,30	7,17	417,26	0,40
BMW-224	693,03	4,32	4319,57	5,92	505,83	493,97	5,23	572,68	0,55
BMW-256	690,86	4,34	4336,18	5,89	508,28	496,36	5,20	575,82	0,55
BMW-384	729,17	4,10	4100,48	3,52	851,12	831,17	2,79	1073,26	1,02
BMW-512	729,74	4,10	4104,98	3,51	853,89	833,88	2,78	1076,57	1,03
CUBEHASH-224	5240,57	0,57	567,60	20,31	147,48	144,02	15,17	197,17	0,19
CUBEHASH-256	5286,08	0,57	566,63	20,54	145,53	142,12	15,35	195,16	0,19
CUBEHASH-384	5268,51	0,57	572,83	20,32	147,48	144,02	15,17	197,58	0,19
CUBEHASH-512	5249,72	0,57	570,21	20,37	147,02	143,58	15,19	197,10	0,19
DJB-2	$2{,}75\cdot10^{-5}$	$2{,}33\cdot10^{7}$	$2{,}33\cdot10^{10}$	$2{,}52\cdot10^{-5}$	$2{,}13\cdot10^{7}$	$2{,}08\cdot10^{7}$	$2{,}48\cdot10^{-5}$	$1{,}91\cdot10^{7}$	18181,8
ECHO-224	4755,88	0,63	629,14	27,72	108,21	105,68	24,56	121,91	0,12
ECHO-256	4696,26	0,64	637,73	27,40	109,31	106,74	24,32	123,17	0,12
ECHO-384	5861,37	0,51	511,03	52,56	57,01	55,67	45,92	65,23	0,06
ECHO-512	5865,36	0,51	510,73	51,13	58,53	57,16	45,43	65,90	0,06
ED2K	291,07	10,24	10239,00	4,02	747,38	729,86	3,75	798,00	0,76
EDONR-256	274,37	11,10	11097,22	3,89	765,94	747,99	3,66	816,65	0,78
EDONR-512	290,55	10,34	10338,95	2,17	1377,89	1345,60	1,91	1569,72	1,50
FUGUE-224	2187,69	1,37	1371,91	19,23	155,81	152,15	17,11	175,14	0,17
FUGUE-256	2144,03	1,40	1396,28	19,13	156,53	152,86	17,06	175,52	0,17
FUGUE-384	3244,03	0,92	923,73	31,62	98,87	96,55	26,40	113,49	0,11
FUGUE-512	4645,23	0,65	645,50	37,46	79,89	78,01	33,01	90,72	0,09
GOST34.11-94	3917,49	0,76	763,72	43,74	68,44	66,83	42,19	70,97	0,07
GROESTL-224	2155,56	1,40	1395,26	23,23	128,93	125,91	21,10	141,83	0,14
GROESTL-256	2132,20	1,40	1404,36	22,98	130,32	127,27	20,88	143,44	0,14
GROESTL-384	5217,52	0,57	574,17	31,30	95,62	93,38	26,67	111,92	0,11
GROESTL-512	5271,46	0,57	572,80	31,39	95,43	93,19	26,25	114,06	0,11

Назва алгоритму	Вхідний текст довжини 2^0 байт			Вхідний текст довжини 2^{10} байт			Вхідний текст довжини 2^{20} байт		
	Cycles/byte	**MB/s**	**KHash/s**	**Cycles/byte**	**MB/s**	**KHash/s**	**Cycles/byte**	**MB/s**	**KHash/s**
HAMSI-224	555,85	5,39	5391,69	32,69	91,67	89,53	32,51	92,07	0,09
HAMSI-256	561,81	5,33	5326,51	32,91	90,99	88,86	32,18	93,02	0,09
HAMSI-384	2035,20	1,48	1481,58	84,47	35,75	34,91	82,00	36,50	0,03
HAMSI-512	1996,16	1,50	1504,52	83,53	35,83	34,99	81,97	36,56	0,03
HAS160	363,10	8,25	8251,31	5,41	554,22	541,23	5,05	592,42	0,56
JH-224	4075,85	0,74	743,45	33,28	90,68	88,56	30,63	97,81	0,09
JH-256	4036,49	0,74	741,85	32,74	91,44	89,30	30,72	97,53	0,09
JH-384	4116,05	0,73	727,22	33,19	90,15	88,03	31,17	96,25	0,09
JH-512	4050,76	0,74	738,80	32,68	91,61	89,46	30,63	97,73	0,09
KECCAK-224	1481,99	2,02	2017,77	10,94	273,21	266,81	9,59	311,15	0,30
KECCAK-256	1467,41	2,04	2041,30	10,98	272,85	266,46	10,21	293,14	0,28
KECCAK-384	1487,79	2,01	2011,35	13,79	218,18	213,07	13,40	223,10	0,21
KECCAK-512	1492,17	2,01	2007,73	20,54	145,72	142,30	19,33	154,91	0,15
KUPYNA-256	1706,85	1,70	1697,96	19,45	157,04	153,36	17,83	173,03	0,17
KUPYNA-512	5519,54	0,54	541,42	34,29	89,77	87,66	28,72	106,81	0,10
LOSELOSE	$2{,}55{\cdot}10^{-5}$	$2{,}33{\cdot}10^{7}$	$2{,}33{\cdot}10^{10}$	$2{,}49{\cdot}10^{-5}$	1,94·107	$1{,}89{\cdot}10^{7}$	$2{,}47{\cdot}10^{-5}$	$1{,}97{\cdot}10^{7}$	18867,9
LUFFA-224	809,50	3,82	3816,75	12,21	240,55	234,92	11,35	265,06	0,25
LUFFA-256	764,97	3,91	3905,89	12,04	248,65	242,83	11,23	266,47	0,25
LUFFA-384	1607,08	1,86	1863,11	17,57	170,58	166,59	15,86	188,42	0,18
LUFFA-512	2124,64	1,41	1411,65	23,54	127,42	124,44	21,36	140,30	0,13
LYRA2REV2	20434,44	0,15	146,51	30,96	96,73	94,46	10,97	273,07	0,26
LYRA2RE	19600,40	0,15	152,81	30,22	99,20	96,88	10,90	275,22	0,26
MD4	264,12	11,35	11354,37	3,97	756,00	738,28	3,74	802,28	0,77
MD5	364,08	8,22	8224,13	5,63	532,00	519,53	5,26	568,03	0,54
PANAMA-256	1859,84	1,61	1610,96	3,42	873,81	853,33	1,58	1906,50	1,82
RIPEMD-160	856,08	3,51	3507,65	13,57	220,71	215,53	12,75	234,84	0,22
SCRYPT	1331505,00	0,00	3,33	883,21	3,41	3,33	23,40	128,19	0,12
SHA1	571,89	5,23	5227,72	8,80	340,12	332,14	8,25	362,95	0,35
SHA2-256	806,83	3,70	3704,69	12,60	237,66	232,09	11,80	253,71	0,24
SHA2-512	1045,45	2,88	2880,07	8,71	344,47	336,40	7,68	389,95	0,37
SHABAL-256	1687,56	1,77	1772,50	8,28	360,71	352,25	6,58	454,32	0,43
SHABAL-512	1677,05	1,79	1785,93	8,32	359,96	351,53	6,55	458,49	0,44
SHAVITE-224	1021,71	2,91	2909,96	16,86	176,32	172,19	15,89	188,29	0,18
SHAVITE-256	1014,74	2,97	2966,94	16,60	180,42	176,19	15,62	191,73	0,18

Назва алгоритму	Вхідний текст довжини 2^0 байт			Вхідний текст довжини 2^{10} байт			Вхідний текст довжини 2^{20} байт		
	Cycles/byte	**MB/s**	**KHash/s**	**Cycles/byte**	**MB/s**	**KHash/s**	**Cycles/byte**	**MB/s**	**KHash/s**
SHAVITE-384	3149,33	0,95	952,58	27,45	109,06	106,50	24,42	122,78	0,12
SHAVITE-512	3112,57	0,96	961,91	27,21	110,04	107,46	24,17	123,90	0,12
SIMD-224	2743,21	1,11	1114,87	22,27	134,05	130,91	20,97	142,84	0,14
SIMD-256	2677,69	1,12	1118,79	22,30	134,35	131,20	20,96	142,82	0,14
SIMD-384	6141,45	0,49	490,09	27,11	110,73	108,13	24,09	124,24	0,12
SIMD-512	6162,56	0,49	485,97	27,04	110,81	108,21	24,00	124,77	0,12
SKEIN-224	582,96	5,15	5151,19	4,57	657,41	642,01	4,22	707,54	0,67
SKEIN-256	591,68	5,06	5062,89	4,59	654,54	639,20	4,25	703,74	0,67
SKEIN-384	598,34	5,03	5026,73	4,69	638,60	623,63	4,34	690,31	0,66
SKEIN-512	583,79	5,14	5136,55	4,53	661,15	645,65	4,21	710,90	0,68
SNEFRU-256	6740,30	0,44	441,18	108,35	27,63	26,99	105,08	28,49	0,03
STREEBOG-256	5278,11	0,57	566,96	35,10	85,31	83,31	29,98	100,02	0,10
STREEBOG-512	5260,38	0,57	569,58	35,01	85,52	83,52	29,86	100,28	0,10
TIGER	319,36	9,37	9371,49	4,25	705,16	688,63	3,95	758,19	0,72
WHIRLPOOL	1157,65	2,59	2586,46	18,05	166,18	162,28	16,93	177,24	0,17
X11	38652,22	0,08	77,50	45,04	66,50	64,95	7,28	411,37	0,39
X12	45946,76	0,07	65,17	52,14	57,40	56,06	7,26	412,83	0,39
X13	52768,44	0,06	56,79	58,66	50,81	49,62	7,22	414,62	0,40
X14	55180,53	0,05	54,29	61,13	48,97	47,83	7,25	413,15	0,39

Таблиця 2

Результати тестування швидкодії алгоритмів ґешування для 64 розрядної обчислювальної платформи (Intel Core i9-7980 2.60 GHz)

Назва алгоритму	Вхідний текст довжини 2^0 байт			Вхідний текст довжини 2^{10} байт			Вхідний текст довжини 2^{20} байт		
	Cycles/byte	**MB/s**	**KHash/s**	**Cycles/byte**	**MB/s**	**KHash/s**	**Cycles/byte**	**MB/s**	**KHash/s**
ARGON2D	1995096226,30	0,000001	0,001308	1904191,27	0,001361	0,001329	1866,23	1,386180	0,001322
ARGON2I	1983073690,80	0,000001	0,001329	1905170,88	0,001360	0,001328	1880,48	1,378617	0,001315
BLAKE-224	662,58	3,94	3936,10	10,25	259,23	253,15	9,20	281,65	0,27
BLAKE-256	648,36	3,99	3991,69	9,80	264,93	258,72	9,22	281,12	0,27
BLAKE-384	868,40	3,03	3025,58	6,63	392,14	382,95	5,81	446,39	0,43
BLAKE-512	846,69	3,06	3059,12	6,74	385,36	376,33	5,92	438,00	0,42
BMW-224	646,35	3,97	3966,77	5,57	462,34	451,50	4,96	524,29	0,50
BMW-256	649,68	3,92	3916,69	5,53	465,62	454,71	4,92	527,19	0,50
BMW-384	626,91	4,13	4131,02	3,01	862,32	842,11	2,39	1073,26	1,02
BMW-512	638,02	4,07	4071,35	3,05	851,12	831,17	2,43	1064,54	1,02
CUBEHASH-224	5272,54	0,49	489,42	20,20	127,94	124,94	14,94	173,40	0,17
CUBEHASH-256	5220,66	0,50	496,62	20,13	128,75	125,74	14,91	173,84	0,17
CUBEHASH-384	5150,19	0,50	503,98	20,71	125,16	122,22	14,69	176,59	0,17
CUBEHASH-512	5244,95	0,49	494,42	20,22	128,20	125,20	14,95	173,15	0,17
DJB-2	$4{,}96\cdot10^{-5}$	$1{,}83\cdot10^{7}$	$1{,}83\cdot10^{10}$	$1{,}53\cdot10^{-5}$	$2{,}27\cdot10^{7}$	$2{,}22\cdot10^{7}$	$1{,}41\cdot10^{-5}$	$2{,}23\cdot10^{7}$	21276,6
ECHO-224	4138,87	0,63	625,13	24,11	107,46	104,94	21,41	121,04	0,12
ECHO-256	4123,82	0,63	628,47	24,05	107,86	105,33	21,35	121,46	0,12
ECHO-384	5106,91	0,51	508,12	46,74	55,45	54,15	41,32	62,69	0,06
ECHO-512	5158,78	0,50	504,41	45,03	57,55	56,21	39,91	64,77	0,06
ED2K	240,10	10,84	10839,12	3,32	780,19	761,90	3,11	835,52	0,80
EDONR-256	265,47	10,15	10153,73	3,69	699,52	683,12	3,44	756,00	0,72
EDONR-512	244,48	10,61	10609,90	1,86	1387,01	1354,50	1,64	1572,08	1,50
FUGUE-224	1925,97	1,35	1348,29	17,63	147,21	143,76	15,62	165,94	0,16
FUGUE-256	1913,48	1,36	1355,17	17,58	147,46	144,00	15,69	165,16	0,16
FUGUE-384	3175,04	0,83	829,77	28,14	92,07	89,91	25,29	102,47	0,10
FUGUE-512	4344,18	0,60	596,73	35,86	72,31	70,62	31,25	82,89	0,08
GOST34.11-94	3311,63	0,79	785,16	36,76	70,44	68,79	35,85	72,40	0,07
GROESTL-224	1879,30	1,38	1379,58	21,74	118,46	115,68	19,22	134,92	0,13
GROESTL-256	1881,80	1,38	1377,75	20,86	124,28	121,37	19,22	136,11	0,13
GROESTL-384	4940,77	0,52	523,82	30,30	85,65	83,64	25,25	102,58	0,10
GROESTL-512	4854,19	0,53	533,60	33,12	85,24	83,25	26,06	102,69	0,10
HAMSI-224	543,88	4,78	4781,69	30,21	85,33	83,33	29,13	88,95	0,08

Назва алгоритму	Вхідний текст довжини 2^0 байт			Вхідний текст довжини 2^{10} байт			Вхідний текст довжини 2^{20} байт		
	Cycles/byte	**MB/s**	**KHash/s**	**Cycles/byte**	**MB/s**	**KHash/s**	**Cycles/byte**	**MB/s**	**KHash/s**
HAMSI-256	536,07	4,86	4857,22	29,49	87,94	85,88	28,94	89,66	0,09
HAMSI-384	2090,89	1,24	1238,35	88,41	29,30	28,61	85,88	30,20	0,03
HAMSI-512	2106,47	1,24	1235,01	87,49	29,31	28,62	85,98	30,33	0,03
HAS160	349,00	7,46	7462,11	5,03	514,51	502,45	4,72	548,99	0,52
JH-224	3640,96	0,70	701,32	30,40	83,81	81,84	28,42	91,23	0,09
JH-256	3648,43	0,71	710,37	29,69	87,48	85,43	27,85	93,26	0,09
JH-384	3748,59	0,69	692,15	30,37	85,42	83,42	28,46	91,12	0,09
JH-512	3706,28	0,70	701,99	30,35	85,60	83,59	27,86	92,64	0,09
KECCAK-224	1352,24	1,92	1916,78	9,45	274,07	267,64	8,36	310,51	0,30
KECCAK-256	1307,78	1,97	1974,12	9,64	268,87	262,56	8,96	289,50	0,28
KECCAK-384	1319,91	1,96	1964,84	12,08	214,70	209,66	11,82	217,68	0,21
KECCAK-512	1269,54	2,04	2041,02	17,41	149,28	145,79	16,40	158,11	0,15
KUPYNA-256	1660,20	1,54	1543,95	19,24	134,85	131,69	17,46	148,59	0,14
KUPYNA-512	5247,62	0,50	500,51	31,92	81,63	79,72	26,86	96,91	0,09
LOSELOSE	$4{,}96\cdot10^{-5}$	$1{,}49\cdot10^{7}$	$1{,}49\cdot10^{10}$	$1{,}41\cdot10^{-5}$	$2{,}55\cdot10^{7}$	$2{,}49\cdot10^{7}$	$1{,}41\cdot10^{-5}$	$2{,}38\cdot10^{7}$	22727,3
LUFFA-224	760,28	3,44	3443,26	11,87	218,09	212,98	11,25	230,56	0,22
LUFFA-256	777,22	3,35	3351,58	12,12	213,82	208,81	11,46	226,62	0,22
LUFFA-384	1622,45	1,59	1590,97	18,07	143,54	140,18	16,34	158,66	0,15
LUFFA-512	2077,64	1,25	1253,18	23,46	110,97	108,37	20,70	125,23	0,12
LYRA2REV2	19363,25	0,13	133,94	28,52	90,86	88,73	9,49	276,67	0,26
LYRA2RE	16422,12	0,16	157,76	25,67	100,92	98,56	9,44	274,50	0,26
MD4	221,46	11,76	11761,93	3,31	783,69	765,32	3,10	836,19	0,80
MD5	317,29	8,21	8206,10	4,91	527,45	515,09	4,62	561,04	0,54
PANAMA-256	1729,69	1,51	1511,79	3,23	806,60	787,69	1,49	1747,63	1,67
RIPEMD-160	743,69	3,49	3489,09	11,85	218,04	212,93	11,15	232,40	0,22
SCRYPT	2463209,70	0,00	1,11	722,61	1,80	1,75	20,56	125,73	0,12
SHA1	505,14	5,14	5136,05	7,81	331,83	324,05	7,41	349,76	0,33
SHA2-256	848,68	3,06	3060,37	13,46	192,54	188,03	12,68	204,88	0,20
SHA2-512	945,51	2,74	2741,59	7,94	327,37	319,70	7,02	369,87	0,35
SHABAL-256	1235,41	2,10	2097,95	6,15	420,61	410,75	4,94	524,55	0,50
SHABAL-512	1274,73	2,04	2039,99	6,31	416,60	406,83	5,01	507,78	0,48
SHAVITE-224	876,95	2,94	2942,55	14,19	180,07	175,85	13,46	191,91	0,18
SHAVITE-256	883,77	2,92	2915,87	14,33	178,63	174,45	13,53	189,93	0,18
SHAVITE-384	2805,29	0,92	922,12	24,41	105,95	103,47	21,67	119,52	0,11

Назва алгоритму	Вхідний текст довжини 2^0 байт			Вхідний текст довжини 2^{10} байт			Вхідний текст довжини 2^{20} байт		
	Cycles/byte	**MB/s**	**KHash/s**	**Cycles/byte**	**MB/s**	**KHash/s**	**Cycles/byte**	**MB/s**	**KHash/s**
SHAVITE-512	2788,42	0,93	925,64	23,77	106,31	103,82	21,04	123,14	0,12
SIMD-224	2319,64	1,10	1095,10	19,98	130,05	127,00	18,78	138,13	0,13
SIMD-256	2335,71	1,12	1115,35	19,32	134,62	131,47	18,14	142,92	0,14
SIMD-384	5564,95	0,47	465,67	24,52	105,58	103,10	21,74	119,05	0,11
SIMD-512	5457,79	0,47	469,52	23,88	108,45	105,91	21,66	119,13	0,11
SKEIN-224	532,88	4,87	4870,75	4,12	629,78	615,02	3,88	669,16	0,64
SKEIN-256	526,25	4,94	4935,40	4,02	643,69	628,61	3,78	684,45	0,65
SKEIN-384	537,25	4,83	4833,48	4,15	627,14	612,44	3,90	666,19	0,64
SKEIN-512	525,12	4,93	4932,85	4,02	645,67	630,54	3,78	685,79	0,65
SNEFRU-256	5557,12	0,47	466,27	89,21	29,02	28,34	86,62	29,95	0,03
STREEBOG-256	4580,11	0,57	566,73	30,40	85,40	83,40	25,96	99,85	0,10
STREEBOG-512	4642,42	0,56	558,79	31,59	82,20	80,28	26,84	96,91	0,09
TIGER	240,20	10,87	10871,71	3,51	739,48	722,14	3,31	787,81	0,75
WHIRLPOOL	975,33	2,66	2659,88	15,57	166,60	162,69	14,62	177,27	0,17
X11	35972,07	0,07	71,73	40,95	63,76	62,26	5,91	439,47	0,42
X12	43318,45	0,06	59,35	48,72	53,19	51,94	5,98	433,12	0,41
X13	49509,65	0,05	52,36	54,36	47,66	46,54	5,93	437,09	0,42
X14	51112,37	0,05	50,75	55,84	46,40	45,31	5,93	437,64	0,42

Аналіз та дослідження показують, що найбільш швидкими є алгоритми ґешування DJB-2 та LOSELOSE. Але, як слідує із їхньої специфікації [1, 2], це не криптографічні алгоритми. Ці функції ґешування обчислюють звичайну контрольну суму і пошук прообразів для таких перетворень є тривіальним.

Далі за швидкістю перетворень йдуть алгоритми MD4, EDONR-256, EDONR-512, ED2K та інші криптографічні перетворення, стійкість яких є на сьогоднішній день не є задовільною [3 – 7].

Більшість криптографічних функцій мають порівняні показники швидкості формування ґеш-кодів [8 – 13], серед яких і алгоритм криптографічного ґешування Купина (національний стандарт України).

Найповільнішими алгоритмами ґешування виявилися ґеш-функції сімейства «X» та алгоритми ARGON2D, ARGON2I та SCRYPT [14-16].

Порівняльні дослідження швидкодії алгоритмів ґешування на графічних обчислювальних системах

Для проведення порівняльних досліджень швидкодії алгоритмів ґешування на графічних обчислювальних системах було обрано наступні апаратні засоби:

- Geforce 740M 2GB;
- Geforce GTX1050ti 4GB;
- Rx580 Aorus 4GB;
- Rx580 Sapphire Pulse 8GB;
- Sapphire Vega 56 8GB.

Проводилися вибіркові дослідження для наступних алгоритмів:

- ГОСТ 34.311;
- СТРИБОГ256;
- СТРИБОГ512;
- KECCAK 256;
- KECCAK 512;
- SHA2 256;
- SHA2 512;
- RIPEMD160;
- Blake2b;
- Wirlpool.

Для проведення досліджень застосовувалося програмне забезпечення HashCat [8]. HashCat – утиліта, яка надає можливість відновлення пароля. Найбільш активно вона використовується для відновлення WPA / WPA2 паролів, а також ключів від зашифрованих офісних документів. З 2015 р. поширюється з відкритим вихідним кодом під ліцензією MIT. Утиліта дозволяє використовувати будь-які пристрої, що реалізують стандарт OpenCL (OpenCL забезпечує паралелізм на рівні інструкцій і на рівні даних і є здійсненням техніки GPGPU. OpenCL є повністю відкритим стандартом, його використання не обкладається ліцензійними відрахуваннями).

Отримані результати тестування наведено у табл. 3 та 4.

В табл. 3 наведено результати Бенчмарку, тобто оцінки швидкості ґешування (кількість сформованих ґеш-кодів за секунду) шляхом послідовного ґешування набору даних.

В табл. 4 наведено питомі показники складності, а саме швидкість ґешування (кількість сформованих ґеш-кодів за секунду), яка приходиться на одне обчислювальне ядро OpenCL застосованого графічного обчислювача.

Таблиця 3

Результат Бенчмарка на графічних обчислювальних системах із застосуванням програмного забезпечення HashCat, Кілоґеш/с

	Geforce 740M 2GB	Geforce GTX1050ti 4GB	Rx580 Aorus 4GB	Rx580 Sapphire Pulse 8GB	Sapphire Vega 56 8GB
ГОСТ 34.311	10442,9	65337,9	89450	91932,3	233100
СТРИБОГ256	3512.6	13613,5	56751,7	55162,9	99485
СТРИБОГ512	3518,4	13569,5	55207,8	56635,9	99641,5
KECCAK 256	38149,7	260400	320400	327800	529600
KECCAK 512	38285	262400	326400	334400	538700
SHA2 256	133400	889600	1700900	1755900	3088100
SHA2 512	38794,4	297300	405600	421600	710100
RIPEMD160	177000	1383100	2356200	2437700	4129100
Blake2b	96515,9	585100	1103300	1132800	1738000
Wirlpool	12516,8	64773,3	324500	333200	591400

На рис. 1 для наочності наведено діаграму швидкостей ґешування із табл. 10. Як бачимо, найшвидшим є алгоритм RIPEMD160, далі слідує алгоритм SHA2, Blake2b та інші.

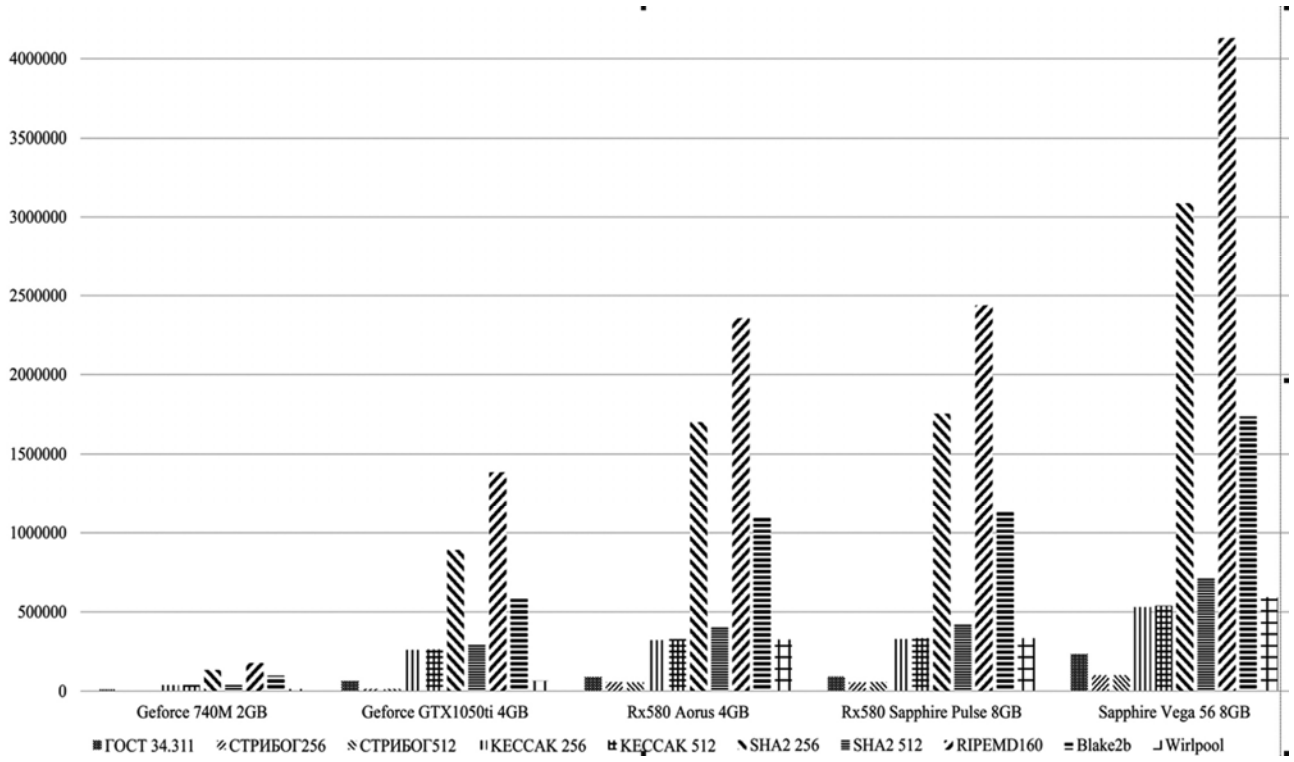


Рис. 1. Порівняння швидкодії алгоритмів ґешування на графічних обчислювачах

На рис. 2 наведено діаграми питомої швидкості, тобто швидкодія ґешування, що приходиться на одне обчислювальне ядро графічного обчислювача. Як бачимо, ранжування алгоритмів за швидкодією майже таке саме (RIPEMD160, SHA2, Blake2b та інш.), але власні значення швидкості змінилися. Тобто кожен графічний пристрій має різну кількість обчислювальних ядер і найбільш швидкими за критерієм питомої швидкодії виглядають обчислювачі Geforce.

Таблиця 4

Швидкість ґешування (кількість сформованих ґеш-кодів за секунду), яка приходиться на одне обчислювальне ядро OpenCL застосованого графічного обчислювача, Кілоґеш/с

	Geforce 740M 2GB	Geforce GTX1050ti 4GB	Rx580 Aorus 4GB	Rx580 Sapphire Pulse 8GB	Sapphire Vega 56 8GB
ГОСТ 34.311	5221,45	10889,65	2484,7	2553,6	4162,5
СТРИБОГ256	2706.3	2268,9	1576,4	1532,3	1776,5
СТРИБОГ512	1759,2	2261,5	1533,5	1573,2	1779,3
KECCAK 256	19074,8	43400	8900	9105,5	9457,1
KECCAK 512	19142,5	43733,3	9066,6	9288,8	9619,6
SHA2 256	66700	148266,6	47247,2	48775	55144,6
SHA2 512	19397,2	49550	11266,6	11711,1	12680,3
RIPEMD160	88500	230516,6	65450	67713,8	73733,9
Blake2b	48257,9	97516,6	30647,2	31466,6	31035,7
Wirlpool	6258,4	10795,5	9013,888889	9255,5	10560,7

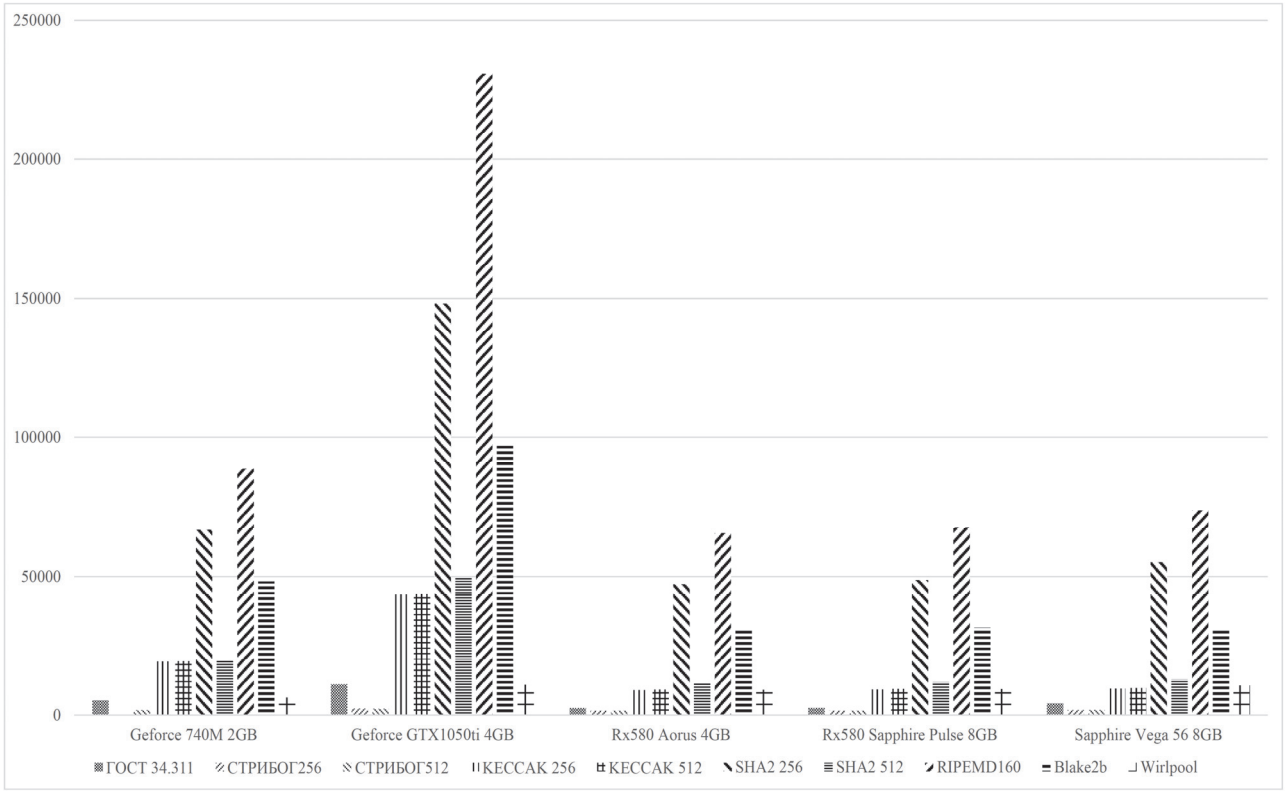


Рис. 2. Порівняння швидкодії алгоритмів ґешування на графічних обчислювачах

Таким чином, проведені дослідження показують, що різні за своєю структурою та математичними перетвореннями криптографічні функції ґешування дають різне прискорення на обчислювальних системах. Найбільш привабливими є графічні або спеціалізовані обчислювальні пристрої.

Методика та результати досліджень статистичної безпеки

Для проведення досліджень різних алгоритмів ґешування за критеріями статистичної безпеки було застосовано пакет статистичного тестування NIST STS, який рекомендований Національним інститутом стандартів і технологій США [18, 19]. Методику статистичного тестування та алгоритм обробки отриманих результатів наведено у роботах [20, 21].

Пакет статистичного тестування NIST STS був розроблений в ході проведення конкурсу AES для дослідження генераторів випадкових або псевдовипадкових чисел і є найбільш поширеним інструментом оцінки статистичної безпеки криптографічних примітивів. Використання даного пакету дозволяє оцінити, наскільки близько досліджувані криптоалгоритми апроксимують генератори «випадкових» послідовностей, тобто з високою ймовірністю стверджувати чи є генерована послідовність статистично безпечною. Порядок тестування окремої двійкової послідовності *S* має наступний вид :

- висувається нульова гіпотеза H_0 – припущення про те, що дана двійкова послідовність *S* є випадковою;
- за послідовністю *S* розраховується статистика тесту *c(S)*;
- з використанням спеціальної функції та статистики тесту розраховується значення ймовірності $P = f\left(c\left(S\right)\right)$;
- значення ймовірності *P* порівнюється з пороговим значенням $\alpha \in [0,96; 0,99]$. Якщо $P \geq \alpha$, то гіпотеза H_0 приймається. В іншому випадку приймається альтернативна гіпотеза.

Пакет містить 15 статистичних тестів, але, фактично, в залежності від вхідних параметрів обчислюються 188 значень ймовірності *P*, які можна розглядати як результат роботи окремих тестів.

1) *Частотний побітовий тест*. Спрямовано на визначення співвідношення між нулями та одиницями у двійковій послідовності певної довжини. Для дійсно випадкової бінарної послідовності кількість нулів та одиниць майже однакова. Тест оцінює, наскільки близькою є доля одиниць до 0,5.
2) *Частотний блоковий тест*. Суть тесту полягає у визначенні долі одиниць всередині блоку довжиною *m* бітів, тобто необхідно з'ясувати, чи дійсно частота повторення одиниць в блоці довжиною *m* бітів приблизно є рівною $m/2$, як можна було б припустити у випадку випадкової послідовності.
3) *Тест на послідовність однакових бітів*. У цьому тесті відбувається пошук рядків, тобто неперервних послідовностей однакових бітів. Ряд (серія) довжиною *k* бітів складається з *k* абсолютно ідентичних бітів, починається та закінчується з біту, який містить протилежне значення. В даному тесті необхідно з'ясувати, чи дійсно кількість таких рядків відповідає їх кількості у випадковій послідовності. Зокрема, визначається швидко чи повільно чергуються одиниці та нулі у початковій послідовності.
4) *Тест на найдовшу послідовність одиниць в блоці*. В даному тесті визначається найдовший рядок одиниць всередині блоку довжиною m бітів. Необхідно з'ясувати, чи дійсно довжина такого рядка відповідає очікуванням довжини найдовшого рядку одиниць у випадку абсолютно випадкової послідовності.
5) *Тест рангів бінарних матриць*. Тут здійснюється розрахунок рангів неперетинних під матриць, побудованих з початкової двійкової послідовності. Метою цього тесту є перевірка на лінійну залежність підрядків фіксованої довжини, що складають початкову послідовність.
6) *Спектральний тест*. Суть тесту полягає в оцінці висоти піків дискретного перетворення Фур'є початкової послідовності. Метою є виявлення періодичних властивостей вхідної послідовності, наприклад, близько розташованих один до одного повторюваних ділянок. Ідея полягає в тому, щоб число піків, що перевищують порогове значення у 95 % за амплітудою, було значно більшим за 5 %.
7) *Тест на співпадіння шаблонів, що не перекриваються*. В даному тесті підраховується кількість заздалегідь визначених шаблонів, які знайдені в початковій послідовності. Необхідно виявити генератори випадкових або псевдовипадкових чисел, що формують занадто часто задані неперіодичні шаблони. Як і в тесті № 8 на співпадіння шаблонів, що перекриваються, для пошуку конкретних шаблонів довжиною *m* бітів

використовується вікно також довжиною m бітів. Якщо шаблон не знайдено, вікно зсувається на один біт. Якщо ж шаблон знайдено, тоді вікно пересувається на біт, який є наступним за знайденим шаблоном, та пошук продовжується далі.

8) *Тест на співпадіння шаблонів, що перекриваються*. Суть даного тесту полягає в підрахунку кількості заздалегідь визначених шаблонів, які знайдені в початковій послідовності. Як і в тесті № 7 на співпадіння шаблонів, що не перекриваються, для пошуку конкретних шаблонів довжиною m бітів використовується вікно також довжиною m бітів. Сам пошук проводиться аналогічним способом. Якщо шаблон не знайдено, вікно зсувається на один біт. Різниця між цим тестом та тестом № 7 полягає лише в тому, що коли шаблон знайдено, вікно пересувається тільки на один біт вперед, після чого пошук продовжується далі.
9) *Універсальний статистичний тест Маурера*. Тут визначається число бітів між однаковими шаблонами в початковій послідовності (міра, що має безпосереднє відношення до довжини стиснутої послідовності). Необхідно з'ясувати, чи може дана послідовність бути значно стиснута без втрат інформації. У разі, якщо це можливо зробити, то вона не є істинно випадковою.
10) *Тест на лінійну складність*. В основі тесту лежить принцип роботи лінійного регістра зсуву зі зворотним зв'язком. Необхідно з'ясувати, чи є вхідна послідовність досить складною для того, щоб вважатися абсолютно випадковою. Абсолютно випадкові послідовності характеризуються довгими лінійними регістрами зсуву зі зворотним зв'язком. Якщо ж такий регістр занадто короткий, то передбачається, що послідовність не є в повній мірі випадковою.
11) *Тест на періодичність*. Даний тест полягає в підрахунку частоти всіх можливих перекривань шаблонів довжини m бітів протягом початкової послідовності бітів. Метою є визначення, чи дійсно кількість появ $2m$ шаблонів, що перекриваються, довжиною m бітів, є приблизно таким як і у випадку абсолютно випадковою вхідної послідовності бітів. Остання, як відомо, володіє одноманітністю, тобто кожен шаблон довжиною m біт з'являється в послідовності з однаковою ймовірністю. Варто відзначити, що при $m = 1$ тест на періодичність переходить в частотний побітовий тест (№ 1).
12) *Тест приблизної ентропії.* Як і в тесті на періодичність, в даному тесті акцент робиться на підрахунку частоти всіх можливих перекривань шаблонів довжини m бітів протягом початкової послідовності бітів. Необхідно порівняти частоти перекривання двох послідовних блоків початкової послідовності з довжинами m та $m+1$ з частотами перекривання аналогічних блоків в абсолютно випадковій послідовності.
13) *Тест кумулятивних сум*. Тест полягає в максимальному відхиленні (від нуля) при довільному обході, визначеному кумулятивною сумою заданих $(-1,+1)$ цифр в послідовності. Необхідно визначити, чи є кумулятивна сума часткових послідовностей, що виникають у вхідний послідовності, занадто великою або занадто маленькою у порівнянні з очікуваною поведінкою такої суми для абсолютно випадкової вхідної послідовності. Таким чином, кумулятивна сума може розглядатися як довільний обхід. Для випадкової послідовності відхилення від довільного обходу повинні бути близько нуля.
14) *Тест на довільні відхилення*. Суть даного тесту полягає в підрахунку числа циклів, що мають суворо k відвідувань при довільному обході кумулятивної суми. Довільний обхід кумулятивної суми починається з часткових сум після послідовності $(0,1)$, перекладеної у відповідну послідовність $(-1,+1)$. Цикл довільного обходу складається з серії кроків одиничної довжини, виконаних у випадковому порядку. Крім того, такий обхід починається і закінчується на одному і тому ж елементі. Мета даного тесту полягає у визначенні того, чи відрізняється число відвідувань певного стану всередині циклу від аналогічного числа в разі абсолютно випадкової вхідної послідовності.

Фактично даний тест є набором, що складається з восьми тестів, які проводяться для кожного з восьми станів циклу: -4, -3, -2, -1 та +1, +2, +3, +4.

15) *Інший тест на довільні відхилення*. У цьому тесті підраховується загальна кількість відвідувань певного стану при довільному обході кумулятивної суми. Метою є визначення відхилень від очікуваного числа відвідувань різних станів при довільному обході. Насправді цей тест складається з 18 тестів, що проводяться для кожного стану: -9, -8, ..., -1 та +1, +2, ..., +9.

Таким чином, в результаті тестування двійкової послідовності формується вектор $P = \{P_1, P_2 ... P_{188}\}$ значень ймовірностей. Аналіз складових P_j цього вектору дозволяють вказати на конкретні дефекти випадковості протестованої послідовності.

Проходження кожного з 15 статистичних тестів є важливим критерієм оцінки псевдовипадкового генератору. Тому навіть не відповідність за одним чи більше критеріями означає, що ключовий потік не може на високому рівні протистояти криптоаналізу. Якщо, з іншого боку, генератор проходить всі тести, це зовсім не говорить про захищеність генератору, оскільки такі тести не враховують особливостей реальної конструкції генератору.

Накопичений досвід проведення статистичного тестування показує, що кількість пройдених тестів досліджуваним генератором безпосередньо залежить від вибраної вихідної послідовності криптоалгоритму. Для забезпечення заданої достовірності результатів статистичного тестування в роботі [20, 21] запропоновано оцінити математичне сподівання числа пройдених тестів X_i досліджуваним генератором (криптоалгоритмом), розглядаючи при цьому кожне i-е тестування як одне спостереження (досвід), тобто як конкретну реалізацію деякої випадкової величини X.

При проведенні статистичних досліджень за кожним алгоритмом було сформовано 100 послідовностей завдовжки 10^8 байтів, тобто розмір статистичної вибірки за кожним алгоритмом сягав 10^{10} байтів. Кожне тестування (за кожною із 100 послідовностей) розглядалося як незалежне спостереження. У табл. 5 наведено узагальнені результати статистичного тестування за кожним дослідженим алгоритмом.

Таблиця 5

Результати статистичного тестування алгоритмів ґешування

Назва алгоритму	M099	D099	S099	P099	M096	D096	S096	P096	MIN
ГОСТ 34.311	132.90	43.93	6.62	1	186.83	1.46	1.20	1	182
СТРИБОГ256	131.92	55.93	7.47	1	186.70	1.81	1.34	1	181
СТРИБОГ512	131.64	48.99	6.99	1	186.76	1.88	1.37	1	182
BALLOON 32	134.20	60.56	7.78	1	187.10	1.09	1.04	1	185
BALLOON 64	126.50	0.25	0.50	1	183.00	1.00	1.00	1	182
BLAKE256	133.31	55.13	7.42	1	186.75	1.90	1.38	1	183
BLAKE512	132.65	55.74	7.46	1	186.73	1.59	1.26	1	183
BMW	132.39	48.99	6.99	1	186.92	1.55	1.24	1	182
CUBEHASH	131.22	55.65	7.46	1	186.80	1.44	1.2	1	183
DJB-2	8.92	1.21	1.10	1	11.64	0.29	0.53	1	10
DJB-2 XOR	2.99	2.16	1.47	1	4.94	1.69	1.30	1	0
ECHO	131.96	53.85	7.33	1	186.51	2.16	1.47	1	182
FUGUE 224	133.07	45.78	6.76	1	186.61	2.11	1.45	1	180
FUGUE 256	132.42	43.74	6.61	1	186.78	2.25	1.50	1	180
FUGUE 384	131.03	57.22	7.56	1	186.66	1.78	1.33	1	182
FUGUE 512	133.02	62.31	7.89	1	186.76	2.14	1.46	1	180
GROESTL 256	133.23	56.01	7.48	1	186.72	2.14	1.46	1	181
GROESTL 512	133.01	58.58	7.65	1	187.14	0.90	0.94	1	184
HAMSI 224	132.84	53.65	7.32	1	186.66	2.36	1.53	1	112
HAMSI 256	131.71	51.42	7.17	1	186.87	1.87	1.36	1	181
HAMSI 384	132.86	51.26	7.15	1	187.19	1.21	1.10	1	182

Назва алгоритму	M099	D099	S099	P099	M096	D096	S096	P096	MIN
HAMSI 512	132.17	51.16	7.15	1	186.45	2.68	1.63	1	179
J-H	131.70	71.63	8.46	1	186.69	2.43	1.56	1	180
KECCAK 256	131.27	58.79	7.66	1	186.52	2.70	1.64	1	181
KECCAK 512	132.40	48.12	6.93	1	186.78	1.41	1.18	1	182
LOSELOSE	16.00	0.00	0.00	1	16.00	0.00	0.00	1	16
LUFFA	133.11	47.07	6.86	1	186.71	1.50	1.22	1	183
PROGPOW	130.00	0.00	0.00	1	188.00	0.00	0.00	1	188
RANDOMX	0.00	0.00	0.00	1	0.00	0.00	0.00	1	0
RIPEMD160	84.79	84.04	9.16	1	132.11	80.95	8.99	1	105
SCRYPT 1024	133.80	68.36	8.26	1	187.10	1.49	1.22	1	184
SCRYPT 16384	140.00	0.00	0.00	1	185.00	0.00	0.00	1	185
SHA2 256	132.70	57.39	7.57	1	186.74	1.67	1.29	1	182
SHA2 512	133.01	51.78	7.19	1	186.87	1.99	1.41	1	182
SHABAL 224	132.76	50.12	7.08	1	186.67	2.52	1.58	1	180
SHABAL 256	133.18	61.72	7.85	1	186.61	2.39	1.54	1	115
SHABAL 384	131.63	45.61	6.75	1	186.54	2.00	1.41	1	180
SHABAL 512	132.81	45.41	6.73	1	186.87	1.57	1.25	1	182
SHAVITE	132.01	53.72	7.33	1	186.9	1.53	1.23	1	182
SIMD	133.09	39.54	6.28	1	186.85	1.58	1.25	1	182
SKEIN	131.90	46.97	6.85	1	186.71	1.26	1.12	1	184
WHIRLPOOL	132.29	47.90	6.92	1	186.78	1.59	1.26	1	182
X11	132.46	46.48	6.81	1	186.80	1.28	1.13	1	184
X12	133.10	21.29	4.61	1	186.20	2.36	1.53	1	183
X13	137.00	74.56	8.63	1	186.90	0.69	0.83	1	185
X14	131.40	24.44	4.94	1	186.90	0.69	0.83	1	123
X15	130.10	31.49	5.61	1	186.20	4.56	2.13	1	182
X16	0.90	0.09	0.30	1	1.00	0.00	0.00	1	1
X17	3.20	0.36	0.60	1	5.80	0.16	0.40	1	5

В табл. 5 наведено такі дані:

– «M096» та «M099» – оцінки математичного сподівання (вибіркові середні) числа пройдених статистичних тестів за критерієм $P_j \geq 0,96$ та за критерієм $P_j \geq 0,99$, відповідно;

– «D096» та «D099» («S096» та «S099») – оцінки дисперсій (середньоквадратичних відхилень) результатів тестування числа пройдених статистичних тестів за критеріями $P_j \geq 0,96$ та $P_j \geq 0,99$, відповідно;

– «P099» – значення довірчої ймовірності для числа пройдених статистичних тестів за критерієм $P_j \geq 0,99$ та при точності $\varepsilon = 2$;

– «P096» – значення довірчої ймовірності для числа пройдених статистичних тестів за критерієм $P_j \geq 0,96$ та при точності $\varepsilon = 1$;

– «Min096» мінімальні значення числа пройдених статистичних тестів за критерієм $P_j \geq 0,96$.

Результати статистичних досліджень (статистичні портрети) алгоритмів ґешування наведено на рис. 3 – 42, де по шкалі абсцис відмічений номер статистичного тесту, а за шкалою ординат – ймовірність проходження тесту.

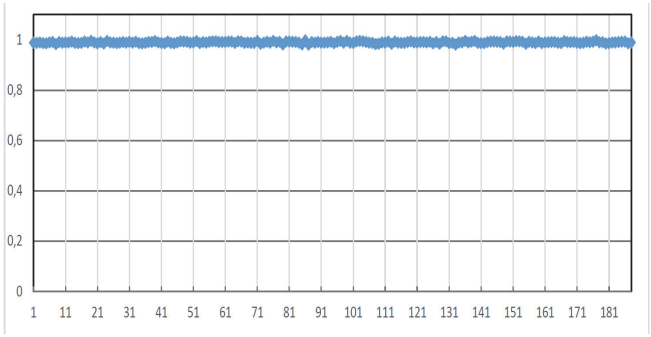


Рис. 3. BALLOON32

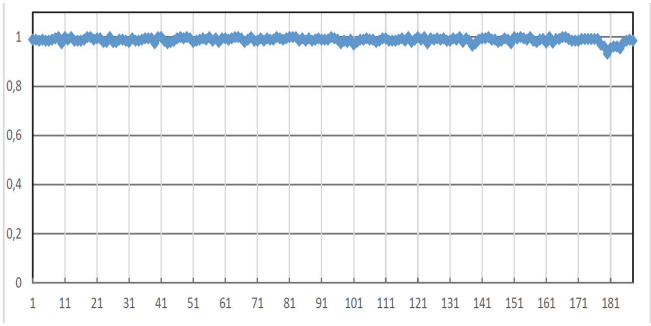


Рис. 4. BALLOON64

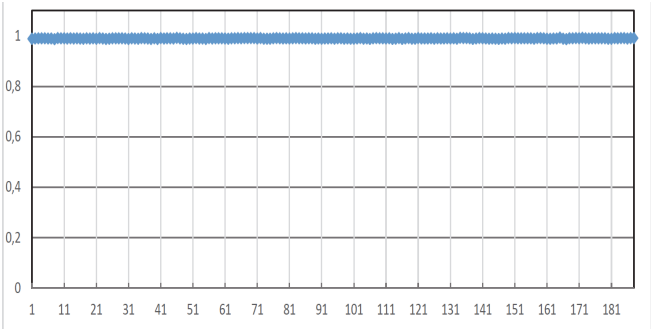


Рис. 5. BLAKE 256

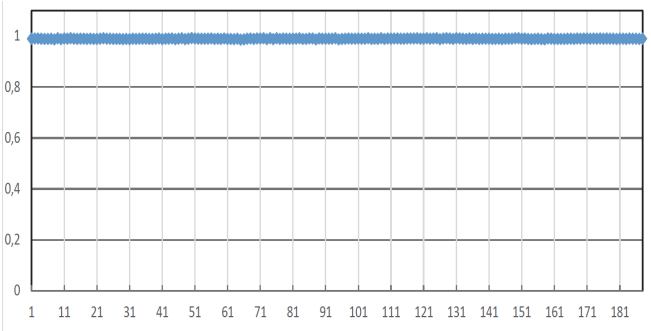


Рисунок 6. BLAKE 512

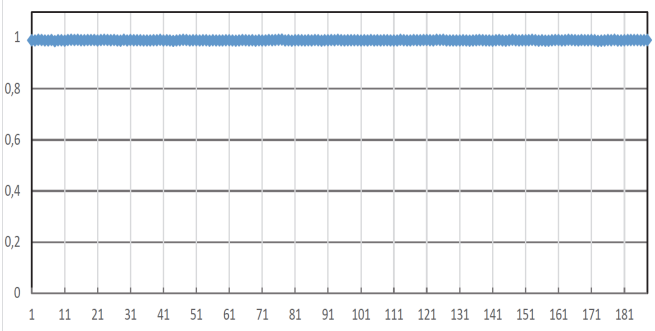


Рис. 7. BMW

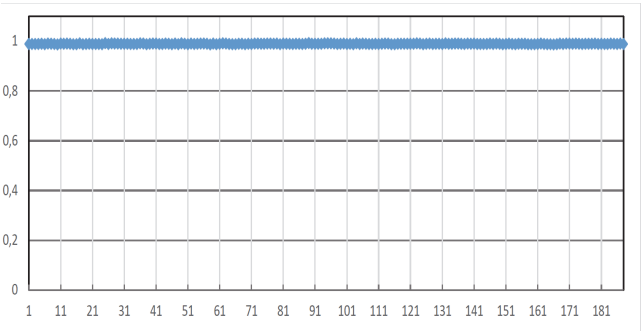


Рис. 8. CUBEHASH

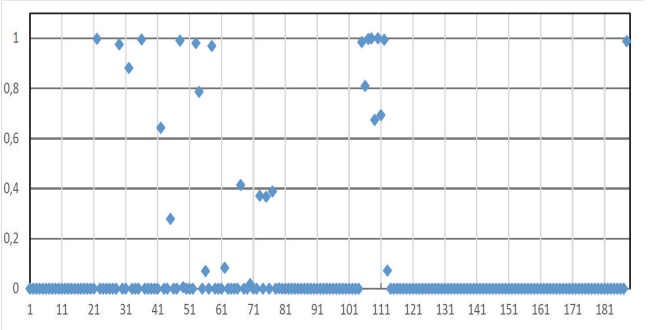


Рис. 9. DJB-2

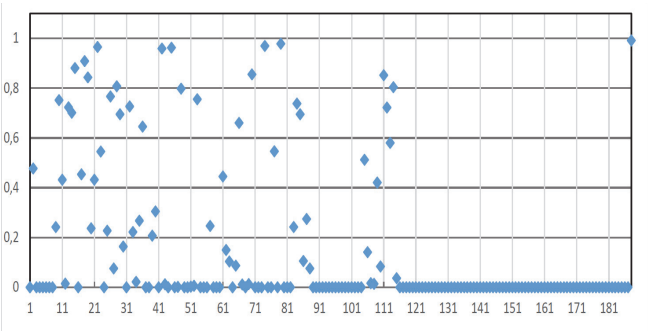


Рис. 10. DJB-2XOR

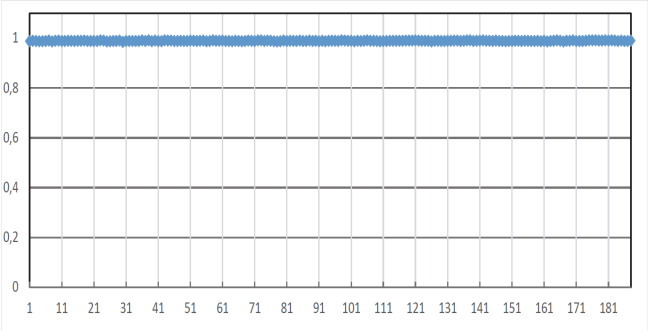


Рис. 11. ECHO

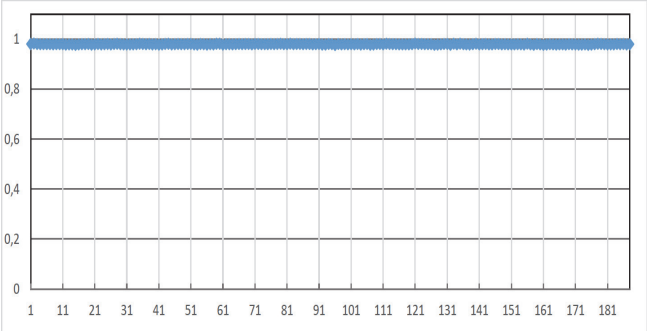


Рис. 12. KECCAK 256

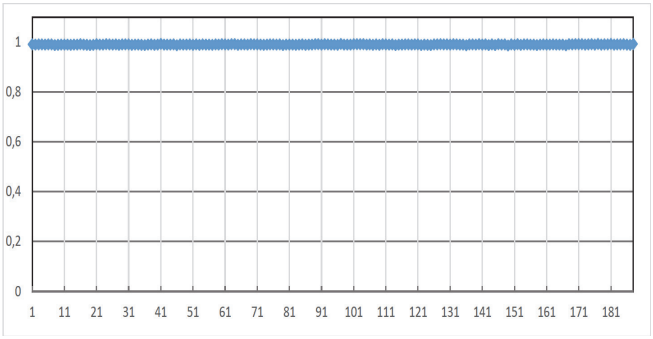


Рис. 13. KECCAK 512

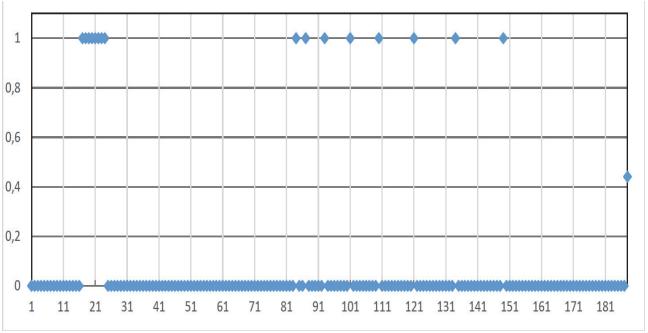


Рис. 14. LOSELOSE

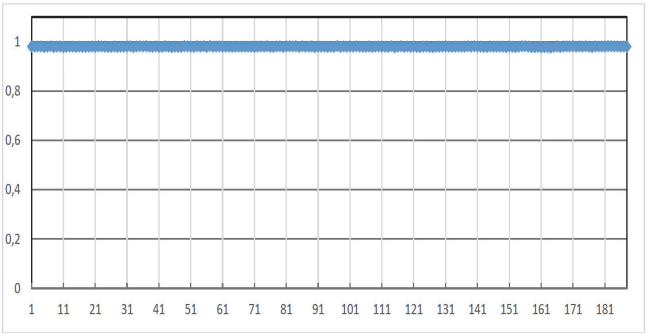


Рис. 15. LUFFA

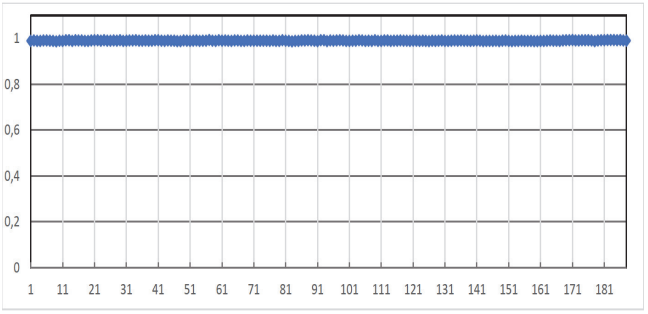


Рис. 16. FUGUE224

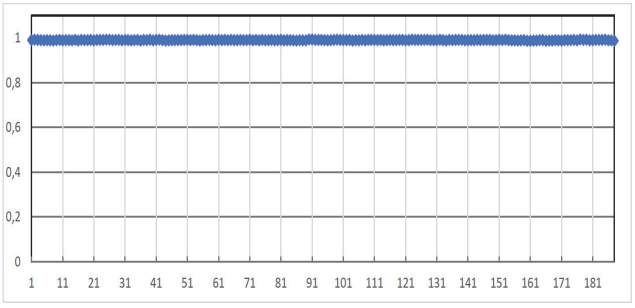


Рис. 17. FUGUE256

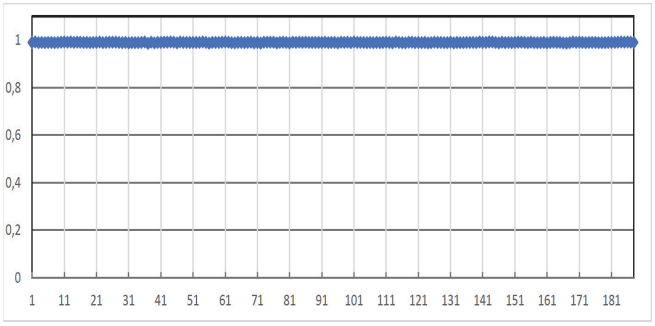


Рис. 18. FUGUE384

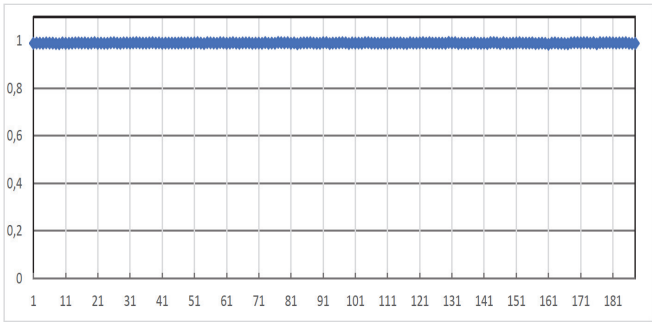


Рис. 19. FUGUE512

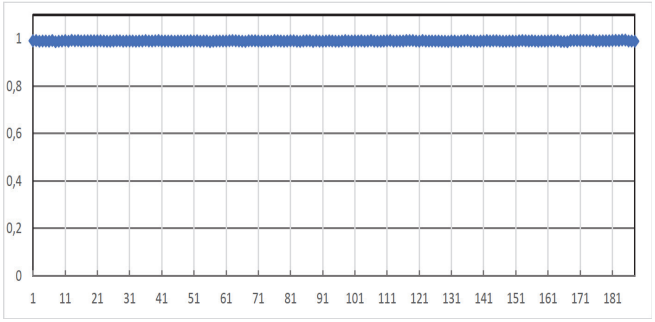


Рис. 20. GOST_256

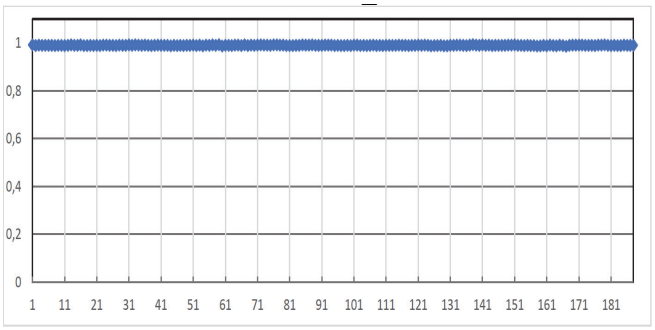


Рис. 21. Stribog_512

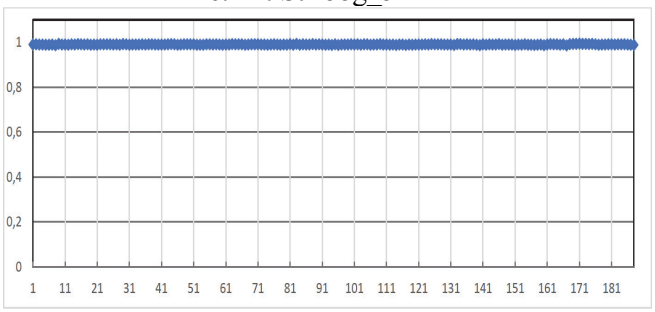


Рис. 22. WHIRLPOOL512

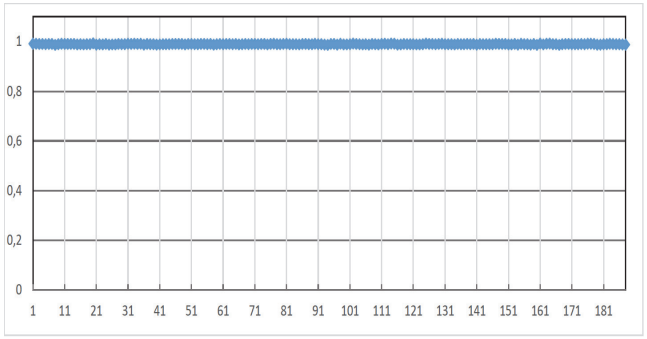


Рис. 23. GROESTL 256

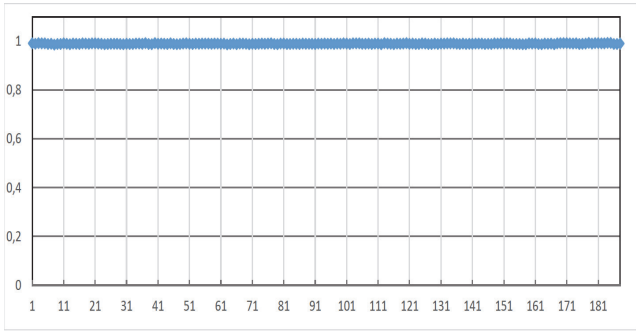


Рис. 24. GROESTL 512

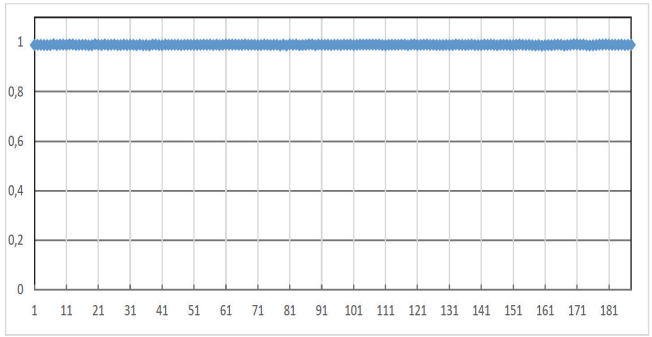


Рис. 25. HAMSI 224

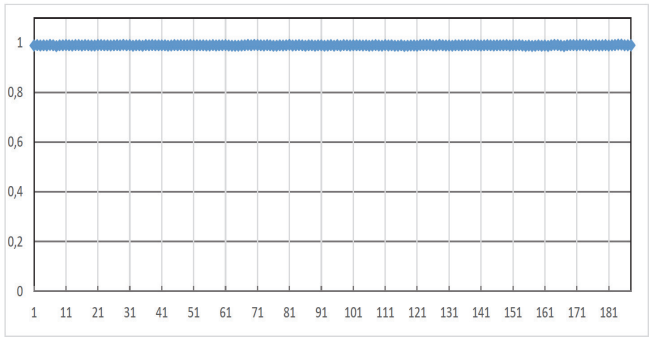


Рис. 26. – HAMSI 256

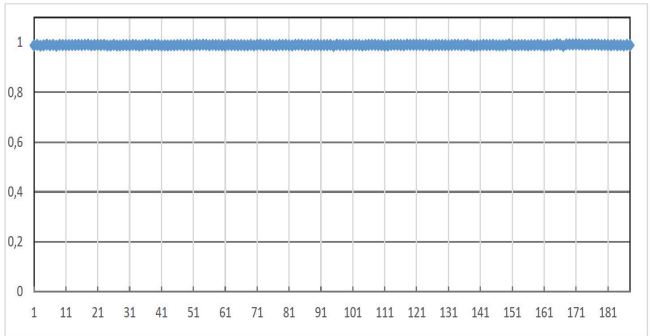


Рис. 27. HAMSI 384

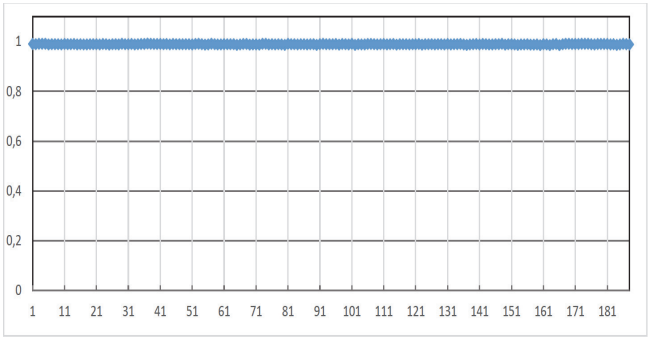


Рис. 28. HAMSI 512

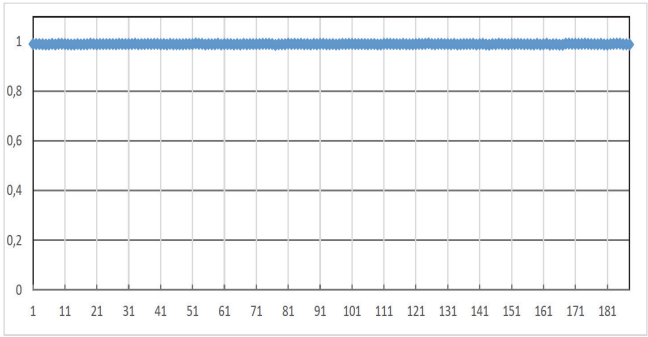


Рис. 29. – J-H

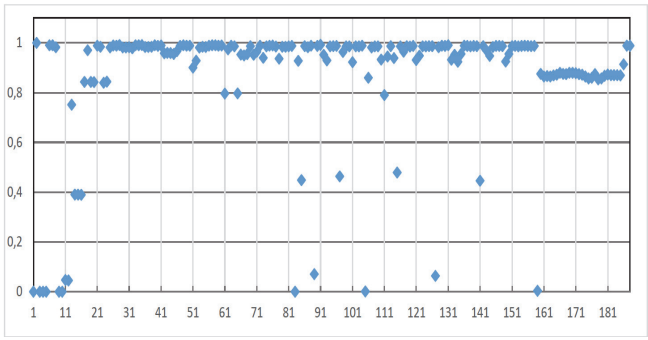


Рис. 30. RIPEMD160

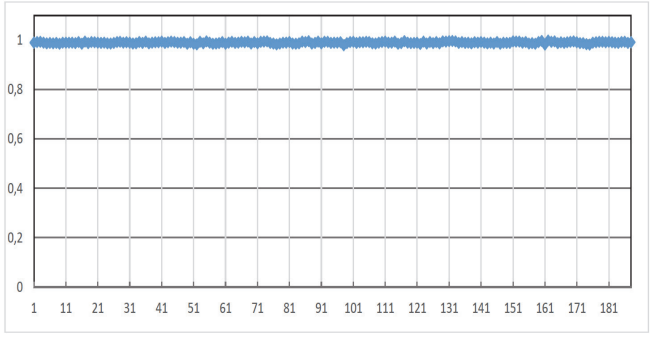


Рис. 31 – SCRYPT 1024

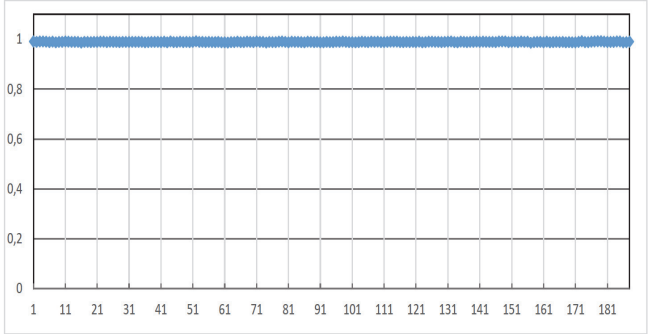


Рис. 32. SHA2 256

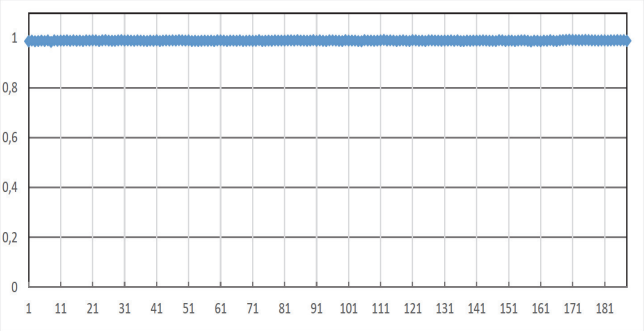


Рис. 33. – SHA2 512

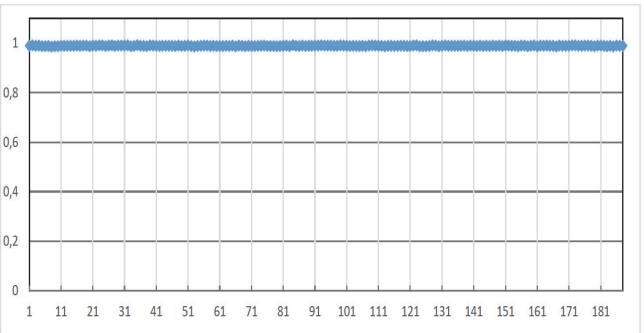


Рис. 34. SHABAL 224

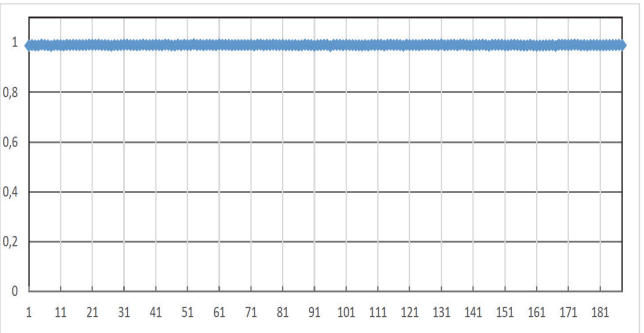


Рис. 35. SHABAL 256

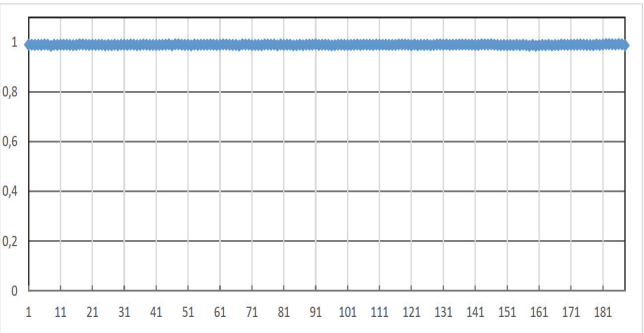


Рис. 36. SHABAL 384

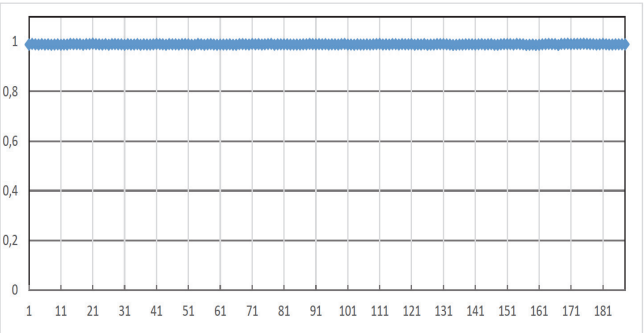


Рис. 37. SHABAL 512

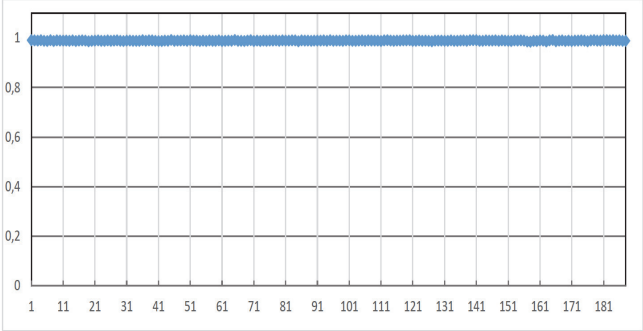


Рис. 38. SHAVITE

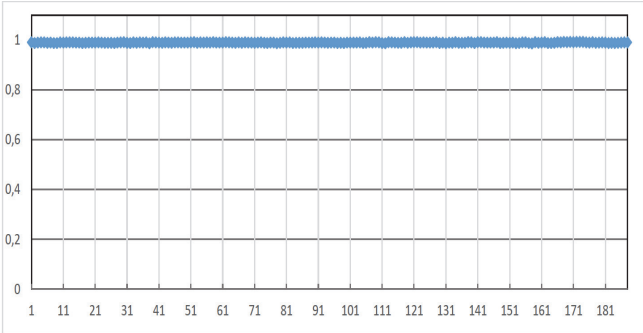


Рис. 39. SIMD

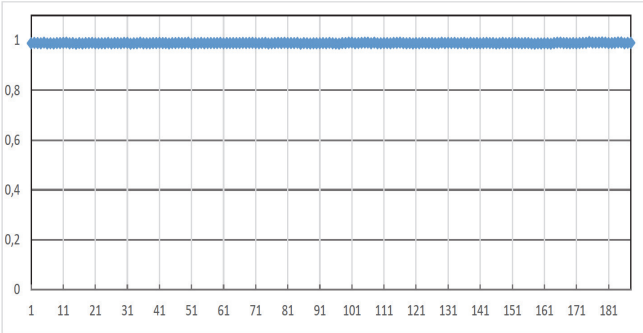


Рис. 40. SKEIN

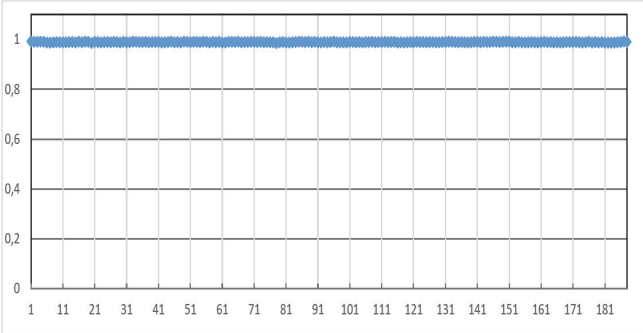


Рис. 41. STREEBOG 256

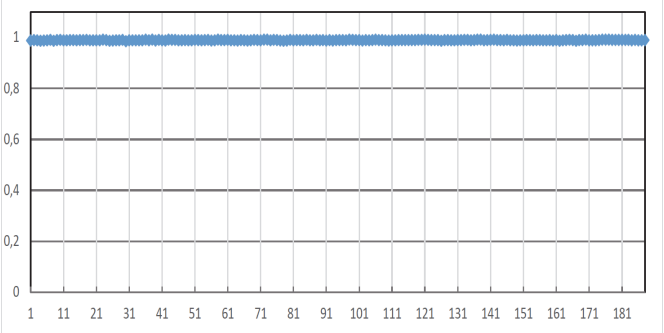


Рис. 42. X11

Отримані результати статистичних досліджень свідчать, що певні швидкісні алгоритми не можуть бути застосовані у криптографічних додатках. Це стосується, наприклад, алгоритмів DJB-2, LOSELOSE та інших, бо ці алгоритми, фактично, обчислюють не криптографічну контрольну суму. Але більшість з досліджених алгоритмів криптографічного ґешування показали високі статистичні властивості і за критерієм нерозрізнюваності із випадковою послідовністю мають високі показники.

Висновки

Функції ґешування являють собою складний і дуже важливий криптографічний примітив, який застосовується практично в усіх механізмах та протоколах криптографічного захисту інформації (формування паролів, шифрування, генерація псевдовипадкових послідовностей, формування електронного підпису, тощо). Останніми роками коло застосування ґешування значно розширилося. Зокрема, із появою та стрімким розповсюдженням децентралізованих розподілених систем, побудованих за технологією так званих «зв'язаних списків» (блокчейн-системи) виникла гостра потреба у швидких, безпечних та надійних функціях ґешування, бо саме на їх властивостях непередбачуваності та необоротності будуються захищені блокчейн-ланцюжки. Задача обрання ґеш-функції значно ускладняється через поширення спеціалізованих обчислювачів, які розробляються та практично застосовуються для пошуку прообразів наперед сформованих ґеш-значень (ASIC-майнінг). Інвестуючи в придбання ASIC-обчислювачів окремі гравці можуть бути поставлені у свідомо більш вигідне становище порівняно з іншими користувачами блокчейн-системи і, отже, можуть стати причиною недовіри та компрометації децентралізованих технологій (наприклад, різних криптовалют, розподілених сховищ, смарт-контрактів, тощо). Отже дослідження властивостей сучасних алгоритмів ґешування та обґрунтування рекомендацій щодо їх застосування для розбудови національного сегменту блокчейн-технологій є безумовно важливим та надзвичайно актуальним науковим завданням.

В цій завершальній статті (із серії робіт, присвячених алгоритмам ґешування у сучасних блокчейн-системах) проведено порівняльні дослідження як стандартизованих на міжнародному та національному рівні алгоритмів ґешування, так і ґеш-функцій, що були представлені на різних криптографічних конкурсах на науково-пошукових проектах. Зокрема в табл. 1 – 5 та на рис. 1 – 42 наведено отримані результати з дослідження таких алгоритмів: ГОСТ 34.311, СТРИБОГ256, СТРИБОГ512, BALLOON 32, BALLOON 64, BLAKE256, BLAKE512, BMW, CUBEHASH, DJB-2, DJB-2 XOR, ECHO, FUGUE 224, FUGUE 256, FUGUE 384, FUGUE 512, GROESTL 256, GROESTL 512, HAMSI 224, HAMSI 256, HAMSI 384, HAMSI 512, J-H, KECCAK 256, KECCAK 512, LOSELOSE, LUFFA, PROGPOW, RANDOMX, RIPEMD160, SCRYPT 1024, SCRYPT 16384, SHA2 256, SHA2 512, SHABAL 224, SHABAL 256, SHABAL 384, SHABAL 512, SHAVITE, SIMD, SKEIN, WHIRLPOOL, X11.

Для всіх розглянутих алгоритмів були проведені порівняльні дослідження швидкодії на різних обчислювальних платформах та із різними вхідними параметрами. Отримані результати частково наведено у табл. 1 – 4. Зокрема встановлено, що більшість проектів розподілених мереж використовує надійні та перевірені часом алгоритми криптографічного ґешування (наприклад, алгоритми KECCAK, SHA2, RIPEMD160, тощо), які володіють відносно високими швидкісними показниками. Але останніми роками для захисту від ASIC-майнерів почали застосовуватися і інші ґеш-функції, які хоча і можуть бути навіть швидшими за KECCAK, SHA2 або RIPEMD160, але володіють певними вразливостями стосовно властивостей необоротності. Як приклад можна навести застосування алгоритмів MD4, EDONR-256, EDONR-512, ED2K, або навіть найпростіших функцій DJB-2 та LOSELOSE. Через простоту обчислення певних показників не можна нехтувати порушенням властивостей необоротності, особливо якщо саме на них базуються основні переваги блокчейн-мереж. Отримані результати порівняльного аналізу дають можливість обирати криптографічні функції ґешування за критеріями швидкодії на різних пристроях та обгартовувати їх практичне застосу-

вання для побудови децентралізованих систем типу блокчейн. Це також стосується і результатів порівняльного аналізу швидкодії на графічних обчислювачів, особливо з приводу можливої розбудови національного сегменту блокчейн-мереж.

Отримані результати статистичної безпеки (див. табл. 5 та рис. 3 – 42) свідчать, що більшість функцій ґешування задовольняють встановленим критеріям (за методикою NIST STS), тобто за різними показниками вихідні послідовності (ґеш-значення) не відрізняються (у статистичному сенсі) від реалізації випадкового процесу. Це стосуються, переважно, відомих та стандартизованих алгоритмів, які застосуються в різних криптографічних додатках та вже були суттєво досліджені та вивчені при попередніх випробуваннях. Але серед алгоритмів із табл. 5 є і такі, показники статистичної безпеки яких є незадовільними, або зовсім неприйнятними. Наприклад, відомий алгоритм ґешування RIPEMD160, який стандартизовано в ISO/IEC 10118-3:2018 та прийнято до застосування в Європейському Союзі, показав невисокі значення статистичної безпеки (середнє число пройдених статистичних тестів за критерієм $P_j \geq 0,96$ не перевищує 85). Тобто, якщо на вхід алгоритму RIPEMD160 подається надмірна послідовність (в наших дослідженнях вхідна послідовність формувалася звичайним лічильником), формовані ґеш-коди за окремими тестами відрізняються від випадкової послідовності, тобто мають певний детермінізм. І хоча нами не знайдено конкретних дефектів алгоритму RIPEMD160, отримані результати свідчать про певні вади формованих ґеш-кодів з точки зору їх випадковості та непередбачуваності. Окремо слід відмітити незадовільні показники статистичної безпеки алгоритмів ґешування DJB-2, DJB-2 XOR та LOSELOSE. Ці алгоритми показали найвищі показники швидкодії (див. табл. 1, 2), але за показниками статистичної безпеки вони є неприйнятними до практичного застосування у криптографічних додатках. Цей висновок є передбачуваним, бо алгоритми DJB-2, DJB-2 XOR та LOSELOSE по суті не є криптографічними, обчислення ґеш-кодів в них є подібним до звичайної контрольної суми. Але, як показують отримані результати, навіть у разі використання статистично-небезпечних алгоритмів у складі каскадних схем майнінгу (наприклад, у складі алгоритмів ґешування сімейства «X»), формовані ґеш-значення також не задовольняють показникам статистичної безпеки (див. останні дві строки таблиці 5).

Таким чином, вибір алгоритму ґешування для побудови елементів блокчейн-систем є надзвичайно важливим і кропітким. З огляду на отримані результати окрім показників швидкодії необхідно також враховувати надійність та безпеку криптоперетворень. Важливим є також наявність спеціалізованих обчислювачів (ASIC), застосування яких значно прискорює майнінг у певних протоколах консенсусу. Отже для обґрунтування вибору алгоритмів ґешування необхідно враховувати різні фактори та показники ефективності, в тому числі особливості побудови конкретної блокчейн-системи, протоколів консенсусу, алгоритмів обробки та обміну повідомленнями, тощо.

Список літератури:

1. Bernstein hash djb2. Електронний ресурс. Режим доступу: https://riot-os.org/api/group__sys__hashes__djb2.html
2. The C Programming Language by Brian W. Kernighan (1978-02-22) Paperback, Prentice Hall, 178 p.
3. Hash Functions. Created January 04, 2017, Updated May 03, 2019. Електронний ресурс. Режим доступу: https://csrc.nist.gov/projects/hash-functions/sha-3-project
4. Classification of the SHA-3 Candidates. By Ewan Fleischmann, Christian Forler, and Michael Gorski. Version 0.90, April 19, 2009. Електронний ресурс. Режим доступу: https://eprint.iacr.org/2008/511.pdf
5. Ed2k-hash. 7 May 2005. Електронний ресурс. Режим доступу: https://wiki.anidb.info/w/Ed2k-hash
6. ed2k-tools. Tools for eDonkey2000 and Overnet. Електронний ресурс. Режим доступу:http://ed2k-tools.sourceforge.net/index.shtml
7. Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD. By Xiaoyun Wang , Dengguo Feng, Xuejia Lai3, Hongbo Yu. August 17, 2004. Електронний ресурс. Режим доступу: http://eprint.iacr.org/2004/199.pdf
8. The hash function RIPEMD-160. Електронний ресурс. Режим доступу: http://homes.esat.kuleuven.be/~bosselae/ripemd160.html

9. Secure Hash Standard. Federal Information. Processing Standards Publication 180-2. 2002 August 1. (FIPS PUB 180-2) Електронний ресурс. Режим доступу: http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf

10. Keccak hashing algorithm (SHA-3) – Keccak Coins and miner for Keccak. Електронний ресурс. Режим доступу: https://coinguides.org/keccak-algorithm-miner-coins/

11. NIST Selects Winner of Secure Hash Algorithm (SHA-3) Competition. Created October 02, 2012, Updated December 11, 2018. Електронний ресурс. Режим доступу: https://www.nist.gov/news-events/news/2012/10/nist-selects-winner-secure-hash-algorithm-sha-3-competition

12. ГОСТ Р 34.11-2012. Информационная технология. Криптографическая защита информации. Функция хэширования. Дата введения 2013-01-01. Електронний ресурс. Режим доступу: http://docs.cntd.ru/document/gost-r-34-11-2012

13. A New Standard of Ukraine: The Kupyna Hash Function. Roman Oliynykov1, Ivan Gorbenko, Oleksandr Kazymyrov,Victor Ruzhentsev, Oleksandr Kuznetsov, Yurii Gorbenko, Artem Boiko, Oleksandr Dyrda, Viktor Dolgov, Andrii Pushkaryov. Електронний ресурс. Режим доступу: https://eprint.iacr.org/2015/885.pdf

14. Argon2. By Dmitry Khovratovich. 30 March 2015. Електронний ресурс. Режим доступу: https://www.cryptolux.org/index.php/Argon2

15. Алгоритм X13 для майнинга на графических процесорах / Александр Марков. 28 мая 2018. Електронний ресурс. Режим доступу: https://miningbitcoinguide.com/mining/sposoby/x13

16. Colin Percival. Stronger key derivation via sequential memory-hard functions. 2009. Електронний ресурс. Режим доступу: https://en.bitcoinwiki.org/wiki/Scrypt http://www.tarsnap.com/scrypt/scrypt.pdf

17. Hashcat. Advanced Password Recovery. Електронний ресурс. Режим доступу: http://hashcat.net/hashcat/

18. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. – Електронний ресурс. Режим доступу: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-22r1a.pdf

19. NIST Cryptographic Toolkit. Електронний ресурс. Режим доступу: https://csrc.nist.gov/projects/random-bit-generation/documentation-and-software

20. Кузнецов А.А., Мордвинов Р.И., Колованова Е.П., Самойлова А.В. Методика статистического тестирования криптографических алгоритмов // Спеціальні телекомунікаційні системи та захист інформації. Київ, 2014. №1(25). С.54-61

21. Кузнецов О.О., Луценко М.С., Андрушкевич А.В., Мелкозерова О.М., Новикова Д.В., Лобан А.В. Статистичні дослідження сучасних потокових шифрів // Прикладная радиоэлектроника. Харьков : ХНУРЭ, 2016. Т. 15. №3. С. 167 – 178.

Харківський національний
університет імені В.Н. Каразіна;
АТ «Інститут інформаційних технологій», м.Харків

Надійшла до редколегії 05.09.2019

UDC 004.056.5 **DOI:10.30837/rt.2019.3.198.07**

KATERYNA ISIROVA, OLEKSANDR POTII, Dr. Sc.(Tecnology),
JENS CHRISTIAN CLAUSSEN, Dr. Sc.

ESTABLISHING TRUST PROTOCOLS IN MUTUAL DISTRUST NETWORK BY CONSENSUS FORMATION

Introduction

TRUST formation in computer networks by means of verification protocols has become an important subject in computer science and impacts the architecture and design of networked applications in a wide range. This issue is particularly relevant in the face of future increasing threats from quantum computing, when it will become impossible to rely only on the cryptographic strength of key system parameters.

In this paper, we draw attention to the analogy between consensus formation in social networks and trust formation in verification computer protocols. In both application domains, trust could be built within different possible topological architectures. It is interesting that consensus formation, in both disciplines, could be reached, among other topologies, both for a hierarchical and for a distributed architecture. However, the computational efficiency, in this context quantified by the time to consensus, can be quite different and depends both on the number of nodes and on the network topology.

The paper is organized as follows. In section 2 we introduce the paradigmatic voter model and present the results of simulation with two types of different architectures. In section 3 we generalize the main principles of the most widespread infrastructure and describe how it could be implemented with different architectures. Finally, in section 4 we compare and discuss the results of consensus formation time simulations, and discuss practical implications.

1. Consensus formation in social networks

1.1. The paradigmatic voter model

In this work, we are primarily concerned with the consensus formation in hierarchical versus distributed consensus protocols, highlighting the network topology dependence of the system size scaling. In complementing perspective, topology-dependence of consensus formation has been extensively studied in the context of consensus or political opinion formation both in homogeneous populations [1, 2] and in social networks [3]. Besides the formal mathematical analogy, we intend to convey the transdisciplinary viewpoint, to view (a) the protocol handshake between two computer nodes as an interaction between agents within an (artificial) society, and (b) social individuals within a population, when exchanging and agreeing on opinions, implicitely perform a formal protocol that is (observing some transmission uncertainties frameable in an information- theoretical treatment) prescribed by a logical set of social rules.

To arrive at a mathematically or computationally tractable model, it appears necessary to razor down the model to the bare necessities of the mechanism. The voter model [4] casts opinion transmission from one agent to annother into only one possible elementary interaction: in each time step, one agent is selected at random, and thereafter persuades annother agent, that is randomly chosen among the next neighbours. We observe that the interaction topology, defined by a graph adjacency matrix, will influence the dynamics of the opinion formation process.

Although it has been prominetly questioned whether the voter model itself serves as a precise model for voters [5] it is acknowledged that statistical scaling features of opinion formation processes are covered even from this simplified model.

1.2. Definition of the Voter Model

The voter model mathematically is defined as a discrete stochastic process [4]. In its standard version, the system is comprised of N nodes forming a connected graph with adjacency matrix (a_{ij}),

and the system state is defined by the state of all nodes, which can assume the values 0 or 1, respectively. These binary states can reflect opinions, an activated gene, an infection status in a contagion system, or the certification status of a node in a computer network. In the case of all nodes being in a homogeneous initial condition, i.e., all 0 or all 1, due to the absence of mutations or spontaneous opinion changes, no further change takes place, hence these states are absorbing states of the dynamics. In each step in discrete time, the following dynamics (algorithm) is executed:

1) One node is chosen at random.
2) One of the nodes j is connected to i (i.e., a_{ij} = chosen at random.
3) Node j assumes the state (opinion) of node i.

The last step can be interpreted in the way that node i convinces node j. If all nodes reach the same state, all 0, or all 1, consensus is reached, and the number of iterations is called time to consensus. It is common to perform Monte-Carlo simulations averaging over a sufficiently large number of initial conditions, to obtain reliable estimates for the expected average time to consensus. As the average time to consensus is largest when the number of 0 states and 1 states in the initial configuration equals $\lfloor N/2 \rfloor$, we have chosen such a symmetric configuration of maximal dissensus as initial configuration for all our simulations of the voter model.

1.3. Consensus Formation in the Voter Model in Different Topologies

To address systematically the average time to consensus, we have performed extensive Monte Carlo simulations of the voter models on different network architectures. These include an all-to-all coupled network (also known as complete graph in graph theory), a ring network as a one-dimensional structure with periodic boundary conditions, and specific hierarchical tree structures that resemble hierarchical both social and computer architectures.

Fig. 1 displays the average time to consensus depending on the total number of nodes N , in double logarithmic plot, for the different architectures.

2. Trust formation in computer networks

The concept of consensus formation takes place not only in the context of social networks, but also in the context of computer protocols. For almost 20 years, the society has been introducing electronic technologies into its life.

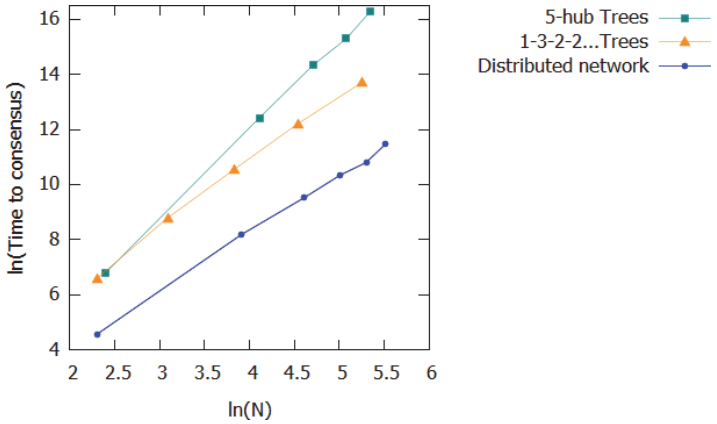


Fig. 1. Simulation results for the voter model on different network topologies. Distribution of different opinions on initial stage is 50/50. Probability of acceptance neighbor's opinion is equal to 0.5. (Results are averaged over 100 experiments)

Building trust in the online environment is a key to economic and social development. Lack of trust makes stakeholders hesitate to carry out transactions electronically and to adopt new services. The critical issue is to ensure trust in an environment of mutual distrust: a network where none of the participants trust the other. Let us mention that not only people can act as participants in the network, but also artificial intelligence agents. For instance, when building a network using the In-

ternet of Things. As we can see, successful implementation of modern technologies of electronic management, electronic trusted services are not possible without the creation of an appropriate infrastructure. The infrastructure for implementing the above mentioned technologies is the public key infrastructure (PKI).

PKI is a set of tools (technical, material, human, etc.), distributed services and components, which are collectively used to support crypto tasks based on private and public keys [8]. It does not matter whether it is a national PKI to support electronic signature tasks or a private PKI for an individual organization to support employee authentication processes, or PKI deployed on a smart home base, the principles for ensuring security remain unchanged.

In fact, PKI are based on several basic principles:

1. Private Key is known only to its owner.

2. Certification Authority (CA) creates an electronic document – a public key certificate, thus certifying the fact that the private key is known exclusively to the owner of the certificate, the public key is freely transferred in the certificate.

3. Nobody trusts each other, but everyone trusts to CA.

4. CA confirms or refutes the belonging of the public key to the given person who owns the corresponding private key.

As we can see CA acts the role of security guarantor. Although, the presence of the guarantor itself cannot ensure the security of iterations between network users. Additionally, to ensure the trust between participants reliable implementation of actual trust model should be done.

According to [7] there are verities of possible trust models:

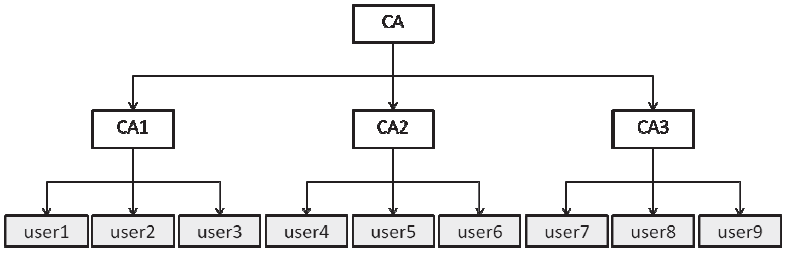


Fig. 2. Strict hierarchy of CA

- strict hierarchy of CA;
- not a strict hierarchy of CA;
- policy based hierarchy;
- distributed trust model;
- quadrilateral trust model;
- trust model around the user;
- web trust model.

In current paper we will consider two of them (a) Strict hierarchy of CA and (b) Trust model around the user, since the first one is the most widespread nowadays and the second is very convenient for developing distributed PKI.

2.1. Hierarchical PKI developing principles

The hierarchical structure is easy to imagine as a tree with a root at the top and leaves at the bottom (as shown in Fig. 2). The end point of trust is the root of the tree. The number of intermediate levels can be different, including zero. Classically, in a hierarchical architecture, security is ensured by the trust of all participants to a third trusted party, certification authorities, based on the fact that they are subject to a certification procedure.

A user requiring knowledge of a public key generally needs to obtain and validate a certificate containing the required public key. If the public key user does not already hold an assured copy of the public key of the CA that signed the certificate, the CA's name, and related information (such as

the validity period or name constraints), then it might need an additional certificate to obtain that public key. Certification paths start with a public key of a CA in a user's own domain, or with the public key of the top of a hierarchy. In both cases, it is impossible to verify the validity of the user's public key certificate without building a complete certificates chain. Such a chain links current user and a tree top. This structure allows reliably implement iterations between users, since CA is considered as trust anchor. On the other hand, that means that in the network there are a number of critical and potentially vulnerable points. Usually, these points should be subject to regular audit and strict control. In order to reduce the risk of data loss, it is necessary to store a large amount of backups, which significantly increases the cost of maintaining the system.

We can conclude that such structure has a number of other drawbacks [11], [12]:

- security of the hole system depends on CA root certificate. In case of its compromise, all certificates in the system are compromised;
- users do not actually dispose of their identity. If necessary, make any adjustments user need to contact CA;
- lack of interoperability, since certificates issued by different CA not always could be used in one system;
- there is no one-to-one correspondence between the user and the certificate, by how many certificates can be issued for one user.

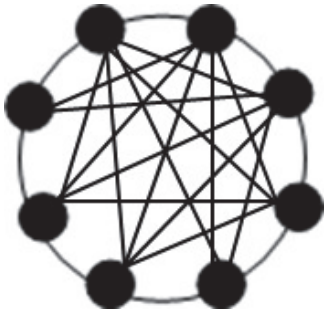


Fig. 3. Example of distributed PKI

2.2. Decentralized PKI developing principles

Another way for developing PKI without building tree structure was proposed by authors in paper [11]. The main idea was to exclude the central point of trust and ensure security using blockchain technology.

The blockchain technology [9] was introduced in 2008, its first implementation, i.e. Bitcoin, was introduced a year later, in 2009, published in the paper "Bircoin: a Peer-to-Peer Electronic Cash System" under alias Satoshi Nakamoto [10]. Since then, this technology has only gained its spread, cryptocurrencies and e-commerce market remain the main areas of application [15 – 17]. However, besides its use in electronic commerce, the blockchain technology can be implemented in other aspects. In particular, in order to avoid the disadvantages associated with the construction of a hierarchical structure. The main idea of using blockchain technology for building a decentralized PKI is that we place the register of the public key certificate status into blocks, thus, ensuring its safe storage. At the same time, the special structure of the distributed database allows users to reliably verify the certificate of the public key of another user with- out referring to a third trusted party. The main difference from the hierarchical structure is that users independently store their key pair, so that a user's public key certificate can be obtained only from himself. Special procedure was introduced only for the new user primary identification process. For this purpose, trusted nodes (analogs of CA) still exist, but their functions are sharply reduced compared with the hierarchical structure. After a new user goes through the primary identification procedure at a trusted site, he will no longer be contacted.

Such a structure can be a variation of distributed graphs (in this paper we will consider only a fully connected graph) as shown in Fig. 3.

According to [12] the main advantages of such propose are:

- considerable reduction in the cost of maintaining a cumbersome hierarchical structure of CA;
- users are able independently control their identification data and is able to immediately report about the need for their correction (compromise);
- leveling ”man in the middle” threat. The intruder will need to attack the entire system;
- the directed attack target disappeared. In contrast to hierarchical structure, when the main targets for the attackers were CA, in this case there is no clear target for the attack, because the information is stored in a distributed manner and in fact the attacker is forced to attack the whole network but not a specific node;
- the proposed system coulb be used not only for the electronic signature service, but also for ensuring the electronic identification;
- collapse of one or more nodes does not result in system shutdown;
- no need to make and store backups;
- system interoperability relies on the fact that certificates issued by various CA can easily be used in a single system;
- easy scalability, because adding a new user (a new node) occurs without changing the basic principles of the architecture.

3. The analogy between opinion formation in social network and trust formation in PKI

By consensus in PKI, we mean the state of the network, in which each node is confident in the legitimacy of all other participants. This is possible after the “network authorization” procedure. The essence of which is to initiate the interaction of each user with all others according to the laws of the certification chain in a hierarchical structure and according to the blockchain laws in a distributed one.

In this section, we present the results of simulations that were carried out using the developed software. Note that we did not integrate the digital signature algorithms directly. For the comparison procedure, it is sufficient for us to state that a uniform digital signature algorithm is applicable in all topologies.

To perform simulations regarding “network authorization” procedure on different topologies, the following initial conditions should be specified:

1. The network is given by a graph (fully connected or hierarchical one).
2. Nodes store the field ”opinion” (boolean values 1 – user is legitimate, 0 – intruder).
3. Participants do not know in advance which of the nodes are controlled by intruders. They can figure it out only in process of pairwise interaction. That means (a) passing through the certification path in the hierarchical architecture or (b) by interacting with each other participant in a distributed one.
4. We assume that at the initial stage, 50 percent of the nodes are controlled by attackers.
5. If an intruder is detected, he (and the nodes that depend on him in the tree) should be excluded from the network and consensus formation should be completed without it. In this way a network of legitimate participants will be formed and they will be able to interact seamlessly and securely.
6. Nodes that have been excluded from the network have to be regenerated (again with a probability of 50 percent) and added to the network.
7. By consensus time, we will mean the time spent on the full authorization of the network (including the time for regeneration and re-connection of nodes)

The following assumptions are necessary to describe the “network authorization” procedure:

- the considered topologies are in a closed / protected space (thus, we consider a private PKI or a private blockchain);

- the time of mutual/cross verification (verification of the public key certificate) takes one iteration;
- the node regeneration time takes two iterations (since it is necessary to regenerate the private key and create a new public key certificate).

3.1. Hierarchical network protocol

For the correct work of the protocol, prerequisite is to follow the requirements that are enshrined in X.509 [7]. Current paper considers only the protocol for a strict hierarchical structure, which means the need to build a certification path up to the root node when initializing the interaction of any two users.

”Network authorization” procedure for hierarchical net- work topology should consist of following.

1. The interaction begins with the leaves of the tree and has a direction to the root.
2. The route is determined by the rules of the certification path in accordance with X.509 [7].
3. When an intruder node is detected, it and all of its child nodes must be excluded and subjected to a regenerating procedure.
4. The interaction procedure should continue for all legitimate sites.
5. After all the legitimate nodes have interacted; the procedure for regenerating the offending nodes should be be started.

Moreover,

- if the node-intruder had no children (acted as an end user), during the procedure of its replenishment there is a probability (we will set it equal to 50 percent) that it will again appear to be intruder.
- if not (acted as a certificate authority), then it should be directly regenerated with the value “1-legitimate”, and all its child nodes, with a probability of 50 percent can again become intruders.

Corresponding simulations were performed for two hierarchical topologies types : for (a.1) tree where every node has a degree of three (Fig. 4) and (a.2) tree with five hubs (Fig. 5).

The tree with every node degree is three is a classic symmetrical PKI, which is well suited for structuring an indepth organizational structure. The limited number of child nodes does not allow the system to expand rapidly in the horizontal direction. An example of the use of such a PKI can be the union of small but clearly structured units.

From the opposite side, it is worth noting that tree with five hubs looks typical for a “wide” company PKI organization. With this topology, we have several (in this case, five) large subtrees (divisions) within which a large number of equal users can interact.

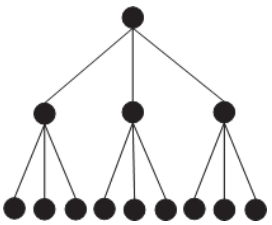


Fig. 4. Tree where every node has a degree of three

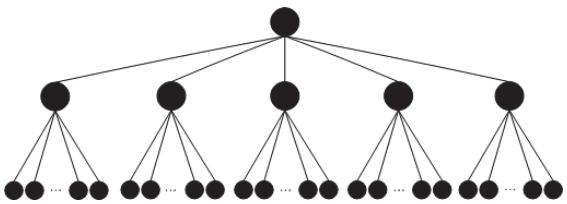


Fig. 5. Five-hubs Tree

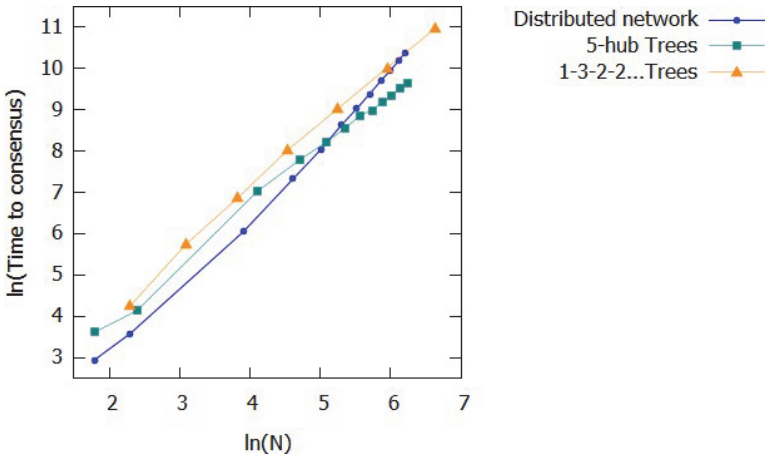


Fig. 6. Simulation results for different topology types. Distribution of legit users/intruders on initial stage is 50/50. Probability of regeneration for node is equal to 0.5. Results are averaged over 100 experiments

3.2. Distributed network protocol

”Network authorization” procedure for hierarchical net- work topology should consist of following.

1. Interaction could be started from any node in the network.
2. It is required to perform pairwise certification (ex- change key certificates) between every nodes
3. When an intruder node is detected, it must be excluded and subjected to a regenerating procedure.
4. The interaction procedure should continue for all legitimate sites.
5. After all the legitimate nodes have interacted, the procedure for regenerating the offending nodes must be launched (taking into account the probability of regeneration equal to 50 percent)

The protocol continues its operation until all nodes are legitimate and do not interrelate with each other.

3.3. Simulation results

Fig. 6 presents results of simulations.

We can conclude that for smaller networks distributed networks shows much better time to consensus and with an increasing number of nodes the advantage ceases to be so noticeable.

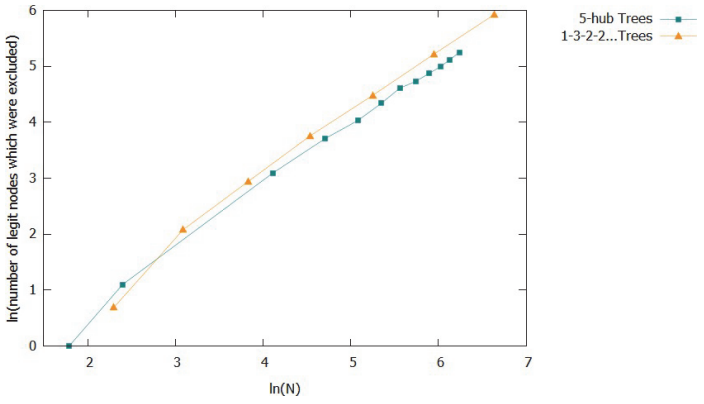


Fig. 7. Simulation results for Tree topologies. Dependence of the number of legitimate nodes that have been excluded from the size of the network. Distribution of legit users/intruders on initial stage is 50/50. Probability of regeneration for node is equal to 0.5

This tendency can be explained by the fact that to establish all links to a node in a distributed structure, it is necessary to perform a greater number of “handshakes” (cross-certification procedures) with a larger system size. At the same time for tree structures, this number grows at a slower rate.

However, another parameter should be taken into ac- count too. The number of legit users, which were excluded from the network. Obviously, such a situation is possible only in tree structures (Fig. 7), and for a distributed net- work, this value will be zero for any size of the network.

Based on the plot it is clear that with the growth of the network, the number of legitimate users who have been excluded and regenerated is growing and can reach 50 percent (and in some cases 100 percent) of the number of all users on the network. This is easy to imagine if the root node of the tree system is compromised. This situation causes network redundancy and leads to inefficient use of resources.

Discussion and conclusions

Nowadays progress in the field of electronic technologies allows providing more efficient electronic trusted services. Building trust is an important task for the reliable critical infrastructures functioning at any level. Moreover, to conduct completely secure interactions, not only the trust among the users (human or artificial intelligence) is required, but also confidence in the technology itself. Distributed trust models and public key infrastructures will become information technologies of increasing importance, especially in the context of information security when quantum computing technologies become available. Industries, trafic and trade networks as well as the public services sector are increasingly relying on secure networks [13], [14], which often operate internationally and on scales of large number of nodes. This imposes challenges on the architecture of the verification protocols, such that these can be managed with computational and network resources scaling with system size in a feasible way. Besides optimizing the protocols themselves, the network topology of each verification concept will have a significant impact on its efficiency. PKI looks like a very promising instrument to ensure trust in a mutual distrust network, however, zero-day threats, such as the active development of quantum computing, should be taken into account. And if we do so, then it should be understood that the security of the whole system cannot rely only on the key parameters cryptographic security, but additional measures are also required to ensure the resilience of the system. Such measures can be distributed technologies, in particular, those based on the blockchain technology. Here, we have investigated distributed PKI in comparison with hierarchical CA architecture. The time (in units of protocol verification steps) for verification of all nodes in the whole network has, as we argue, an important analogy to the time to consensus formation in a social network. Consequently, we investigate this time to consensus on different topologies, both for computer verification protocols, and for consensus formation in social networks as described by the well-established voter model. We have performed Monte-Carlo simulations for all-to-all coupled networks, or complete graphs, resembling a fully distributed PKI, or a maximally connected social network in which information may reach out to any node. This is not unrealistic in some social subnetworks, like a school class, or the scientific com- munity where all university lecturers and active researchers can be reached electronically through their departmental email documented in the web. For representative hierarchical architectures, we have focused on a tree where all top and middle layer nodes have a node degree of three, and a hub-tree akin of a company with 5 departments and flat hierarchy therein. These two trees represent a many- layer structure and a few-layer structure, respectively. We find that, in the social opinion formation by the voter model, the distributed network, by large margin, provides fastest consensus. This result holds over the whole range of investigated network sizes. While both tree architectures perform similarly on small networks, for large networks the 5-hub trees converge slower, even with a different scaling, to consensus. For medium-size networks, the results for the verification protocols are similar, although the 5-hub trees outperform the many-layer trees for medium and large net- works. Counterintuitively, the largest networks investigated (around $N = 200$) show a decaying performance for the distributed networks.

This may be related to our assumption, originally motivated from the opinion formation analogy, of 50 of 100 nodes being not trusted, which puts the system in a dynamical regime where recovery to a fully certified net- work may take long. Further, our investigation confirms our hypothesis that the network topology has a significant role in the time to consensus. However, the network architecture which is optimal may still depend on the type of protocol (where we made some generic assumptions), and on the system size. This aspect, together with addressing resilience to targetet attacks, should be subject of further investigation.

Acknowledgments

Kateryna Isirova acknowledges funding through the Erasmus programme during her research visit to Aston University, Birmingham (UK).

References:

1. Jan Lorenz. Continuous Opinion Dynamics Under Bounded Confidence: a Survey // J. Mod. Phys. C 18, 1819-1838 (2007).

2. Claudio Castellano, Santo Fortunato and Vittorio Loreto. Statistical physics of social dynamics // Rev. Mod. Phys. 81, 591 (2009).

3. Petter Holme and M. E. J. Newman. Nonequilibrium phase transition in the coevolution of networks and opinions // Phys. Rev. E 74, 056108 (2006).

4. Thomas M. Liggett. Interacting Particle Systems. Springer, Berlin (2012).

5. Juan Fern a´ndez-Gracia, Krzysztof Suchecki, Jose´ J. Ramasco, Maxi San Miguel, and V´ıctor M. Egu´ıluz, Is the Voter Model a Model for Voters? // JPhys. Rev. Lett. 112, 158701 (2014).

6. Quantum Safe Cryptography and Security; An introduction, benefits, enablers and challenges. Quantum Safe Cryptography // ETSI White paper, 2015

7. ISO / IEC 9594-8 ITU – T Rec. The X.509 ”The Basic s e PROVISIONS certification key and certificate attributes”.

8. PKI: technology, architecture, construction and implementation: a tutorial / Potiy A.V., Lenshin A.V., Soroka L.S., Esin V.I., Moroz B.I. Dnepropetrovsk : Akadimiya border service of Ukraine 2011. 202 pp.

9. Draft NISTIR 8202 : Blockchain Technology Overview.

10. Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. Technical report.

11. Isirova Kateryna. Decentralized Public Key Infrastructure Development Principles / Kateryna Isirova, Oleksandr Potii // The 9th IEEE International Conference on Dependable Systems, Services and Technologies, DESSERT’2018, 24-27 May, 2018, Kyiv, Ukraine. P. 320-326.

12. K. Isirova, Blockchain technology as the perspective instrument for ensuring electronic trusted services in conditions of cyberthreats // European Cybersecurity Journal. Volume 5 (2019), Issue 1, pp. 34-42.

13. Stefan L a¨mmer, Hiroshi Kori, Karsten Peters and Dirk Helbing. Decentralised control of material or traffic flows in networks using phase-synchronisation // Physica A 363, 39-47 (2006).

14. Laura Alessandretti, Abeer ElBahrawy, Luca Maria Aiello and Andrea Baronchelli. Anticipating Cryptocurrency Prices Using Machine Learning // Complexity 2018, 8983590 (2016).

15. Deepak Puthal, Nisha Malik, Saraju P. Mohanty, Elias Kougianos and Chi Yang. The Blockchain as a Decentralized Security Framework // IEEE Consumer Electronics Magazine 7, 18 (2018).

16. Zibin Zheng, Shaoan Xie, Hongning Dai, Xiangping Chen and Huaimin Wang. An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends // 2017 IEEE 6th International Congress on Big Data, DOI: 10.1109/BigDataCongress.2017.85.

17. Francesco Parino, Mariano G. Beir o´ and Laetitia Gauvin. Analysis of the Bitcoin blockchain: socio-economic factors behind the adoption // EPJ Data Science 7, 38 (2018).

Kharkiv National V.N. Karazin University;
JSC "Institute of Information Technologies";
Mathematics EAS, Aston University, United Kingdom

Received 11.09.2019

UDC 004.772.2 **DOI:10.30837/rt.2019.3.198.08**

M. OSADCHUK, R. OLIYNYKOV, Dr. Sc. (Tecnology)

METHOD OF PROOF OF WORK CONSENSUS ALGORITHMS COMPARISON

Introduction

After the breakthrough paper proposed by Satoshi Nakamoto, blockchain systems have received wide distribution. The blockchain technology allows to record and secure store big amount of various transactions. The most popular usage the technology received in digital currencies. Also it is used in the areas of government, agriculture, environment, healthcare, education, and much more. Moreover it is being implemented in telecommunications area for dealing with fraud, quickly resolving disputes over roaming agreements, verification of billing and user identification (e.g. ENCRY [1]).

The decentralization has shown the new concept of trust – without any third party. So there is no central authority that can set the rules and there is no single point of failure.

There can be two types of blockchain consensus protocols – permissioned [2] and permissionless [3]. The permissioned one requires permission to read the information stored in the blockchain. The system based on permissionless protocol is publicly available, so everyone can join and participate in consensus.

The consensus algorithm is the most important part of the blockchain system. It is responsible for the way how users will come to the consensus among all honest participants. Firstly, they are used in fault tolerance real-time systems, i.e. nuclear power stations, space systems, aviation systems, and other critical systems. Secondly, they are used in fault tolerance counting systems, like clusters and database controllers. Thirdly, consensus algorithms are used in digital currencies to form the transactions base. The last one can be used either in cryptocurrencies (Bitcoin [4], Cardano [5], Ethereum [6], etc.) or in centralized currencies (e.g. Ripple [7]).

For today there are three widely spread variants of reaching consensus, i.e. Proof of Work [8], Proof of Stake [9] and Byzantine Fault Tolerance [10]. There are also a lot of algorithms derived from them, so the amount of consensus algorithms is big enough to think about "How to choose the most suitable for defined criteria algorithm among all existing?"

In that case, the comparison of algorithms should be conducted. There are publications with analysis of typical consensus algorithms and some of their contemporaries. Du Mingxiao, Ma Xiaofeng, Zhang Zhe in the paper "A review on consensus algorithm of blockchain" [11] have performed a deep analysis of consensus algorithms, their benefits and disadvantages. Zibin Zheng, Shaoan Xie and Hongning Dai have researched the scalability and security problems [12]. L.M. Bach, B. Mihaljevic and M. Zagar perform a comparative analysis of typical consensus algorithms and some of their contemporaries that are currently in use in modern blockchains [13]. They perform the overview of algorithms and their analysis focuses on the algorithmic steps taken by each consensus algorithm, the scalability of the algorithm, the method the algorithm rewards validators for their time spent verifying blocks, and the security risks present within the algorithm.

This articles provides an overview of advantages and disadvantages of consensus algorithms and then one can choose the optimal solution for their system to be developed using the proposed method to make a decision on the most suitable algorithm for given system requirements.

The method proposed in this article allows making a decision on which of algorithms will match better with known conditions and requirements to the system. This method is proposed for PoW consensus, but can be also used for other types of consensus algorithms. The method uses the weight or, in another words, the priority of algorithms properties, so the party decides which of properties have the highest priority for their system. Thus, the proposed method can be used to choose among the variety of consensus algorithms the best one, based on the priority of algorithm's properties.

1. Main features of consensus algorithm

The consensus algorithm has the following main properties. First of all it is the type of consensus mechanism. It can be Proof of Work, Proof of Stake, delegated Proof of Stake, Practical Byzantine Fault Tolerance, etc. Among all algorithms were chosen the most popular for permissionless systems Proof of Work algorithm and its variants of improvement. In this article there are compared the next algorithms: Proof of Work, delayed Proof of Work, Proof of Activity, Proof of Burn and Proof of Capacity.

The Proof of Work involves scanning for a value that when hashed the hash begins with a number of zero bits [8]. They are used to provide security to an entire network and ensure that all transactions will be processed in a timely manner [14].

Delayed Proof of Work leverages the hashrate of the Proof of Work network to protect its network [15]. Because of this process, which is called notarization, delayed Proof of Work provides the higher level of security.

Proof of Activity is a hybrid of Proof of Work and Proof of Stake. This feature provides high level of scalability [16].

Proof of Burn is a consensus algorithm, where users send their money to one defined address, or in other words burn their coins [17]. This algorithm is more environment-friendly and do not need any additional tools to increase the mining power.

Proof of Capacity is very similar to the Proof of Burn, but instead of using counting power, it uses memory [18]. That's why users do not need to have additional tools, and after they stop mining, they can use their memory for themselves.

The second feature is the transaction confirmation latency (i.e. number of blocks needed to secure accept the transaction). In other words, it is the minimum amount of blocks that are needed to ensure security against double-spending attack.

The third feature is the attack prevention mechanism. This feature responds for the way how algorithm prevents the system from the double-spending attack. It can be based on the hashrate of the system – the bigger the system is, the harder to perform the attack. Also it can be some additional features in the algorithm (e.g. PirlGuard System [19] and ChainLock Mechanism [20]).

The next feature of consensus algorithms is their scalability – the way how an algorithm works in case of increasing number of users reaching consensus. By this feature the algorithms can be divided to easy scalable, scalable with additional conditions and not scalable. This feature is important especially for digital currencies, as the number of its users can increase very fast.

The last feature considered is the level of decentralization. Algorithms can be fully decentralized, partially decentralized and centralized.

2. The decision-making method in uncertainty conditions

Decision making [21] can be described as the process of reducing uncertainty about solution options by gaining sufficient knowledge of the options to allow a reasonable selection from among them. In this context certainty does not mean an exact knowledge of every detail relevant to the problem under consideration, but it does mean that there is a reasonably good idea of the value of all relevant factors.

To choose the most suitable consensus algorithm for the system there is applied a number of criteria, so the multi criteria decision making methods (MCDM) should be used. There can be Multi-Attribute Utility Theory (MAUT), Analytic Hierarchy Process (AHP), Case-Based Reasoning (CBR), Data Envelopment Analysis (DEA), Fuzzy Set Theory, Simple Multi-Attribute Rating Technique (SMART), Goal Programming (GP), ELECTRE, PROMETHEE and etc. The AHP method is easy to use and scalable. Also its area of application is performance-type problems, strategy development and planning [22].

Among all MCDMs it was chosen the method based on AHP proposed by Thomas L. Saaty [23]. The AHP helps the parties find one decision that best matches their goal and their understand-

ing of the problem. This process provides a holistic, rational and comprehensive decision for representing the problem, its elements and evaluating alternative solutions.

To use the AHP, the party represents their problem in the hierarchy, where the top reflects to the goal, the interim levels reflects to the technical-economic parameters, and the bottom level reflects to the set of alternatives.

The AHP hierarchy is a structured representation of the decision. The technical-economic parameter, or criteria, can be divided into the subcriteria, sub-subcriteria in as many levels as the problem requires. Also the type of the hierarchy depends on the knowledge, opinions, values and needs of the parties.

After representing the problem into the hierarchy, the properties priority is set and each alternative is estimated with each property. In the AHP the elements are pairwise compared in relation to their impact to their common property. This system of pairwise comparison can be represented into the inversely symmetric matrix. The element $a(i,j)$ of the matrix relates to the intensity of occurrence of element i regarding the element of hierarchy j. This intensity is evaluated from 1 to 9, where:

1 – the equal importance
3 – the medium leverage
5 – the supreme leverage
7 – the significant leverage
9 – the large leverage
2, 4, 6, 8 – the relative interim value

The comparisons and evaluation relies on the judgements made by experts and represent how much more, one element dominates another with respect to a given attribute.

3. The method of consensus algorithms comparison

First of all, to use these comparison methods the criteria should be defined. The set of criteria may be different for each distributed system. The following criteria help estimate and chose the consensus algorithm that will fully match the existing requirements. All requirements can be divided into required, desired and additional. The algorithm matches better, when it meets the required requirements, despite the number of desired requirements met.

In this article the requirement to the distributed system is to perform the most securely and fully decentralized system. To perform the holistic assessment of each consensus algorithm the following criteria are proposed.

- Amount of blocks for transaction confirmation – this criterion performs the amount of blocks that must be created after the block in which transaction is included, for its confirming.
- Difficulty of attack performing – this criterion performs the hashpower (or another parameter depends on the algorithm) needed for malicious user to get the control on the system.
- Scalability – this criterion shows the work principles in increasing amount of users.
- Decentralization degree – this criterion performs how much decentralized is an algorithm.
- Smart-contracts support – this criterion performs the ability of smart-contract creation and the simplicity of its usage for this consensus algorithm.

After the criteria are set, their priorities should be defined. To assess the priority of each criterion the matrix with pairwise comparison should be created. This matrix represents the result of each comparison. As mentioned above, the estimations are based on experts' decisions. In this article all estimations were made by experts from Information Systems and Technologies Security department at V.N. Karazin Kharkiv National University [24] and from the Distributed Lab Company [25].

After comparing all criteria the normalized value for each criterion should be computed. This value describes the priority weight for each criterion. In the table 1 it is given the representation of the pairwise comparison of aforementioned criteria.

Table 1

The numeric estimations of pairwise comparisons

Pairwise comparison of consensus protocol properties	Amount of blocks for transaction confirmation	Difficulty of attack performing	Scalability	Decentralization degree	Smart-contracts support	Summary	Normalized value
Amount of blocks for transaction confirmation	1	1/5	1/3	1/7	2	3.67619	0.08813
Difficulty of attack performing	5	1	2	1/3	2	10.33333	0.24775
Scalability	3	1/2	1	1/5	2	6.7	0.16063
Decentralization degree	7	3	5	1	2	18	0.43156
Smart-contracts support	1/2	1/2	1/2	1/2	1	3	0.07193
Summary	-					41.70952	1

The next step is the pairwise comparison of defined algorithms. The comparison should be performed with each criterion, i.e. the number of matrices with algorithms comparison is the same as the number of criteria. Also the normalized value should be counted for each algorithm.

Below are presented the tables 2 - 6 with pairwise comparison of consensus algorithms.

Table 2

The pairwise comparison of consensus algorithms with criteria “Amount of blocks for transaction confirmation”

Consensus protocol	Nakamoto	dPoW	PoA	PoB	PoC	Summary	Normalized value
Nakamoto	1	1/9	1/7	1/9	1/7	0.50794	0.01006
dPoW	9	1	2	1	2	15	0.29698
PoA	7	1/2	1	1/2	1	10	0.19799
PoB	9	1	2	1	1/2	13.5	0.26728
PoC	7	1/2	1	2	1	11.5	0.22768
Summary	-					50.50794	1

Table 3

The pairwise comparison of consensus algorithms with criteria “Difficulty of attack performing”

Consensus protocol	Nakamoto	dPoW	PoA	PoB	PoC	Summary	Normalized value
Nakamoto	1	1/3	1/5	1	1	3.53333	0.09037
dPoW	3	1	1/2	3	3	10.5	0.26854
PoA	5	2	1	5	5	18	0.46036
PoB	1	1/3	1/5	1	1	3.53333	0.09037
PoC	1	1/3	1/5	1	1	3.53333	0.09037
Summary	-					39.09999	1

Table 4

The pairwise comparison of consensus algorithms with criteria “Scalability”

Consensus protocol	Nakamoto	dPoW	PoA	PoB	PoC	Summary	Normalized value
Nakamoto	1	1/9	1/5	1/5	1	2.51111	0.0504
dPoW	9	1	2	2	9	23	0.46164
PoA	5	1/2	1	1	5	12.5	0.25089
PoB	5	1/2	1	1	1	8.5	0.1706
PoC	1	1/9	1/5	1	1	3.31111	0.06646
Summary	-					49.82222	1

Table 5

The pairwise comparison of consensus algorithms with criteria “Decentralization degree”

Consensus protocol	Nakamoto	dPoW	PoA	PoB	PoC	Summary	Normalized value
Nakamoto	1	1/9	1/5	1/5	1	2.51111	0.04736
dPoW	9	1	2	2	9	23	0.43378
PoA	5	1/2	1	1	5	12.5	0.23575
PoB	5	1/2	1	1	5	12.5	0.23575
PoC	1	1/9	1/5	1/5	1	2.51111	0.04736
Summary	-					53.02222	1

Table 6

The pairwise comparison of consensus algorithms with criteria “Smart-contracts support”

Consensus protocol	Nakamoto	dPoW	PoA	PoB	PoC	Summary	Normalized value
Nakamoto	1	1/3	1/5	1	1	3.53333	0.09037
dPoW	3	1	1/2	3	3	10.5	0.26854
PoA	5	2	1	5	5	18	0.46036
PoB	1	1/3	1/5	1	1	3.53333	0.09037
PoC	1	1/3	1/5	1	1	3.53333	0.09037
Summary	-					39.09999	1

The last step of the method is making the decision. For this, the final value should be counted including all normalized values received from pairwise algorithms comparisons and the priority weight for each criterion. The table 7 represents the results of this summarizing.

Table 7

The results of calculations

Protocol properties / Consensus protocols	Amount of blocks for transaction confirmation	Difficulty of attack performing	Scalability	Decentralization degree	Smart-contracts support	Summary
	Weight of criteria					
	0.08813	0.24775	0.16063	0.43156	0.07193	1
Nakamoto	0.01006	0.09037	0.0504	0.04736	0.09037	0.05831
dPoW	0.29698	0.26854	0.46164	0.43378	0.26854	0.37338
PoA	0.19799	0.46036	0.25089	0.23575	0.46036	0.30666
PoB	0.26728	0.09037	0.1706	0.23575	0.09037	0.18159
PoC	0.22768	0.09037	0.06646	0.04736	0.09037	0.08006

The results of method usage shows that among defined consensus algorithms and defined requirements for the system the delayed Proof of Work algorithm is the most suitable that is indicated by its value 0,37338 that is the highest among all other.

Conclusions

In blockchain-system development the key role belongs to a consensus algorithm. It is a huge variety of consensus algorithms proposed in the literature for solving various amount of different tasks, but there is no algorithm how to compare them and choose the one for very this system.

For making a decision in uncertainty conditions with multi criteria there exist methods like Multi-Attribute Utility Theory (MAUT), Analytic Hierarchy Process (AHP), Case-Based Reasoning (CBR), Data Envelopment Analysis (DEA), Fuzzy Set Theory, Simple Multi-Attribute Rating Technique (SMART), Goal Programming (GP), ELECTRE, PROMETHEE and etc. It was decided to use the method by T. Saati because it is oriented to strategy and planning and it is scalable.

In this method the analytic hierarchy process is used. It represents the problem in the hierarchy, where the top reflects to the goal, the interim levels reflect to the technical-economic parameters, and the bottom level reflects to the set of alternatives.

Basing on the Saati's method, the method of consensus algorithms comparison was proposed. There were defined 5 criteria: the amount of blocks for secure transaction confirmation, difficulty of attack performing, scalability, decentralization degree and the possibility of smart-contracts. The priority of each criterion was calculated via pairwise comparison. After performing the pairwise comparison of consensus algorithms it was defined that the algorithm delayed Proof of Work gets higher value and has best correspondence to given criteria than the other consensus algorithms researched in this article.

References:

1. Electronic resource: https://encry.com/.
2. Novotny P., Qi Zhang, Hull R., Baset S., Laredo J., Vaculin R., Ford D. L., Dillenberger D. N. Permissioned Blockchain Technologies for Academic Publishing // https://arxiv.org/ftp/arxiv/papers/1809 /1809.08529.pdf.
3. Chunpeng Ge, Siwei Sun, Szalachowski P. Permissionless Blockchains and Secure Logging // https://arxiv.org/abs/1903.03954.
4. Electronic resource: https://bitcoin.org.
5. Electronic resource: https://www.cardano.org.
6. Electronic resource: https://www.ethereum.org.
7. Electronic resource: https://www.ripple.com.
8. Satoshi Nakamoto Bitcoin: A Peer-to-Peer Electronic Cash System // https://bitcoin.org/bitcoin.pdf.
9. Ganesh Ch., Orlandi C., Tschudi D. Proof-of-Stake Protocols for Privacy-Aware Blockchains // Cryptology ePrint Archive: https://eprint.iacr.org/2018/1105.pdf.
10. Lamport L., Shostak R., Pease M. The Byzantine Generals Problem // https://pdfs.semanticscholar.org/1689/f401f9cd18c8fd033d99d1e2ce99b71e6047.pdf.
11. Du Minigxiao, Ma Xiofeng, Zhanh Zhe, Wang Xiangwei, Chen Qijun A review on consensus algorithm of blockchain // https://ieeexplore.ieee.org/abstract/document/8123011.
12. Zibin Zheng, Shaoan Xie, Hongning Dai, Xiangping Chen, Huaimin Wang An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends // https://ieeexplore.ieee.org/abstract/document/8029379.
13. Bach L. M., Mihaljevic B., Zagar M. Comparative analysis of blockchain consensus algorithms // https://ieeexplore.ieee.org/abstract/document/8400278.
14. Proof of Work: A History & Overview of Proof of Work Systems // https://komodoplatform.com/proof-of-work.
15. Security: Delayed Proof of Work (dPoW) // https://komodoplatform.com/security-delayed-proof-of-work-dpow.
16. Bentov I., Lee Ch., Mizrahi A., Rosenfeld M. Proof of Activity: Extending Bitcoin's Proof of Work via Proof of Stake // Cryptology ePrint Archive: https://eprint.iacr.org/2014/452.pdf.
17. What is Proof of Burn (ELI5)? // http://slimco.in/proof-of-burn-eli5.
18. What is Proof-of-Capacity? // https://www.burst-coin.org/proof-of-capacity.
19. PirlGuard – Innovative Solution against 51% // https://medium.com/pirl/pirlguard-innovative-solution-against-51-attacks-87dd45aa1109.
20. Block A. Mitigating 51% attacks with LLMQ-based ChainLocks // https://blog.dash.org/mitigating-51-attacks-with-llmq-based-chainlocks-7266aa648ec9.
21. Electronic resource: https://www.decision-making-solutions.com/how-to-make-a-decision.html.
22. Velasquez M., Hester P. T. An Analysis of Multi-Criteria Decision Making Methods // https://www.researchgate.net/profile/Patrick_Hester/publication/275960103_An _analysis _of_multi-criteria_decision_making_methods/links/55eefed208ae199d47bff202. pdf.
23. Thomas L. Saaty Decision making with the analytic hierarchy process // https://pdfs.semanticscholar.org/e3c5/61049eb532e328fc2b8288c490986cd9403f.pdf.
24. Electronic resource: https://www.univer.kharkov.ua.
25. Electronic resource: https://distributedlab.com.

Kharkiv National V.N. Karazin University;
Kharkiv NationalUniversity of Radio Electronics

Received 15.09.2019

UDC 004.056.5 **DOI:10.30837/rt.2019.3.198.09**

V.I. YESIN, Dr. Sc. (Engineering) V.V. VILIHURA

SOME APPROACH TO DATA MASKING AS MEANS TO COUNTER THE INFERENCE THREAT

Introduction

Information in the modern world has become one of the most important resources of society, and information systems (IS) whose main functional components are databases (DB) have become a necessary tool in almost all spheres of human activity providing him reliable information for making optimal decision. That in turn was reflected in the reverse side of this process. Namely, interest in the information circulating inside the IS has increased not only from legitimate users and owners, but also from attackers. The database, as the most important corporate information resource, is one of the most vulnerable and attractive elements of IS for attackers. Attackers are interested in many things: internal operational information of the organization, enterprise, company, personal data of employees, financial information, information about customers, intellectual property products, market research results, etc. The main threats to database security today are [1-5]:

- excessive and unused privileges;
- legitimate privilege abuse;
- input injection;
- malware;
- weak audit trail;
- storage media exposure;
- exploitation of vulnerable databases;
- unmanaged sensitive data;
- inference;
- denial of service;
- limited security expertise and education (the human factor);
- database communication protocol vulnerabilities and some others.

Attacks on data warehouses (DW) and databases are one of the most dangerous for various enterprises and organizations, practically any branch, be it banks and finance, medicine, trade, high technology, industry, transport, government agencies and law enforcement agencies, education, municipal institutions, etc.

According to statistics, in recent years the number of breaches and the amount of compromised data in the world has been steadily increasing. Thus, according to the estimates of the InfoWatch analytical center over the past 12 years (2007 – 2019), more than 30 billion personal data records have been compromised in the world, including over 20 billion in the last two years [6, 7]. And this is despite the fact that various security experts around the world are constantly doing a great job of researching, creating and developing principles and systems for protecting information.

Among the threats that are difficult to control within the framework of the database management systems (DBMS) and data warehous, a special place is occupied by the threat associated with the ability of attackers to make conclusions based on various algebraic calculations, comparisons, filtering the data to which they are admitted, without the need for direct access to the protected object itself [8]. This so-called threat of inference. Inference is a way to infer or derive sensitive data from nonsensitive data [1, 2, 8], which is closely related to two other ways of obtaining confidential information – aggregation and association of data [1]. The problem of aggregation and association occurs whenever a set of aggregating or associated data forms more important information than the importance of the individual data based on which this information is obtained. So information about the activities of one department or branch of the corporation have a certain weight. The data for the entire corporation are already much more significance. Or another example: a list consisting of the names of all employees and a list containing the salaries of employees are open on their own, and

the combined list with the names of employees and their salaries is already confidential information.

The usage of complex as well as a sequence of simple logically related queries allows an attacker to obtain data that is not directly accessible to him. Such a possibility is available, first of all, in databases that allow to obtain statistical data [9, 10]. Big Data is also subject to the same threat [8]. In the era of Big Data, the problem of protecting sensitive data is further exacerbated, as technical methods of protecting privacy are losing ground [11]. In this regard, for example, Google when implementing the Google's Street View (for this, photos of roads and houses in many countries of the world were collected) in Germany faced widespread public and media protests. People feared that photos of their houses and gardens could aid gangs of robbers to choose profitable targets [11].

To date, there are no perfect solutions to the inference, aggregation and association problems [8]. As a rule, countering to similar threats is carried out by such methods as [1, 2, 8, 10]:

– blocking the response with the wrong number of queries;
– control of incoming queries;
– distortion of the response by deliberate correcting the data: limited response suppression; combined results; random data perturbation; swapping; concealment;
– random sample;
– context-oriented protection;
– polyinstantiation;
– auditing and some others.

At the same time, it is important to understand that, by limiting the attacker's ability to make inferences based on the information obtained using such methods, we automatically limits queries from users who do not intend to implement unauthorized access to data. Moreover, attempts to check requested accesses for possible unacceptable inferences may actually degrade the performance of the DBMS [8].

Much of the research on inference based on data obtained in various ways from databases, data warehouses was done in the early 1980s. However, and today, aspects related to the compromise of data confidentiality obtained through inference remain largely unresolved [8].

The following is an approach to hiding data, making it difficult for an attacker to implement the inference threat. It is based on the principles of random permutation of elements (bytes, characters) of a specific field of the corresponding column (attribute) of a row (tuple) of data and dynamic data masking (DDM), defined by Gartner as an emerging technology that aims at real-time data masking of production data [12]. At that, a feature of the proposed approach is that a preliminary physical change of sensitive data is made in the production database, and it is possible, if necessary (if masking is no longer required), to lead all changes made during the masking to the initial state (without data masking) by the user who has the corresponding rights to it. This profitably distinguishes the proposed mechanism from most of the typical commercial tools for masking sensitive data. The legitimate user in the proposed approach gets access to sensitive data due to the ability to transform (rewrite) the query "on the fly", and the attacker can only read the previously modified data that is stored in the database. This approach can also be used in non-production databases, expanding the possibilities of the so-called static data masking.

1. Preliminaries. Data masking method based on the calculation of modulo operations

Today, various masking methods are known (in various sources one can also find the following terms related to information hiding, such as data anonymization, data de-identification, data scrambling, data scrubbing , data obfuscation), which are widely used in certain classes of tasks, namely [2, 13 – 15]:

– substitution. This technique consists in randomly replacing the contents of a data column by information that looks similar but completely unrelated to the real data (for example, real customer last names in the database can be replaced with last names taken from a large random list). Substitu-

tion is very effective in terms of preserving the appearance of existing data. The disadvantage is that for each column to be replaced, a large amount of replaceable information must be available;

– shuffling is a technique of random shuffling the existing field values in a table column (for example, data of a table column containing medical records about the patients' health status are randomly shuffled);

– random data deviation (random decimal numbers, random dates, random digits, random strings) is number and date variance technique. The existing value is replaced with a random one in a certain range. This technique can prevent attempts to discover true records using known date data or the exposure of sensitive numeric or date data;

– encryption. The format preserving encryption (FPE) method is used, since ordinary encryption, as rule, changes the format of the original data and may increase the data dimension, which is not always desirable;

– nulling out or deletion is simple deletion of column data by replacing it with NULL;

– masking out. This technique is a special case of the substitution technique, when all masked characters are replaced with the same symbol, for example, “X” (in this case, the credit card number would be 4343 XXXX XXXX 7357);

– technique of masking numerical data using modulo operations (MOBAT – Modulus Based Technique);

– compound masking is the technique of masking related columns as a group, ensuring that masked data across the related columns retains the same relationship. For example, consider masking address fields, such as city, state (region), and postal codes. These values must be consistent after masking;

– tokenization. In this technique, data elements are replaced with random tokens – values that should not be associated with the replaced sensitive data either mathematically or in any other way. The token does not carry any confidential information, it is only logically associated with real data that is stored in a well-protected database

and some others.

However, most of the masking techniques described above, except for the encryption, tokenization and MOBAT techniques are used for static masking of non-production databases and, after their application, do not allow canceling operations in order to return to the original data, that is not always acceptable. This is especially important for production databases if it is supposed that this mechanism will be used to counter the inference threat.

At that, the encryption method is quite resource-intensive, and MOBAT is specifically designed to mask only numerical values [15]. But, quite often there is a need for masking not only numerical values. For example, in databases built on the basis of the schema with the universal basis of relations [16, 17], that can be used, including as data warehouse of various subject domains, the attributes (columns) of relations (tables) containing sensitive data are defined on the domain of character strings.

Therefore, the need arose to find some new solution that would be no worse than the existing ones and would allow, to a certain extent, to reduce the probability of the inference threat.

After analyzing the capabilities of the above techniques, as well as the best practices of hiding information from leading vendors in the masking market [18], first an attempt was made to use mathematical transformations based on calculating modulo operations not only for numerical, but also for data types such as a character string, converted to a numerical value.

So, for example, the characters of the string 'Abc123-Ю', first converted into a hexadecimal string `4162633132332DD0AE` (similar character conversion is done in the encryption method with preservation of the format [19]), is converted to a numerical value in the decimal number system – `120612792912120891409 4`, over which the transformation is then performed similar to the MOBAT technique:

$$R'_{ij} = R_{ij} - ((K_3^i \bmod K_1^R) \bmod K_2^j) + K_2^j, \quad (1)$$

This is a direct transformation, which is necessary to mask a specific value of the field R_{ij} of the i tuple of specified attribute j of the certain table (relation) R, where R'_{ij} is the masked value of the field; K_1^R is a 128-bit random generated value (private key) that is constant for the table R; K_2^j is a 128-bit random generated value (private key) that is constant for the attribute j of the table R; K_3^i is a public key, each value of which is determined by the value of the i-th row field, one of the selected column (attribute) of the table. For this purpose, it is recommended to use a long integer typed column. Namely, as a public key, it is best to use the value of the integer identifier (ID) column of the primary key of the table R.

The inverse transformation is performed to unmask a specific value of the tuple field of the specified attribute of the certain table R:

$$R_{ij} = R'_{ij} + ((K_3^i \bmod K_1^R) \bmod K_2^j) - K_2^j, \quad (2)$$

To perform mathematical transformations (1), (2) over the obtained long integers, algorithms and programs for their implementation were developed. The algorithms associated with the implementation of computational operations on converting long integers into different number systems, without which in practical implementation it was impossible to do, are based on the Horner's rule, which allows reducing computation and memory.

However, after calculating modulo operations over converted characters, in general, the result contained not only normal (print), but also control characters, which made it difficult to represent them in a readable form. Therefore, the result was represented in the form of a hexadecimal string, which is convenient to store and process in the table field with the type of a character string, or convert to decimal form, which is convenient for numerical processing. However, such a representation could not always be suitable for various kinds of applications. In addition, when using long character strings, the implementation time was not only increased, but the high bytes were practically not transformed due to insufficient key length, except for converting string type values to numerical type. This is shown in example 2 below.

Example 1. Let in some sequence of rows of one of the table columns, whose data should be masked, the string of Cyrillic characters 'ГРС-56' is stored.

For a better representation and understanding of the corresponding data transformation, let it be sequential rows (table rows with sequentially increasing numbering (increment on ID)).

The result of applying the formula (1) to the values ('ГРС-56') of the column (we define it as DATA_X) the corresponding rows of some table is represented below:

ID	DATA_X
19746	E144A5509211952AEBE3C470A25F8C0
19747	CF518B3AC5B88CB41835AE555349764
19748	8BDC0D5DE66F27BF9280344028AB827
19749	8F48799ABEC3709CE2D45656532205E
19750	506741F22A4834CDEA6DEE65EACD651
19751	E5CBFC700E29E9374A81A902FF7BD9F
19752	C70DA272B74C3303321CA6E4FCF0A20
19753	1600D8CE183AEE9AE073EA796A06275D
19754	1649CD15E479D33CADD0193E19894C1D
19755	14D66A41674FFD516D9CC884BD8A6530
19756	C39366DA0C9E841C5A7516A42A41EDF
19757	FF1C5FA372FBA9337CE685174794AC7
19758	90A9BDB60B7BFDD79602050C8E8B1E9
19759	11DA56F36CB5A2611031BAD1BE2CC97F

Where ID column values are the public keys K_3^i for the i-th row of the table. In considered case these values are in the range [19746, 19759] ($K_3^i \in [19746,\ 19759]$).

Example 2. Let the Cyrillic character string `'Фосфоритный рудник'` is stored in some rows of the same column DATA_X.

Applying formula (1) to the corresponding data column leads to the following result (representation of the transformed character string in different table rows):

ID	DATA_MASKING
20659	D0A4D0BED181D184D0BED180D0B8D182D0BDD19EC501E1077C3320E176F0F1B28A3505
20660	D0A4D0BED181D184D0BED180D0B8D182D0BDD193555D8DA87CF439DBA3A015B0ACC2C7
20661	D0A4D0BED181D184D0BED180D0B8D182D0BDD196A3BCDB859EA0ED94B53DB4421A399B
20662	D0A4D0BED181D184D0BED180D0B8D182D0BDD1957238A723EF5F6E3989C1D314106159
20663	D0A4D0BED181D184D0BED180D0B8D182D0BDD1913798672CA2D33A572138BA761E3594
20664	D0A4D0BED181D184D0BED180D0B8D182D0BDD193224CFBA0E0590C3D70A2EACD6F05BE
20665	D0A4D0BED181D184D0BED180D0B8D182D0BDD1A1F3333683E2691BA17399EC724353CD
20666	D0A4D0BED181D184D0BED180D0B8D182D0BDD193A795A0B46246692D738F4A63F4B648
20667	D0A4D0BED181D184D0BED180D0B8D182D0BDD19F44BF9C5416208F6C5E33D44F9B811C
20668	D0A4D0BED181D184D0BED180D0B8D182D0BDD190AEE542469687FCE3FF8DD1C18DEA7C
20669	D0A4D0BED181D184D0BED180D0B8D182D0BDD19D68DAD6C0BAA7FC24FAE62B2526F3D1
20670	D0A4D0BED181D184D0BED180D0B8D182D0BDD19BF2CB316D1D2908F7FDC4F142E73CD0
20671	D0A4D0BED181D184D0BED180D0B8D182D0BDD196FA6E18263F66DA86C3793598F6B2FD
20672	D0A4D0BED181D184D0BED180D0B8D182D0BDD1A0892F4B6BE930B3E3D0D7CFFEE311CA
20673	D0A4D0BED181D184D0BED180D0B8D182D0BDD19EBF7F8AAC53585510F40366E236CF5A
20674	D0A4D0BED181D184D0BED180D0B8D182D0BDD19C1A189D7A0A289CB6F5AC214962308A

As you can see from the last example, the part of the transformed data (highlighted in color) for the same original row is the same.

To eliminate this shortcoming, it became necessary to mix them. As one of the expedient options, an obvious solution was seen, the essence of which lies in the random permutation of the corresponding bytes of the received row code with the possibility of their inverse recovery.

As is known, most of the cryptographic algorithms are still combining substitutions and permutations (transposition) [20, 21], what else C. Shannon noticed in his work [22], summarizing the experience gained before him in developing ciphers. As it turned out, even in complex ciphers, simple ciphers, such as substitution, permutation, or a combination of them, can be distinguished as its typical components. “Substitution and transposition are still the most important kernel techniques in the construction of modern symmetric encryption algorithms” [21]. Well-known computer security experts, cryptography N. Ferguson, B. Schneier in the monograph [23], talking about what would the ideal block cipher look like, note that this should be a random permutation. Specifying at the same time that for each key value the block cipher must be a random permutation of the plaintext variants and the different permutations for the different key values should be chosen independently.

The proposed solution led further to a new method.

Thus, the initial approach led to a new more efficient and less computationally expensive method – the method of random permutation of the data elements of the row field. Although in some cases, for example, when small length of rows or increasing the key length and parallelizing the computation processes, and the initial approach can be used.

2. Hiding sensitive data of row field by method of random permutation of its elements

As it is known, a permutation of n objects is an arrangement of *n* distinct objects in a row [24]. If we number the places of these objects from left to right (1,2,…,*n*), then we can formulate the following definition: the one-to-one mapping $p: A \to A$ of a finite ordered $A = \{a_1, a_2, ..., a_n\}$ set from n elements onto itself is called a permutation of elements of the A set. In the general case, for n-element set A with a fixed order of $a_1, a_2, ..., a_n$ elements the permutation is an arbitrary sequence of length *n* from different elements of the set A. Permutations of n elements of the set A differ from each other only in the order of their elements.

Permutation p can be written as a matrix of two rows. For example, the permutation $\pi: A \to A$ of the set $A = \{a,b,c,d,e\}$ such that $\pi(a) = e$, $\pi(b) = d$, $\pi(c) = a$, $\pi(d) = b$, $\pi(e) = c$ can be written as follows:

$$\pi = \begin{pmatrix} a & b & c & d & e \\ e & d & a & b & c \end{pmatrix}. \tag{3}$$

Usually the nature of the elements of the set A are inessential, so without loss of generality, we can be considered that $A = \{1,2,...,n\}$ (otherwise you must go to the element numbers (a_i, where $i \in \{1,...,n\}$)). Then each permutation π of these elements can be written as a matrix of the following two rows:

$$\pi = \begin{pmatrix} 1 & 2 & 3 & \dots & n \\ a_1 & a_2 & a_3 & \dots & a_n \end{pmatrix}, \tag{4}$$

where $\{a_1, a_2, a_3, ..., a_n\} = \{1,2,3,...,n\}$, $\pi(i) = a_i$, for all $i \in \{1,...,n\}$.

The number of all permutations π from *n* different elements is equal to $\pi_n = n!$

For each permutation π, there is a inverse permutation π^{-1} that undoes the effect of π. The product $\pi \cdot \pi^{-1}$ equals the identity permutation $\pi_e = \pi \cdot \pi^{-1}$.

The inverse permutation $a'_1, a'_2, a'_3, ..., a'_n$ is obtained, if in (4) swap the rows of the matrix, and then arrange the columns in ascending order by the upper elements:

$$\begin{pmatrix} a_1 & a_2 & a_3 & \dots & a_n \\ 1 & 2 & 3 & \dots & n \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & \dots & n \\ a'_1 & a'_2 & a'_3 & \dots & a'_n \end{pmatrix}. \tag{5}$$

Let us apply the above theoretical information from combinatorics to the solution of our task of masking the specific value of R_{ij} field of i tuple of the specified attribute j of the specific table R, through the random permutation of the sensitive data elements of the row field. Randomness in this case means the equiprobability of obtaining any of $n!$ possible permutations from the set A.

The proposed method of random permutation of elements (bytes, characters) of data of a specific field of a different type (numeric, character strings, Binary Large Objects (BLOBs), Character Large Objects (CLOBs)) of table row is based a modern version of the Fisher-Yates shuffle algorithm [25], presented by R. Durstenfeld in [26]. This algorithm (called "Algorithm FY") in pseudocode is presented below.

Algorithm FY

```
Input: A is an array with n≥2 elements (n is permutation length)
Output: random permutation on A
   for i = n downto 1
     j = random(1..i) /* a random number is generated in the range [1,i] */
     swap(A[i], A[j]) /* exchange */
   end for
```

The main reasons for choosing the Fisher-Yates shuffling algorithm were its following advantages:

– a small number of steps performed operations. The asymptotic computational complexity of the modern version of the algorithm is $O(n)$, where n is the number of elements of the set (in this case, this is the number of elements (characters, bytes) of the data of specific field);

– when using a high quality unbiased random number generator, the algorithm guarantees an unbiased result;

– its efficiency and simplicity have so far stood the test of time [27].

The Fisher-Yates algorithm uses a sample of uniformly distributed random numbers from different ranges. Therefore, it is important that one could take advantages of this algorithm, it is necessary to use pseudorandom number generators (PRNGs), which form random numbers that exactly are unbiased in some part of the interval. On the other hand, as noted in [28], the Fisher-Yates algorithm is not able to generate more than m different permutations, that is, it cannot create more permutations than the number of internal generator states. And even when the number of possible generator states exceeds the number of permutations, some of them may appear more often than others. In order to avoid the appearance of distribution unevenness, it is usually recommended that the number of internal states of a random number generator exceed the number of permutations by several orders of magnitude, if this is actually possible. Although in most cases, there is actually no need to receive all permutations [28].

Therefore, based on the foregoing, the proposed method provides for the possibility of using, depending on the situation (this is mainly due to the need to perform the procedure of permutation of the data elements of various type fields with minimal time costs), different PRNGs that satisfy the above requirements (cryptographically strong PRNG in this case is not required). We need to perform such transformations – permutations in advance so that an attacker cannot, using complex as well as sequences of simple logically related queries, obtain data that is not directly accessible to him, based on inference (that is realize the inference threat) for an acceptable time for him. At that, a legitimate user could get the required sensitive data quickly enough and simply.

In the proposed implementation of the method were used:

1) linear congruential random number generator, popularized in [29]:

$$X_{j+1} = (aX_j + c) \bmod m \tag{6}$$

with constants (multiplier $a = 1664525$ and increment $c = 1013904223$) chosen by D. E. Knuth and H. W. Lewis, where $m=2^{32}$;

2) random number generator of built-in DBMS_RANDOM package for Oracle DBMS, namely DBMS_RANDOM.VALUE, which generates floating-point numbers with 38 digits to the right of the decimal (38-digit precision), with the possibility of setting them various range;

3) G. Marsaglia pseudo-random number generator (Xorshift) [30] with a period of 2^{128}-1:

```
unsigned long xor128()
{static unsigned long x=123456789, y=362436069, z=521288629, w=88675123;
unsigned long t;
t=(x^(x<<11)); x=y; y=z; z=w;
return(w=(w^(w>>19))^(t^(t>>8)));
}
```

In the Xorshift generator, some initial sequence is specified, to which the operations of the exclusive OR (XOR) and logical shift are applied. This PRNG was selected based on the recommendations given in [31]. In principle, based on these recommendations, you can choose any other PRNG, including those given in the same work [31].

All the generators in the software implementation were checked the correspondence of the output random numbers in the given range to the uniform distribution law.

Masking algorithm

Preliminary remarks. The proposed solution uses a universal scheme for hiding data of various type fields of a tuple row of some database table, based on the use of public and private keys K_1^R, K_2^j, K_3^i. K_1^R is a unique 128-bit random value (private key) generated by a cryptographically strong PRNG for each table R (it is constant for all values that will be masked in this table). K_2^j is a unique 128-bit random value (private key) generated by a cryptographically strong PRNG for each attribute j of the table R (it is constant for all values that need to be masked in this column).

K_3^i is a public key based on the value of the integer identifier of the primary key of the i-th row of the table R (it is constant for all values in the columns that will be masked in this row).

An authorized user, with appropriate privileges, which will provide the correct key in an open session to decrypt the row of the special table (R^{secret}) encrypted with the AES-256 algorithm (keys and some other information, such as table and column names, are stored in the rows of this table), has the mediate access (through the corresponding middleware) to the private keys K_1^R, K_2^j. All other users, even privileged, without knowing this key, will not be able to extract the private keys from the table R^{secret}, and, therefore, will not be able to perform neither data masking operations nor reverse operations (unmasking or inverse masking).

Algorithm operations

1. The initial value ($X_{R_{ij}}^0$) of PRNG is generated.

For each row i of the corresponding column j of the selected table R, this value is different:

$$X_{R_{ij}}^0 = hash(K_1^R + K_2^j - K_3^i) \bmod (N_{\max}), \quad (7)$$

where $hash()$ is one of the cryptographic hash functions (such as: MD4, MD5, SHA-1, SHA-256, SHA-384, SHA-512, SHA-3). The purpose of using a hash function is mixing (non-injectively transform) private and public keys to make it impossible to recover them from the final result and getting significantly different from each other formed initial values $X_{R_{ij}}^0$ for PRNG, even if at least one of these keys changes by one character (one). $N_{\max}$ is the maximum allowable integer in the corresponding implementation. Modulo operation $N_{\max}$ is also non-injective.

2. Actions are performed in accordance with the Fisher-Yates algorithm:

– a random integer is generated (using one of the selected PRNG, to the input of which the generated initial value $X_{R_{ij}}^0$ is supplied);

– the permutation procedure of elements (bytes, characters) of the source string A of length n is performed.

As a result, we have the transformed (masked) field value (A) of each row i of the column j of the corresponding type for the selected table R.

The general scheme of the masking algorithm (MA-1) is represented below in pseudocode.

Masking algorithm 1 (MA-1)

```
Input: name_table, name_column, K3^i, A, N_max
Output: masked value of the data string – A
    Decrypt(R^secret[name_table, name_column]) → (K1^R, K2^j, PRNG, hash)
    X0_Rij = hash(K1^R + K2^j – K3^i) mod (N_max)
switch(PRNG)
{case 1: linear congruential generator (LCG)
 case 2: built-in random number generator (package DBMS_RANDOM)
 case 3: pseudo-random number generator Xorshift
 ...        }
 for i = n downto 1
  j=random_PRNG(1..i) /*a random number is generated in the range [1,i]*/
  swap(A[i],A[j])     /*exchange*/
 end for
```

Without knowing the initial value $X_{R_{ij}}^0$, it is rather difficult, and with long data strings (large dimensions A (large n)) it is almost impossible to determine the sequence of random numbers generated for the permutation (the number of which is equal $n!$). This means that it will be difficult for an attacker to determine the source strings after their corresponding transformation. Thus, we restrict an attacker to use a various set of queries for inference, for example, by presenting the same data stored in the database in a different form. As a result, misleading the attacker, we counteract the inference threat.

Example 3. Let the following string of Cyrillik characters `'КРП-17'` ($A=\{K,P,П,-,1,7\}$) be stored in some row of one of the table columns whose data should be masked.

Applying the MA-1 algorithm (*Masking algorithm 1*), and getting, for example, one of the random permutations:

$$\pi=\begin{pmatrix}1&2&3&4&5&6\\a_1&a_2&a_3&a_4&a_5&a_6\end{pmatrix}=\begin{pmatrix}1&2&3&4&5&6\\6&5&3&4&1&2\end{pmatrix},$$

we have the transformed (masked) value, namely, the string of characters $'7P-K1П'$, as a result of the mapping: $'КРП-17'\to'71П-КР'$, that is $\pi('К')='7'$, $\pi('Р')='1'$, $\pi('П')='П'$, $\pi('-')='-'$, $\pi('1')='К'$, $\pi('7')='Р'$, where a_i are the numbers of the i-th element ($i\in\{1,...,6\}$) of the source character string to be masked, each which should be placed in the appropriate position after transformation. So a_1 indicates to the sixth character ('7') of the string $'КРП-17'$ that should be placed in the first position of the transformed string; a_2 indicates to the fifth character ('1') of the string $'КРП-17'$ that should be placed in the second position of the transformed string, etc. for a_3, a_4, a_5, a_6.

Algorithm inverse to masking algorithm

Preliminary remarks. The proposed solution for recovering the masked data of the row field uses an inverse permutation algorithm, similar to that described in [24], with the peculiarity that it does not limit the permuted elements to only numbers {1, 2, 3, ..., n}, but can use any characters of national alphabets, numbers represented in hexadecimal or other number system. Namely, having an initial permutation:

$$\pi=(\pi(1),\pi(2),...,\pi(n)) \tag{8}$$

and the result of its application:

$$(y_1,y_2,...,y_n)=(x_{\pi(1)},x_{\pi(2)},...,x_{\pi(n)}), \tag{9}$$

the inverse permutation can be obtained using the formula [21, 24]:

$$\pi^{-1}(\pi(i))=i, \tag{10}$$

as:

$$(x_1,x_2,...,x_n)=(y_{\pi^{-1}(1)},y_{\pi^{-1}(2)},...,y_{\pi^{-1}(n)}), \tag{11}$$

or in matrix form, as:

$$X[\pi(i)]=Y(i). \tag{12}$$

If through π_{ch} we denote the permutation $\pi_{ch}=\begin{pmatrix}x_1&x_2&x_3&...&x_n\\y_1&y_2&y_3&...&y_n\end{pmatrix}$, where $y_i=\pi(x_i)$, $i\in\{1,...,n\}$, and through $\pi_{num}=\begin{pmatrix}1&2&3&...&n\\a_1&a_2&a_3&...&a_n\end{pmatrix}$ the corresponding permutation for it, as a mapping of the numbering onto the corresponding elements of the set, where

$\{a_1, a_2, a_3, ..., a_n\} = \{1,2,3,...,n\}$, then the formula (12) for the inverse permutation can be written as follows:

$$\pi_{ch}^{-1}(\pi_{num}(i)) = \pi_{ch}(i). \quad (13)$$

However, first you need to get the initial permutation before finding the inverse for it. Therefore, the sequence of actions will be as follows.

1. The initial value ($X_{R_{ij}}^0$) of PRNG is generated.

It is necessary to choose exactly the PRNG which was used during the initial permutation. For each row i of the corresponding column j of the selected table R, the initial value is determined in accordance with formula (7).

It should be noted that an attacker to obtain an initial value (to form an initial permutation equivalent to that formed in the MA-1 algorithm) needs to know at least the values of two private keys (K_1^R, K_2^j), as well as the type of cryptographic hash function used and PRNG used. In addition, the number of permutations (how many times has a set of operations been performed to permute row characters) for each specific row of the protected table can be different. Brute force of only two 128-bit random numbers generated by cryptographically strong PRNG is a very resource-intensive task in order to realize it in a reasonable time. The number of combinations (excluding the definition of the hash function used, PRNG and the number of times performed operations when permutation of row characters) that need to be checked for the columns j of the R table is at least $\frac{1}{2} \cdot j \cdot 2^{128} \cdot 2^{128} = j \cdot 2^{255} \approx 5.8 \cdot j \cdot 10^{76}$ (half of the total amount of possible brute force tests; 50% chance).

2. The initial permutation is determined.

Thanks to the possibility of repeating a sequence of numbers formed by the PRNG from the same initial value, performing actions in accordance with the Fisher-Yates algorithm, we obtain the initial permutation (similar to that obtained when the implementation of the MA-1 algorithm): $\pi = (\pi(1), \pi(2), ..., \pi(n))$ or in other notation $\pi_{num} = \begin{pmatrix} 1 & 2 & 3 & ... & n \\ a_1 & a_2 & a_3 & ... & a_n \end{pmatrix}$, as mapping the numbering onto the corresponding elements of a set.

3. The inverse to masking transformation is performed.

Having gotten the initial permutation $\pi = (\pi(1), \pi(2), ..., \pi(n))$, and, having the input string of the masked data $Y(i)$, in accordance with expression (14) you can determine the original (not masked) value of the row field $X(i)$, where $i \in \{1, ..., n\}$.

The general scheme of the inverse masking algorithm (IMA-1) is represented below in pseudocode.

Inverse masking algorithm 1 (IMA-1)

```
Input: name_table, name_column, K3^i, Y, N_max
Output: inverse of masked value – X
    Decrypt(R^secret[name_table, name_column]) → (K1^R, K2^j, PRNG, hash)
    X^0_Rij = hash(K1^R + K2^j – K3^i) mod (N_max)
  for i = 1 to n
    π_num[i] = i      /*array preparation*/
  end for
switch(PRNG)
{
case 1: linear congruential generator (LCG)
case 2: built-in random number generator (package DBMS_RANDOM)
```

```
case 3: pseudo-random number generator Xorshift
...
}
  for i = n downto 1    /*getting initial permutation*/
   j=random_PRNG(1..i)
     swap(π_num[i], π_num[j])
  end for

  for i = 1 to n        /*inverse of a permutation*/
    X[π_num[i] ] = Y(i)
  end for
```

Example 4. Let the character string '$7P-K1\Pi$' is stored in some field of the tuple i of the attribute j of table R as a result of masking (see Example 3). It is required to transform it to its original (before masking) state.

1. On the basis of the read values of the keys (two private and one public), knowledge of the hash function used and the PRNG, in accordance with the formula (7), the initial value for PRNG is formed. This step, naturally, is available only to an authorized user with the appropriate privileges and knowledge of the access key to the table R^{secret}.

2. Determination of the initial permutation (π_{num}) of the numbers of the elements for the character data string. After the corresponding initialization of the PRNG, which was used during the masking, actions are performed in accordance with the Fisher-Yates algorithm.

In this example, the initial permutation has the form (see Example 3):

$$\pi_{num} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ a_1 & a_2 & a_3 & a_4 & a_5 & a_6 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 6 & 5 & 3 & 4 & 1 & 2 \end{pmatrix}.$$

3. Inverse to masking transformation.

In accordance with the formula (12) we have: $X[6] = Y(1) = '7'$; $X[5] = Y(2) = '1'$; $X[3] = Y(3) = '\Pi'$; $X[4] = Y(4) = '-'$; $X[1] = Y(5) = 'K'$; $X[2] = Y(6) = 'P'$.

Having ordered the indexes in ascending order, we get the string: 'КРП-17'. The resulting value is equivalent to the masked value (see Example 3). That confirms the correctness of the algorithm IMA-1 (*Inverse masking algorithm 1*).

3. Comparative analysis of the quantitative and qualitative characteristics of the proposed algorithm and an encryption algorithm

Comparative analysis of quality characteristics

For a better representation and understanding of the corresponding data transformation (quality characteristics), Table 1 below shows the masking results of the character string 'КРП-17' using the proposed algorithm using various PRNGs and the ordinary encryption algorithm AES-128. The following PRNGs were used: 1 – linear congruential generator (expression (6)); 2 – random number generator of built-in DBMS_RANDOM package for Oracle DBMS; 3 – pseudorandom number generator Xorshift with the (2^{128}-1) period. For clarity, the adjacent rows were selected (rows of the table with sequentially increasing numbering).

When encrypting character strings using the AES algorithm, the same mechanism for extracting and applying keys from the table R^{secret} was used as in the above proposed masking algorithm. Herewith the result of the formula (7) was used to determine the initialization vector in the encryption algorithm.

As we can see from Table 1, all the resulting strings are different from the original string. And this is important! But at that, the converted strings using the proposed approach retain the format and do not increase the dimension of the string, as in the case with the encryption algorithm. That in certain applications is critical and unacceptable. In this respect, the ordinary encryption algorithm concedes the proposed method. To eliminate this shortcoming, you can use encryption algorithms

with format preservation (format-preserving encryption – FPE), but their implementation, for example, in Oracle, as noted in [14], involves significant processing.

Table 1

Character string masking results

ID	DATA_MASKING 1	DATA_MASKING 2	DATA_MASKING 3	DATA_ENCRYPT
18136	П-Р7К1	Р1КП7-	-71ПРК	F03E00DBD4D478A1D331F6CFAEC6A3DF
18137	1П7-КР	П71РК-	К1Р7-П	689783C6F95F526126C48FE6F11BEF13
18138	П-7КР1	РП17-К	КП1Р7-	C5B6BAFB17C10C528A7490B66E19A2B0
18139	П1К-Р7	Р-К7П1	-1К7ПР	E908DB627314C84B572486457ED0670A
18140	Р17П-К	КП7Р-1	Р7КП-1	4766A178B39586108DDB57F8ADBE5A5A
18141	17ПРК-	71-КРП	КП17Р-	6ECF167783564759E8D31753D59A53E3
18142	-КР7П1	1К-РП7	КР-П71	0F0D486A4F9CB189FA7092CB2CA4D47F
18143	1ПР7К-	71КП-Р	-РКП71	315690B8B84DA3460A4F4933DCA7648F
18144	Р7-К1П	КПР1-7	РК-71П	59DAA907A9CC38FF1A52C8AB9CD9DCC8
18145	7-РК1П	К7-1РП	1КР7П-	58F46C87949558D2C02CCBE3F368D09F
18146	КП7Р1-	П-17КР	Р1П7-К	DD3F2CC50EBD51FE991B8334EB50C347
18147	Р1П7-К	К-7ПР1	КР17-П	CD50A2ECA5D987419BB8C6794BDCBE71
18148	-К7ПР1	ПК71Р-	К-7ПР1	6549D15CC334245CDF7182CC2983C3C4
18149	-1РП7К	РК1-7П	7-РК1П	B37403B75997F2A535FA002239EDDEF5
18150	П71РК-	-П1КР7	К7П-1Р	D0C9509DDBD4ED6952E4434EA5FC13C2
18151	К7П-1Р	-К7Р1П	-Р7П1К	975AE4AEF567EDEEC9E40EC7F48A75D9
18152	Р17П-К	7-К1ПР	71П-КР	82084A1E15F818543D0845A535810F7D
18153	П1К-7Р	Р-К7П1	ПК1-Р7	9E7F647331CAF130D02C9CBCBB53968A
18154	П-Р17К	-К71ПР	Р-КП71	2C85EDFE7307EAF7EF11B7766B7416DF
18155	Р7ПК-1	К7-П1Р	ПР-71К	8FECDE765DE3773788C79040118B9A6B
18156	7-1КРП	КП17-Р	К-П1Р7	25F2F78484CD4872FC8FE6E0A35AB22F
18157	П1КР7-	КР-7П1	Р7-1ПК	CD62C80E53122CCA4AEF2472734EC138
18158	Р-7КП1	К7-П1Р	ПК-Р17	2DF64FC7780264AEF4E7D3BF49214ACF

Analyzing the results of Table 1, it can conclude that an attacker has little chance to determine from the available information that all these values are associated with the same character string `'КРП-17'` (with the same real object). Although the number of permutations is not so large 6!=720, but in this particular case even statistical cryptanalysis is difficult, since, in fact, after the corresponding transformations, it may turn out that completely other and different objects of the modeled subject domain will be associated with the obtained transformed names. This significantly makes difficult the implementation of inference threat for an attacker. And, therefore, the proposed approach to hiding will not allow an attacker to obtain data in a reasonable time for him, access to which is directly closed to him.

Sometimes it is necessary to mask part of the field value of a table row, for example, for masking phone numbers, discount cards, bank cards, serial numbers of equipment and devices, car numbers, etc. In this case, the ordinary encryption method is not acceptable. The proposed approach, on the contrary, is suitable for solving this problem; for this, only some modification of the MA-1 algorithms (*Masking algorithm 1*) and IMA-1 (*Inverse masking algorithm 1*) represented above is necessary.

Schemes of the modified algorithms MA-2 and IMA-2 (differences from MA-1 and IMA-1 are highlighted in color) are represented below.

Masking algorithm (MA-2)

Input: *name_table*, *name_column*, K_3^i, A, $N_{\max}$, i_end, i_beg

Output: masked value of the data string – A

Decrypt(R^{secret}[*name_table*, *name_column*]) → (K_1^R, K_2^j, *PRNG*, *hash*)

$X^0_{R_{ij}} = hash(K_1^R + K_2^j - K_3^i) \bmod (N_{\max})$

```
switch(PRNG)
{
case 1: linear congruential generator (LCG)
case 2: built-in random number generator (package DBMS_RANDOM)
case 3: pseudo-random number generator Xorshift
...
}
 for i = n-i_end downto i_beg
  j=random_PRNG(i_beg..i) /*a random number is generated in the range [i_beg,i]*/
  swap( A[i], A[j])      /*exchange*/
 end for
```

Inverse masking algorithm (IMA-2)

Input: *name_table*, *name_column*, $K_3^i, Y, N_{\max}$, i_end, i_beg

Output: inverse of masked value – X

Decrypt($R^{sec\,ret}$[*name _ table*, *name _ column*]) → (K_1^R, K_2^j, *PRNG*, *hash*)

$X^0_{R_{ij}} = hash(K_1^R + K_2^j - K_3^i) \bmod (N_{\max})$

```
  for i = 1 to n
    π_num[i] = i      /*array preparation*/
  end for
switch(PRNG)
{
case 1: linear congruential generator (LCG)
case 2: built-in random number generator (package DBMS_RANDOM)
case 3: pseudo-random number generator Xorshift
...
}
  for i = n-i_end downto i_beg    /*getting initial permutation*/
   j=random_PRNG(i_beg..i)
     swap(π_num[i], π_num[j])
  end for
  for i = 1 to n         /*inverse of a permutation*/
    X[π_num[i]] = Y(i)
  end for
```

Where the parameters i_beg, i_end define the boundaries of the transformation interval, namely: i_beg – from which position from the beginning of the string the permutation should be performed; i_end – up to which position from the end of the string permutation should be performed.

In addition, to mask, for example, bank cards after the corresponding permutation procedure (see algorithms MA-2, IMA-2), it is also necessary to calculate the last check digit using the Luhn algorithm [32], in accordance with the ISO/IEC 7812 standard.

Tables 2, 3, 4 show the masking results using the proposed algorithm (with different PRNG: 1 – LCG – the result in the DATA_MASKING 1 column; 2 – built-in PRNG – result in the DATA_MASKING 2 column; 3 – PRNG Xorshift – result in the DATA_MASKING 3 column) of data on card numbers stored in the corresponding sequences of adjacent rows of some table, namely, on Visa bank card – number '4454102135347018', Master Card bank card – number '5167135104128196', American Express payment system card – number '378282246310005', with preservation of the card type.

If you need to mask some table fields that store, for example, important text documents (data in one of the text formats: both in TXT format itself and in other, for example, such specialized formats as INI, HTML, XML, TeX, JSON, LOG, source texts of programming languages and others for which it serves as the basis), which are significantly larger in volume (length) the character strings discussed above, then this task can also be effectively solved using the proposed method.

Table 2

The masking results of Visa bank card number

ID	DATA_MASKING 1	DATA_MASKING 2	DATA_MASKING 3
35359	4411443305170521	4401235043741510	4400451172135348
35360	4413021740415533	4437312451510402	4413535021740418
35361	4434230511514078	4451124347013504	4433412071450155
35362	4441031715425306	4454317254110308	4445701534021132
35363	4410203415715438	4401473215035144	4401112347355406
35364	4471310435041525	4431424310075158	4403441103715257
35365	4442031413755102	4453113442705013	4450104341275318
35366	4434032105174511	4470310431215454	4432740405113516
35367	4442714501531309	4413274510410354	4401351344017250
35368	4411233574040152	4401354023541715	4470315450124131

Table 3

Masking results the number of Master Card bank card

ID	DATA_MASKING 1	DATA_MASKING 2	DATA_MASKING 3
35369	5174311028615912	5121381165147098	5173192416150188
35370	5193751482161100	5116471112850931	5111091716482355
35371	5193711261150849	5173051612198419	5131072681541199
35372	5174513918161028	5124311678019515	5171350492111688
35373	5110341258911670	5176018951231413	5162081179141356
35374	5111501217936485	5114811259061374	5193110548211674
35375	5105311179468123	5168143975012118	5143196250811174
35376	5113517462018918	5129713164051180	5171591241013689
35377	5101512147198361	5164113082517196	5192304161118750
35378	5141351291186076	5118107163219455	5171084112195363

Table 4

Masking results the card number
of the American Express payment system

ID	DATA_MASKING 1	DATA_MASKING 2	DATA_MASKING 3
35379	372800263182043	370020288642317	371288240036022
35380	378022631084025	370816438022205	372206321048087
35381	372006822180433	370010822348265	372142063008825
35382	373206140282809	372382004682011	376482020308215
35383	372860104823205	371242806280300	372200346821081
35384	370280463102285	376182320802400	372302880420161
35385	372843260180024	376048820221039	372042318028604
35386	378231264800025	376082412008322	371862302802040
35387	372012648080237	370102628084325	373802460082124
35388	372020023868415	378023801262045	372822460183004

The proposed method, as it was noted above, can be applied to the masking of the fields of tables that store data not only of the character string type (*character*, *character varying*) and numeric types (*integer*, *number*), but also data of the BLOB, CLOB type that allow you to store in the data-

base, for example, documents of such formats as: DOC, DOCX, RTF, XLS, XLSX, XPS, PDF, PPT, BMP, GIF, TIF, MP3, AVI, etc. But this important question, which has interesting features of practical implementation, in this paper, in view of its limited scope, will not be considered. This is the subject of a separate article.

Quantitative characteristics of comparative analysis

The results below were obtained with the appropriate data transformations of the database tables. The database was implemented on the Oracle 12.2 c DBMS platform installed on the Windows 10 (x64) operating system on various computers.

Option A. Computer with Intel (R) Core (TM) 2 Duo CPU 2.16 GHz, RAM 4 GB, HDD 320 GB was used.

Option B. Computer with Intel(R) Core (TM) i3-7100 CPU 3.90 GHz, RAM 4 GB, HDD 500 GB was used.

About 18,000 data rows of some *varchar2*(255) column of a table of real database were subjected to transformation (masking, encryption) with recording to the database. Fig. 1, 2 shows the results of the average times (in seconds) of masking and unmasking, encrypting and decrypting character strings with writing to the database all 18,000 data rows using the proposed permutation algorithm (were used linear congruential PRNG (PRNG-1) and pseudo-random number generator Xorshift (PRNN-3)) and the ordinary AES-128 encryption algorithm for options A and B respectively.

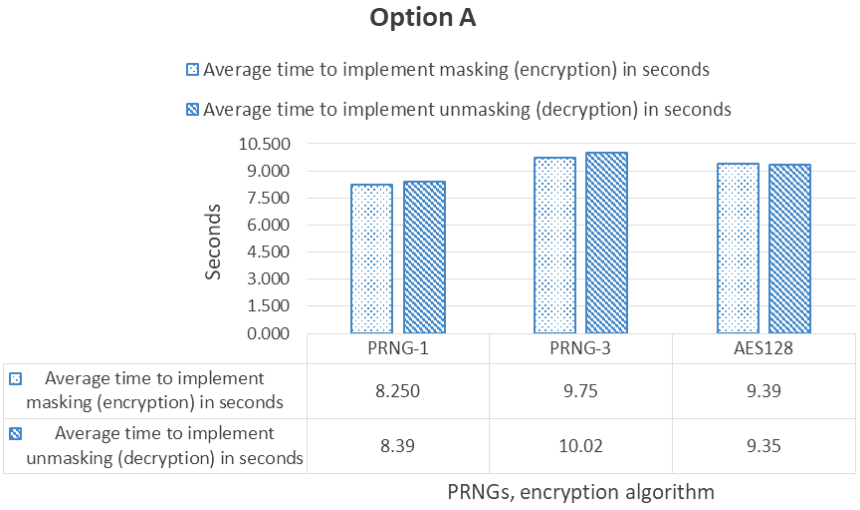


Fig. 1. Average time to transform character strings (option A)

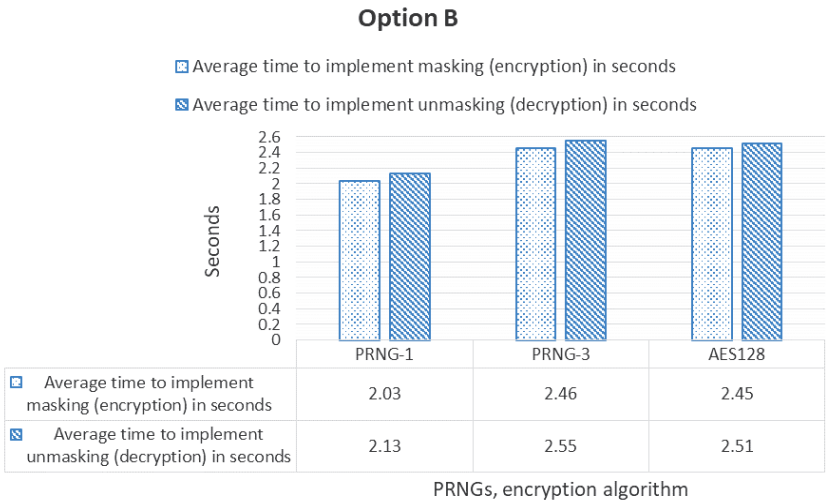


Fig. 2. Average time to transform character strings (option B)

As follows from the conducted research, the usage of the ordinary encryption algorithm loses to the proposed method not only in qualitative characteristics, namely, changes in the presentation format, but also quantitative due to the increase in the length of the stored data and the transformation time for hiding data. If, when using Xorshift PRNG, the compared times of corresponding transformations for both methods are almost the same, then already compared to the permutation algorithm, when using the linear congruential PRNG, the implementation of the encryption algorithm loses to it (11-12)% for option A and (15-17)% for option B, respectively. (Note. The current implementation of Xorshift PRNG in PL/SQL is currently not an optimized implementation due to this language does not support hardware implementation of logical shifts, XOR operations, and therefore these ones are solved algebraically).

4. Usage restrictions of the proposed method and features of its implementation

When the length of the string is less than three characters or, if all the characters are the same, masking using the proposed method is inexpedient. In this case, it is advisable to use:

– or a substitution method with elements of the proposed approach, namely, by creating a mapping table with a random selection from it of characters or their combinations, using the available PRNGs and the shuffling mechanism of the proposed permutation method for this purpose (with the ability to subsequently restore the sequence, required for the inverse transformation, of the generated numbers from the initial value $X_{R_{ij}}^{0}$);

– or a method of masking data based on the calculation of operations modulo, by transformation of a short string to a numerical form, if it includes letter characters, special characters and digits, or by simply converting a string of digits to a number using standard conversion functions (for example, for Oracle, this is the `TO_NUMBER` function).

The privacy of the masking keys depends on where the keys are stored and who has access to them. The proposed solution uses three keys (K_1^R, K_2^j, K_3^i): two are private (generated cryptographically strong PRNG) and one is public. All private keys are encrypted using the AES-256 algorithm and stored in a special database table R^{secret}. The values of these keys are never shown and not known either to the database administrator (if he does not combine the functions of the security administrator), or to any other user. An authorized user, with appropriate privileges, which will provide the correct key in an open session to decrypt the row of the special table, has the mediate access through the corresponding middleware to the private keys K_1^R, K_2^j. At that, the value of this key is not shown anywhere in the clear, it cannot be traced even through the available means of documenting executed queries (a historical command log). It is extracted in a certain way by the special software of the DBMS server from the key container file, which an authenticated user with the appropriate privileges must provide during the session opening.

Actions in case of encryption key compromise of the table R^{secret}

If the key with which the data (the private K_1^R, K_2^j keys and some other information) of the R^{secret} table is encrypted, is compromised, it can be replaced with a new generated one using specially developed software (cryptographically strong PRNG is used for this). After that, the data of the entire table R^{secret} is encrypted with the new key. At that, the private K_1^R, K_2^j keys are not shown anywhere explicitly. Then, using special software, for legitimate users key container files for decrypting the data of the table R^{secret} are formed.

Actions in case of compromise of separate private K_1^R, K_2^j keys

When the compromise of separate private keys from sets of K_1^R, K_2^j they can also be generated using cryptographically strong PRNG and special developed software similarly the decryption

key of the R^{secret} table. After that, it remains only to replace the compromised private keys with new ones, pre-encrypting them with an AES-256 algorithm. At that, the newly generated private keys are also not shown anywhere explicitly.

Conclusions

1. Analyzing the best practices of hiding information from leading vendors, a new approach to data hiding was proposed, making it difficult for an attacker to implement the inference threat. This approach was based on the principles of random permutation of the elements of a specific field of the corresponding column (attribute) of the row (tuple) of the production database table data and dynamic masking. It differs from the existing ones in that a preliminary physical change of sensitive data is made in the production database, and a user who has the appropriate rights can cancel these changes if it is necessary. That is, the corresponding user can restore all data changes made during the masking procedure to their original state.

2. The authenticated user in the proposed solution gets access to sensitive data due to the ability to transform (rewrite) the query "on the fly", and an attacker, even using complex as well as sequences of simple logically related queries, is limited in implementing the threat of inference during an acceptable time for him (due to the fact that only the masked data is available to him).

3. It is possible to mask both an entire value of the field of the table row and its part using the proposed solution. This is relevant on the one hand for masking such data as phone numbers, discount cards, bank cards, serial numbers of equipments and devices, car numbers, etc. and on the other hand, it allows reducing the number of operations for transformation large binary objects, and, consequently, the implementation time, without losing the effectiveness of the masking procedure.

4. Studies have shown that the use of the ordinary encryption algorithm loses to the proposed method not only by qualitative characteristics (primarily due to a change in the data representation format and the inability to transform part of the row field value), but also by quantitative characteristics (due to an increase in the length of the stored data and the transformation time for hiding data (by about (10-17)%)).

5. A distinctive feature of the proposed solution is the approach to the process of data shuffling, namely, shuffling data value elements within the demanded row field. At that, the basic operations of the proposed method can also be used for vertical shuffling – permutation of values within the column of the selected table.

6. The proposed approach to data masking can be used in both production and non-production databases, expanding the possibilities of so-called static data masking.

References

1. Sandhu R. S., Jajodia S. (1993) Data and database security and controls // Handbook of information security management. Auerbach Publishers, pp. 481-499.

2. Kulkarni S., Urolagin S. (2012) Review of attacks on databases and database security techniques // International Journal of Emerging Technology and Advanced Engineering, **2**(11), pp. 2250-2459.

3. Top ten database security threats. The most significant risks of 2015 and how to mitigate them, Imperva Whitepaper, 2015 [Electronic resource]. Access mode : https://informationsecurity.report/Resources/Whitepapers/e763d022-6ee4-4215-9efd-1896b0d9c381_wp_topten_database_threats%20imperva.pdf, last accessed 2019/09/01.

4. Rohilla S., Mittal P. K. (2013) Database Security: Threads and Challenges // International Journal of Advanced Research in Computer Science and Software Engineering, **3**(5), pp. 810–813.

5. Top 5 Database Security Threats, Imperva Whitepaper, 2016 [Electronic resource]. Access mode: https://www.imperva.com/docs/gated/WP_Top_5_Database_Security_Threats.pdf, last accessed 2019/09/01.

6. Infowatch. Analytics. Digests and Reviews. Over 12 years, more than 30 billion personal data records have leaked [Electronic resource]. Access mode : https://www.infowatch.ru/analytics/digest/15281, last accessed 2019/09/01. (in Russian)

7. Global research on confidential information leaks in 2018, Analytical center InfoWatch, 2019 [Electronic resource]. Access mode: https://www.infowatch.ru/sites/default/files/report/analytics/russ/InfoWatch_Global_Report_2018_year.pdf?rel=1, last accessed 2019/09/01. (in Russian)

8. Pfleeger C. P., Pfleeger S. L., Margulies J. (2015) Security in Computing (Fifth Edition). Prentice Hall, 944 p.

9. Wang L., Jajodia S. (2008) Security in Data Warehouses and OLAP systems // Handbook of Database Security, Springer, Boston, MA, pp. 191-212.

10.Zavgorodniy V. I. (2001) Complex information protection in computer systems. M. : Logos, PBOYUL N.A. Egorov, 264 p. (in Russian).

11.Mayer-Schonberger V., Cukier K. (2013) Big Data: A Revolution That Will Transform How We Live, Work and Think. Canada, Eamon Dolan/Houghton Mifflin Harcourt, 242 p.

12.Gartner IT Glossary [Electronic resource]. Access mode : https://www.gartner.com/it-glossary/dynamic-data-masking-ddm, last accessed 2019/09/01.

13.A Net 2000 Ltd. White Paper. Data masking: What You Need to Know Before You Begin [Electronic resource]. Access mode : http://www.datamasker.com/DataMasking_WhatYouNeedToKnow.pdf, last accessed 2019/09/01.

14.Data Masking and Subsetting Guide [Electronic resource]. Access mode : https://docs.oracle.com/en/database/oracle/oracle-database/12.2/dmksb/introduction-to-oracle-data-maksing-and-subsetting.html#GUID-24B241AF-F77F-46ED-BEAE-3919BF1BBD80, last accessed 2019/09/01.

15.Santos R. J., Bernardino J., Vieira M. A. (2011) Data masking technique for data warehouses // Proceedings of the 15th Symposium on International Database Engineering & Applications, ACM, pp. 61-69.

16.Yesin V. I. (2018) Invariant to subject domains database schema and its distinctive features // Radiotekhnika : 193, pp. 133-142 (in Russian)

17.Yesin V. I., Yesina M. V., Rassomakhin S. G., Karpinski M. (2018) Ensuring Database Security with the Universal Basis of Relations // Saeed K., Homenda W. (eds) Computer Information Systems and Industrial Management. CISIM 2018. Lecture Notes in Computer Science, **11127**, Springer, Cham, Chapter 42, pp. 510-522.

18.Tirosh A., Meunier M. (2015) Magic Quadrant for Data Masking Technology, Worldwide Published: 22 December 2015 ID: G00273093 [Electronic resource]. Access mode : https://docplayer.net/12460751-Magic-quadrant-for-data-masking-technology-worldwide.html, last accessed 2019/09/01.

19.Dworkin M. (2019) Recommendation for block cipher modes of operation. Methods for format-preserving encryption // Draft NIST Special Publication, № 800-38G Revision 1 [Electronic resource]. Access mode : https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-38Gr1-draft.pdf, last accessed 2019/09/01.

20.Schneier B. (1996) Applied cryptography: protocols, algorithms, and source code in C (2nd edition), John Wiley & Sons, Inc., 758 p.

21.Mao W. (2003) Modern Cryptography: Theory and Practice. Prentice Hall PTR, 707 p.

22.Shannon C. (1949) Communication Theory of Secrecy Systems // Bell System Technical Journal, **28**(4), pp. 656-715.

23.Ferguson N., Schneier B. (2003) Practical cryptography. New York, Wiley, 432 p.

24.Knuth, D. E., (1997) The Art of Computer Programming, Volume 1: Fundamental Algorithms (3rd ed.), Addison-Wesley Professional, 650 p.

25.Fisher R. A, Yates F. (1948) Statistical Tables for Biological, Agricultural and Medical Research (3rd Edition). Edinburgh and London, **13**(3), pp.26-27.

26.Durstenfeld R. (1964) Algorithm 235: Random permutation // Communications of the ACM, **7**(7), pp. 420.

27.Bacher A., Bodini O., Hollender,A., Lumbroso J., (2018) Merge Shuffle: a very fast, parallel random permutation algorithm // Proceedings of the 11th International Conference on Random and Exhaustive Generation of Combinatorial Structures Athens, Greece, June 18-20, CEUR-WS.org/Vol-2113, pp. 43-52.

28.Knuth D. E. (1997) The Art of Computer Programming, Volume 2: Seminumerical Algorithms (3rd ed.). Addison-Wesley, Reading, MA, 762 p.

29.Press W. H., Flannery B. P., Teukolsky S. A., Vetterling W. T. (1992) Numerical Recipes in C: The Art of Scientific Computing (Second Edition). Cambridge University Press, 994 p.

30.Marsaglia G. (2003) Xorshift rngs // Journal of Statistical Software, **8**(14), pp. 1-6.

31.Press W. H., Teukolsky S. A., Vetterling W. T. (2007) Flannery B. P. Numerical Recipes: The Art of Scientific Computing (3rd ed.). New York, Cambridge University Press, 1235 p.

32.Patent No. 2,950,048, United States, Computer for Verifying Numbers / H. P. Luhn, Armonk, N.Y., assignor to International Business Machines Corporation, New York, N.Y., a corporation of New York. Ser. No. 402,491; Aug. 23, 1960.

33.Bacher A., Bodini O., Hwang H. K., Tsai T. H. (2017) Generating random permutations by coin tossing: Classical algorithms, new analysis, and modern implementation, ACM Transactions on Algorithms, **13**(2), 43 p.

34.Ravikumar G. K., Justus R., Ravindra S. H., Manjunath T. N., Archana R. A. (2011) Experimental study of various data masking techniques with random replacement using data volume // International Journal of Computer Science and Information Security, **9**(8), pp.154-158.

Kharkiv National V.N. Karazin University *Received 05.09.2019*

УДК 621.391:519.2:519.7 **DOI:10.30837/rt.2019.3.198.10**

Ю.І. ГОРБЕНКО, канд. техн. наук, М.В. ЄСІНА, канд. техн. наук,
Д.В. МЯЛКОВСКИЙ, О.С. АКОЛЬЗІНА, В.А. ПОНОМАРЬ, канд. техн. наук

СУЧАСНІ ПРОБЛЕМИ ЦЕНТРАЛІЗОВАНИХ ТЕХНОЛОГІЙ ТИПУ «КЛІЄНТ – СЕРВЕР» ТА МОЖЛИВОСТІ ЇХ УДОСКОНАЛЕННЯ НА ОСНОВІ ДЕЦЕНТРАЛІЗАЦІЇ

Вступ

Нині спостерігаються інтенсивні процеси розроблення та впровадження у різноманітні інформаційні технології (ІТ) принципів децентралізації. Основні з них закладені в технології блокчейн [1 – 5, 9]. Практично широке впровадження технології блокчейн (ТБЧ) зроблено в якості криптовалюти. Але, незважаючи на появу та впровадження на основі ТБЧ перспективних широкомасштабних розробок, продовжують існувати певні сумніви відносно перспектив застосування ТБЧ. Вони, як і більшість нових інформаційних технологій, можуть бути обмеженими в застосуванні, а то і непотрібними [1 – 6].

Нині широке розповсюдження та застосування знайшли ІТ, що ґрунтуються на технології типу «Клієнт – сервер» [7, 8]. Технології «Клієнт – сервер» (англ. Client-server) є обчислювальними або мережевими архітектурами, у яких завдання або мережеві навантаження розподілені між постачальниками послуг, яких називають серверами, та замовниками послуг, яких називають клієнтами. По суті, технології «Клієнт і сервер» є різної складності програмним забезпеченням, що розташовується на різних обчислювальних машинах і взаємодіють між собою через обчислювальну мережу за допомогою мережевих протоколів. Вони можуть бути розташовані також і на одній машині. Сервери очікують від клієнтських програм певні запити і надають їм свої ресурси у вигляді даних, сервісних функцій. Оскільки одна програма-сервер може виконувати запити від безлічі програм-клієнтів, її розміщують на спеціально виділеній обчислювальній машині або машинах, налаштованих особливим чином, наприклад, спільно з іншими програмами-серверами. Тому, продуктивність та захищеність тощо, такої «машини – сервера» повинні бути високими. Враховуючи особливу роль такої машини в мережі, специфіки її обладнання та програмного забезпечення, її також називають сервером, а машини, які виконують клієнтські програми, відповідно, клієнтами. Технології «Клієнт – сервер» мають ряд переваг, але в останні роки проявились і ряд їх недоліків.

До основних переваг можна віднести [7, 8]:

- відсутність необхідності дублювання програмного забезпечення сервера клієнтами;
- вимоги до комп'ютерів, на яких встановлено клієнт, знижуються, так як усі обчислення виконуються на сервері;
- усі дані зберігаються на сервері, який, як правило, захищений набагато краще ніж клієнт;
- на сервері простіше організувати контроль повноважень, щоб вирішувати доступ до даних тільки клієнтам з відповідними правами доступу тощо.

Але, в останні роки щодо технологій клієнт – сервер виявлено і ряд недоліків [7, 8], до яких необхідно віднести:

- втрата працездатності сервера приводить до непрацездатності клієнтів або неякісного їх функціонування (непрацездатним сервером слід вважати сервер, продуктивності якого не вистачає на обслуговування всіх клієнтів, а також сервер, що знаходиться на ремонті, профілактиці тощо);
- при зниженні захищеності сервера появляється можливості несанкціонованого доступу до даних, що розміщені на клієнті та компрометації даних, ключів та ключових даних тощо;
- відновлення якісного надання основних послуг безпеки вимагає суттєвих часових та матеріально – технічних ресурсів ;

- підтримка роботи працездатності та безпеки даних ІТ технології клієнт – сервер вимагає використання окремих фахівців – системного адміністратора та адміністратора безпеки тощо;
- висока вартість обладнання та програмного забезпечення сервера тощо.

Тому важливими є відповіді на питання – чи можна та яким чином удосконалити системи «клієнт – сервер». При цьому першим питанням, на яке потрібно відповісти, це як оцінити покращення. Відповіді на дане питання можуть бути для звичайних користувачів та суспільства достатньо простими – нові технології повинні давати принципові та істотні покращення, в порівнянні з тими, що вже є.

Важливим є такий фактор, як людська звичка. Суспільство та суб'єкти (люди) переходять на нові технології тільки в разі, якщо вони дають суттєву перевагу, не просто на 5 – 10 %, а мінімум в 2-3 рази. При цьому, як підтверджено практикою, що для нових ІТ надважливими є такі критерії та показники оцінки та порівняння як вартість, складність (часова та просторова), швидкість, прибутковість, безпечність, анонімність, гнучкість, дизайн тощо [1, 9 – 12].

Наприклад, в найбільш поширеній децентралізованій технології «Біткоїн» необхідно вирішити проблеми курсу та ризики щодо криптовалюти, обміну в звичайні долари, легальності криптовалюти тощо [1, 9]. Також для певної цільової аудиторії необхідно щоб нові властивості, які дає технологія криптовалюти «Біткоін», відповідали новим вимогам. Тому вже при проектуванні та розробленні і впровадженні нових ІТ, в першу чергу що грунтуються на децентралізації, потрібні як глибокі теоретичні дослідження, так і практичні результати щодо вказаних вище характеристик.

За результатами аналізу системних підходів та досвіду застосування нових ІТ спеціалістами [2, 4, 12] сформульовано наступні необхідні, (але недостатні) умови можливого широкого розповсюдження та застосування децентралізованих технологій, в тій чи іншій бізнес-сфері чи інформаційних системах. До них необхідно віднести такі [9, 1 – 6].

1. Застосування децентралізації повинне поліпшити хоча б один із важливих для цільового застосування параметрів: вартість, складність (часова та просторова), швидкість, прибутковість, безпечність, анонімність, гнучкість, дизайн тощо. При цьому важливо, щоб параметр повинен бути використаний не той, що пропонує розробник, а той, що пропонує користувач (замовник).

2. Покращення має бути суттєвим, децентралізація повинна покращувати хоча б один важливий для користувачів параметр, як мінімум в два рази.

3. Нова технологія, в нашому випадку ТБЧ, не повинна істотно програвати старим технологіям за іншими параметрами. Наприклад [9], якщо ТБЧ працює в три рази швидше, але якщо вона при цьому в два рази дорожче і в 1,5 рази складніше – вона, скоріше всього, не отримає визнання та застосування. В якості прийнятного порогу програшу можна взяти біля 30 – 50 %. У цілому нова технологія повинна бути в чомусь краще в 2-3 рази, а за всіма іншими параметрами програвати не більше ніж в 1,5 рази. Тобто, покращення повинне давати суттєві переваги, але ще і компенсувати побічні ефекти щодо програшу за іншими параметрами.

Таким чином, "кращість" і перевага нової технології – це суб'єктивна річ, вона формується користувачами, а не розробниками. Тільки продажі доводять наявність "кращості" та переваг. Відсутність продажів показує, що явної переваги цільовою аудиторією поки ще не визнано.

Таким чином, є проблема, сутність якої в тому, що децентралізація, в тому числі у вигляді ТБЧ, є начебто перспективною технологією, але де саме та як її краще використовувати, де вона буде мати в порівнянні з існуючими технологіями « Клієнт – сервер» кращою, на наш погляд, є не вирішеною .

Метою статті є:

- аналіз основних принципів побудування децентралізованих технологій з використанням ТБЧ та вимоги до них в частині безпеки;
- аналіз особливостей та умов застосування захищених ТБЧ;
- опис на аналіз потенційних атак, коли застосування БЧ є суттєвим механізмом захисту від них;
- сутність та пропозиції відносно протидії атакам спеціального виду.

Автори розуміють, що стаття носить характер первинного загальносистемного аналітичного огляду та відображає погляди авторів на ТБЧ. Ми сподіваємось, що в подальшому буде опубліковано серію науково-практичних статей щодо теорії та практики ТБЧ.

1. Основні принципи побудування ТБЧ

З метою реалізації системного підходу до аналізу та оцінки захищеності розглянемо визнані основні принципи та вимоги щодо побудування ТБЧ. Згідно [4, 9 – 12] при побудуванні ТБЧ повинні бути застосовані чи рекомендовані до застосування такі базові принципи як : мережевої цілісності; розподілення влади; цінності як стимулу для користувачів; захисту (безпеки) інформації та ресурсів; приватності інформації та ресурсів. Важливими також є принципи створення програмного забезпечення; послуг технологій; бізнес моделей та ринків; організація функціонування; при необхідності також управління БЧ тощо. Вказані принципи в основоположній роботі Сатоші у явному вигляді [9,10] не виділялись, але вони використовуються практично для усіх платформ БЧ. Розглянемо базові з них детальніше та будемо враховувати їх при аналізі та оцінках в подальшому [2, 3].

1.1. Принцип мережевої цілісності

Сутність цього принципу в тому, що довіра стосовно БЧ є внутрішньою, а не зовнішньою [3]. Цілісність забезпечується на кожному етапі процесу і є основною її цінністю. Учасники можуть обмінюватись такою цінністю, сподіваючись, що інша сторона діятиме також чесно. Основна її складова – цінність цілісності – чесність у своїх словах і вчинках, врахування інтересів інших, відповідальність за наслідки своїх рішень і дій, а також прозорість у прийнятті рішень та дій. Вона закладена в правах на рішення та в структурах стимулювання. Основна вимога полягає в тому, що діяти без цілісності неможливо або це потребує набагато більше часу, грошей, енергії та репутації [2, 3, 9], якщо вона не забезпечується.

1.2. Принцип розподілення влади

Принцип розподілення влади у наступному [2, 3, 9 – 12]. В одноранговій мережі БЧ влада по суті розподілена, причому точка контролю відсутня. Будь – яка зі сторін не може припинити роботу системи БЧ. Якщо певному центральному управлінню вдасться компрометувати або відрізати індивідуума або групу, система БЧ все ще буде життєздатною. Якщо більше половини мережі спробує знищити всю мережу, то усі побачать, що діється, та будуть протидіяти.

1.3. Принцип цінності у якості стимулу для користувачів

Принцип цінності міститься у наступному. В ТБЧ стимули одинакові для всіх зацікавлених сторін. Так bitcoin або інша одиниця цінності є невід'ємною частиною цього вирівнювання та кореляції репутації [2, 3, 9 – 12]. Сатоші розробив програмне забезпечення таким чином, щоб винагородити тих, хто працює і належать до них, були більшими.

1.4. Принципи захисту (безпеки) інформації та ресурсів

Принципи захисту (безпеки) інформації та ресурсів [2, 3, 8 – 12] повинні застосовуватись безумовно. Основним принципом (механізмом) захисту інформації та ресурсів у ТБЧ є застосування криптографічних методів – симетричних та асиметричних криптографічних перетворень та криптопротоколів. Причому, основним призначенням криптографічних механі-

змів є забезпечення цілісності, справжності, неспростовності, доступності та, в певному змісті, криптоживучості ключів та технології БЧ у цілому. Усі користувачі повинні застосовувати такі криптоперетворення як ґешування, ЦП та асиметричне шифрування і протоколи встановлення та управління ключами.

На наш погляд, вказаного механізму для забезпечення кібербезпеки в ТБЧ недостатньо. В моделях порушника та загроз не враховуються можливі канали витоку та спеціальні атаки щодо криптографічних перетворень тощо [9, 17 – 20], а також вимоги щодо до криптографічних перетворень для постквантового періоду та їх стандартизації [ ].

Проблема, яку необхідно вирішити щодо криптографічних перетворень, у наступному. Зловмисники особливу увагу в своїх зловмисних діях звертають на крадіжку особистих даних, шахрайство, кібер-залякування, фішинг, спам, шкідливе програмне забезпечення, вимагання – все це підриває безпеку інформації та особистості в суспільстві. В останні роки ними використовуються можливі канали витоку інформації та небезпечного впливу [12 – 13]. Недостатньо, а то і мало в ряді випадків, того, що робиться для підвищення безпеки клієнтів, установ, економічної безпеки та ІТ і ТБЧ.

1.5. Принципи приватності інформації та ресурсів

Принцип приватності інформації та ресурсів у наступному [2, 3, 8 – 10]. Особа повинна контролювати свої власні дані. Причому особи повинні мати право вирішувати, що, коли, як і скільки стосовно своєї особистості вони повинні повідомляти кому-небудь іншому. Повага до права на приватність – це не одне й те ж, що повага до приватності. Необхідно забезпечити і те, і інше. Усуваючи необхідність довіряти іншим, Сатоші усунув необхідність знати справжність особистості, щоб взаємодіяти з нею.

Проблема, яку необхідно вирішити у наступному. Визнано, що приватність – це основне право людини і фундамент вільного суспільства. Нині існує порушення конфіденційності, коли спочатку збираються і використовуються наші дані без згоди або дозволу особи, а потім інформація надійно не захищається від хакерів.

Наслідки для економіки БЧ критичні. Безумовно, що БЧ надає можливості зупинити тиск та спостереження за суспільством. Це, наприклад, для корпорацій дозволяє мати повну інформацію про особу. Швидко з'являються особисті дані про здоров'я та фітнес, щоденні поїздки, життя всередині будинків тощо.

Завдяки технології БЧ можна володіти своїми особистими правами як у віртуальному світі Second Life, але з реальними наслідками. Віртуальний користувач може захистити свою особисту інформацію, роздаючи лише інформацію, необхідну для будь-якої соціально-економічної роботи. При цьому можна переконатись, що отримує особа компенсацію за будь-які дані, які мають значення для іншої сторони.

1.6. Особливості блокчейн, які повинні бути враховані при аналізі захищеності

Розглянемо деякі особливості БЧ, що пов’язані з його рекламою щодо переваг при застосуванні тощо [3, 8 – 15].

Важливою особливістю є необґрунтованість застосування ТБЧ. Існує тенденція до непомірного пропагування щодо використання ТБЧ, яка нині розвивається. Виконано значне число проектів, що рекламуються та впроваджуються [2, 3]. Іноді є намагання включати ТБЧ, навіть, якщо вона не зовсім потрібна чи не потрібна. Вказане пов'язано з тим, що технологія ТБЧ є відносно новою і не дуже зрозумілою. Нижче розглянемо деякі обмеження і, можливо, необґрунтовані погляди, що пов'язані з технологією БЧ.

Незмінність технології. Більшість джерел, що стосуються ТБЧ, описують регістри ланцюга блоків як незмінні. Насправді, це не зовсім так. Вказане можна пояснити тим, що вони захищені від несанкціонованого доступу, внаслідок чого є довіра, наприклад через фінансові транзакції. Але ТБЧ не можуть вважатися повністю незмінними, тому що є ситуації, в яких ТБЧ може бути зміненою. Тому вкажемо на способи, за допомогою яких може бути порушена концепція незмінності щодо реєстрів БЧ.

1.7. Особливості забезпечення кібербезпеки в блокчейн

Використання технологій БЧ не усуває властиві для кібербезпеки ризики [2, 3, 8 – 11], що вимагають продуманого та активного управління ними. Скоріше всього, більшість з них пов'язані з людським фактором. Тому повинна бути розроблена та використовуватись надійна програма кібербезпеки ТБЧ, захисту мережі та організацій – учасників від кіберзагроз. Це повинне бути зроблене з урахуванням того, що хакери отримують все більше знань про ТБЧ та їх вразливості.

Також існуючі стандарти і керівництва в області кібербезпеки, як і раніше, мають велике значення для забезпечення безпеки систем, що взаємодіють та/або покладаються на ТБЧ. За умови певних коригувань, для розгляду конкретних характеристик ТБЧ існуючі стандарти і рекомендації забезпечують значну основу для захисту мереж БЧ від кібератак.

На додаток до загальних принципів і засобів контролю також прийнятні конкретні стандарти кібербезпеки, що мають відношення до ТБЧ, які вже існують і широко використовуються в багатьох галузях, наприклад NIST Cybersecurity Framework [9]. В ньому стверджується, що не існує «єдиного підходу до усунення загроз кібербезпеки», через те, що «організації будуть і далі мати унікальні ризики: різноманітні загрози, вразливості, схильності до ризиків, а відповідно, спосіб практичного використання ТБЧ в рамках структури буде різним.

1.8. Особливості застосування інфраструктури відкритого ключа та ідентифікація

Якщо ТБЧ включає інфраструктуру відкритого ключа (ІВК), то деякі користувачі відразу ж думають, що при її використанні здійснюється і ідентифікація особистості [9, 13 – 16]. Але це не зовсім так, оскільки відносин «один на один» з використанням особистих ключів може і не існувати (користувач може мати декілька особистих ключів), також як і може не бути відносин «один на один» між адресами в ТБЧ і відкритими ключами (множинні адреси можуть випливати з єдиного відкритого ключа).

Часто ЦП використовуються для підтвердження ідентичності у цифровому світі при забезпеченні кібербезпеки, що може призвести до плутанини відносно потенційного застосування ТБЧ для управління ідентифікацією. Процес перевірки ЕП транзакції в блокчейні пов'язує транзакції з власниками особистих ключів, але не надає можливості для зв'язування реальних ідентифікаторів з цими власниками. У деяких випадках можна пов'язувати ідентифікатори реального цифрового світу з особистими ключами, але ці зв’язки виконуються через зовнішні процеси, а не підтримуються БЧ напряму. Наприклад, правоохоронні органи можуть запитувати записи з біржі, які пов'язують транзакції з конкретними особами. Іншим прикладом є індивідуальна публікація криптовалютної адреси на своєму особистому вебсайті або сторінці, наприклад для пожертвувань в соціальній мережі, що забезпечить зв'язок між адресою та ідентифікатором у реальному цифровому світі.

Хоча, ТБЧ можна використовувати в інфраструктурах управління ідентифікацією, для яких потрібен розподілений компонент реєстру, важливо розуміти, що типові реалізації БЧ не призначені для автономних систем управління ідентифікацією. Існує більше можливостей мати надійні цифрові ідентифікатори, ніж просто реалізувати ТБЧ.

Таким чином, як при аналізі, так і в процесі синтезу ТБЧ важливими є основоположні принципи, що визначають сутність, вимоги та можливості ТБЧ.

2. Особливості та умови застосування ТБЧ

Як зазначалося вище, ТБЧ мають свої особливості та вразливості. Серед особливостей, що стосуються безпеки ТБЧ, слід відмітити такі [11 – 13]: складність системи, розміри мережі, швидкість і ефективність мережі, політика використання, зловмисні користувачі, відсутність довіри, ігнорування транзакцій конкретних користувачів, вузлів або навіть цілих держав, створення зміненого, альтернативного ланцюга щодо таємниці. Розглянемо та проаналізуємо їх.

Складність системи. Якщо вирішено створити систему на основі ТБЧ з нуля, то одна невелика помилка може стати фатальною та зруйнувати всю розробку. Звичайно ж, це не можна вважати недоліком самого БЧ – це, скоріше, стосується особливостей його використання. Розробник, що створює блокчейн або займається його розробкою та підтриманням дієздатності, має бути дуже досвідченим, бо вірогідність допустити помилку у такій складній системі підвищується. Це можна побачити у сфері криптовалют, де регулярно відбуваються викрадення криптовалюти у користувачів або загалом компрометація усієї мережі навіть у найбільших проектах [20].

Розміри мережі. Для роботи БЧ необхідні як мінімум кілька сотень, а ще краще кілька тисяч узгоджено працюючих вузлів. Саме через це ТБЧ є вкрай вразливою до атак на початкових етапах роботи. Наприклад, якщо який-небудь користувач зможе отримати контроль над 51 % вузлів системи, то він зможе повністю контролювати створення блоків у мережі БЧ. А якщо в системі всього 20 вузлів, то подібний варіант розвитку подій більш, ніж можливий. Проте навіть ця атака вже враховується при побудуванні сучасних мереж БЧ та нівелюється за допомогою більш досконалих систем консенсусу.

Швидкість і ефективність мережі. Структура мереж БЧ – це також одна з причин, з використанням якої може бути порушено нормальне функціонування мережі БЧ. Так, якщо мережа БЧ отримає надто широке поширення, а інфраструктура БЧ виявиться не готовою до такого обсягу операцій, то в результаті може знизитися швидкість проведення транзакцій, можуть з'явитися проблеми зі зберіганням даних, що негативно вплине на ефективність мережі БЧ. Також, до цієї проблеми можна віднести ситуацію, коли кількість користувачів (транзакцій) в системі буде перевищувати теоретичну максимально можливу кількість транзакцій в мережі (транзакцій в секунду), що може призвести до довгого очікування підтвердження транзакцій.

Політика використання. З огляду на те, що валюта в мережі БЧ є міжнародною і децентралізованою, це, по суті є загрозою для контрольованих державою валют. На даний момент керівні органи деяких держав прагнуть ввести більш суворі обмеження на використання БЧ. У різних країнах сподіваються взяти систему під контроль до того, як вона стане серйозним конкурентом і почне загрожувати їхній економіці. Непрямим чином це також є суттєвою загрозою для сучасної банківської системи. Наразі ця загроза є вкрай реальною, наприклад, за неофіційними даними Китай контролює близько 60 % обчислювальних потужностей біткоіну та потенційно може його скомпрометувати [9].

Застосування ключів. В транзакціях в мережі БЧ використовуються асиметричні пари ключів – публічні і особисті криптографічні ключі. Самі по собі такі ключі зламати майже неможливо на звичайному комп'ютері, проте зловмисник може отримати їх більш простим і звичним способом. Наприклад, ключі можна дістати в тому випадку, якщо ви зберігаєте їх на небезпечній або незахищеній платформі. Використання соціальної інженерії та викрадення ключів із менш захищених ресурсів є найбільш популярним способом на сьогоднішній день серед криптовалютних зловмисників [1 – 6, 9, 19].

Зловмисні користувачі. Мережа БЧ не може нав'язувати правила та інструкції з проведення транзакцій, вона не може диктувати користувачам норми поведінки. Тому це проблематично для permissionless мереж БЧ через те, що користувачі виступають під псевдонімами і немає відповідності «один до одного» між ідентифікаторами користувачів мережі і іменами користувачів системи. Permissionless мережі часто надають винагороду (наприклад, криптовалюту), щоб мотивувати користувачів діяти справедливо; проте деякі можуть вибирати зловмисну поведінку, якщо це дає більшу винагороду. Найбільша проблема для зловмисних користувачів – отримати достатню потужність (чи то ставку в системі, обчислювальну потужність тощо), щоб завдати шкоди.

Аналіз показав, що якщо створюється досить велике зловмисне співтовариство, злочинні дії можуть зводитись до наступних дій та поведінки [1 – 6, 9].

Ігнорування транзакцій конкретних користувачів, вузлів або навіть цілих держав.

Створення зміненого, альтернативного ланцюга таємниці, а потім його відправка, як тільки альтернативний ланцюг довший реально побудованого. Чесні вузли будуть перемикатися на ланцюг, який має найбільшу «роботу» (за протоколом БЧ). Це може порушити основний принцип мережі БЧ – порушити її прозорість до підробки та захист від несанкціонованого використання [3].

Відмова передавати блоки на інші вузли. По суті, порушуючи розподіл інформації (це не є проблемою, якщо мережа БЧ є досить децентралізованою). У той час як зловмисні користувачі можуть створювати неприємності і наносити короткочасну шкоду, мережі БЧ можуть виконувати жорсткі розгалуження для боротьби з ними. Чи будуть відшкодовані збитки (втрачені гроші), залежить від розробників і користувачів мережі БЧ [9].

На додаток до наявності зловмисних користувачів в мережах, адміністратори інфраструктури БЧ в permissioned мережах можуть також діяти зловмисно. Наприклад, адміністратор інфраструктури може (в залежності від точної конфігурації) мати можливість захопити виробництво блоків, виключати деяких користувачів з виконання транзакцій, переписувати історію блоків, двічі витрачати монету, видаляти ресурси, переадресовувати чи блокувати мережеві підключення [9, 19].

Відсутність довіри. Інша поширена невірна інтерпретація щодо роботи БЧ може надходити від осіб, які знають, що в БЧ немає «третьої довіреної сторони», а мережі БЧ є «не довіреними» середовищами. Не дивлячись на те, що третя довірена сторона не сертифікує транзакції в permissionless мережах блокчейн (в permissioned системах це менш помітно, оскільки адміністратори системи діють, як довірені особи, надаючи користувачам допуски і дозволи). Для правильного функціонування мережі БЧ все ж необхідний досить високий ступінь довіри всередині мережі, в тому числі коли [2 – 6, 9].

1) Існує довіра до криптографічної технології, що використовується, але криптографічні алгоритми або їх реалізації можуть мати недоліки.

2) Існує довіра до правильної і безперервної роботи смарт-контрактів, які можуть мати ненавмисні лазівки і недоліки.

3) Існує довіра до розробників, які виробляють програмне забезпечення якомога більш стабільним.

4) Існує впевненість в тому, що більшість користувачів блокчейну не вступають у таємну змову. Якщо окрема група або фізична особа може керувати більш, ніж 50 % всієї потужності створення блоків, виникає можливість підірвати permissionless мережу блокчейн. Однак, як правило, отримання необхідної обчислювальної потужності є надмірно дорогим.

5) Для користувачів мережі БЧ , які не мають повного вузла, існує довіра до того, що вузли приймають і обробляють транзакції справедливо.

2. Опис та аналіз потенційних атак, коли застосування БЧ є суттєвим механізмом захисту від них

Аналіз показав[9 – 12], що проблема безпеки інформації та безпеки взагалі ІТ, стоїть перед сучасним цифровим світом досить гостро. Зростає кількість кіберзагроз, що пов'язані з крадіжкою ідентифікаційних даних. За інформацією аналітичного агентства Cybersecurity Venture, щорічний збиток від кіберзлочинів досягне до 2021 р. порядку шість трильйонів доларів. У 2017 р. збиток становив чотири трильйони. Тому зростає і кількість коштів, вкладених в кібербезпеку, – до 2021 р. витрати перевищать один трильйон доларів. Це пояснюється тим, що ТБЧ при їх застосуванні безпосередньо, а також у інших ІТ забезпечує захист від цілого спектру різних атак. Розглянемо та проаналізуємо основні із них [10 – 13, 21, 22].

3.1. Атаки типу «людина посередині»

Нині для захисту з'єднань (наприклад, HTTPS і TLS) [23], тобто каналів зв’язку, використовується ІВК, що включає засвідчувальні центри (ЗС) та центри сертифікації ключів(ЦСК). Кожен учасник мережі має пару відкритий/особистий ключ. Особистий ключ він зберігає в таємниці. Відкритий ключ зберігає ЦС. Коли користувач хоче встановити безпечне з'єднання

(зайти на сайт), він відправляє запит на відкритий ключ ресурсу у сертифікаційного центру і шифрує дані перед відправкою, використовуючи свій особистий ключ та відкритий із сертифікату. Щоб розшифрувати дані, сервер (сайт) використовує свій особистий ключ та відкритий відправника.

У цьому випадку криптостійкість (надійність) системи залежить від того, наскільки добре захищений засвідчувальний центр. Якщо зловмиснику вдається компрометувати засвідчувальний центр, то він отримає можливість провести атаку man-in-the-middle – MITM («людина посередині»). У цьому випадку виконується розсилка підроблених відкритих ключів, до яких у хакерів є відповідні особисті (із пари фальшивих) ключі. З їх допомогою виконується розшифрування інформації, що передається від клієнта до серверу чи навпаки.

Однак, проведений аналіз показав, що в системі, що побудована на блокчейні, MITM атаку не можна реалізувати. Коли користувач публікує відкритий ключ в БЧ, про це «дізнаються» всі вузли мережі (наприклад, БЧ біткоіна має 10 тисяч активних вузлів). Ця інформація записується в блок, і цілісність реєстру захищається криптографічно. Тому опублікувати підроблені ключі зловмисники з великою ймовірністю не зможуть, таку спробу відразу розпізнають користувачі та не приймуть вузли мережі БЧ. Це однак із основних переваг ТБЧ.

3.2. Маніпулювання даними

Важливою вимогою до даних ІТ є коректність даних у мережі. Наприклад, якщо завантажується файл із Інтернету та для того, щоб перевірити його цілісність та автентичність, як правило використовується його ґеш-значення. Так як в ІТ «клієнт – сервер» файл і дані про його ґеш-значення зберігаються на сервері централізовано, то зростає вірогідність підробки цих даних[16 – 19]. Навіть використання простої перевірки ґеш-значення, що міститься на офіційному ресурсі цього файлу, несе в собі елемент довіри користувача до цього ресурсу. Наприклад, відомі випадки, коли подібний офіційний ресурс атакували та робили підміну ґеш – значення на своє. У результаті користувач, що хоче перевірити ґеш-значення файлу зловмисника потрапить на сайт, що знаходиться під контролем того самого зловмисника, пройде перевірку та інсталює шкідливе програмне забезпечення.

Блокчейн у свою чергу дозволяє записати ґеш-значення у БЧ та бути з великою вірогідністю певним, що воно залишиться незмінним, а вірогідність підробки його з часом буде майже нульовою, оскільки це буде вимагати також підробки усіх наступних блоків БЧ, що є майже неможливою задачею для сучасних комп'ютерних систем.

3.3. DDoS-атаки

Завданням розподілених мережевих атак є обмеження пропускної здатності мережевого ресурсу, наприклад інфраструктури, що підтримує сайт компанії чи системи «клієнт – сервер» [12 – 14]. Так, веб-сервери завжди мають обмеження за кількістю запитів, що оброблюються одночасно (пропускна здатність). Якщо число звернень до сервера перевищує можливості будь-якого компонента інфраструктури, виникають проблеми з рівнем обслуговування. Причому масштаб цих проблем залежить від мети DDoS-атаки.

Наприклад, масована DDoS-атака на американського DNS-провайдера Dyn залишила мільйони користувачів без таких сервісів, як Twitter, PayPal, Netflix і GitHub[ ]. DDoS-атака на Dyn проводилася за допомогою гігантського ботнету Mirai, що включав десятки мільйонів пристроїв: роутери, принтери, IP-камери і інші пристрої, підключені до Інтернету. Всі разом вони транслювали дані на сервери Dyn зі швидкістю 1,2 Тбіт/с. А в жовтні цього року почав поширюватися вірус Reaper, що заражає «розумні» пристрої по всьому світу.

Атака на DNS-провайдер показує, наскільки централізовані системи роблять всю інтернет-інфраструктуру вразливою. Більш серйозним сценарієм розвитку атак на DNS-сервери є його компрометація з метою перенаправлення користувачів на сайти із шкідливим програмним забезпеченням.

Однак можна відмовитися від центральних DNS-серверів і реалізувати систему, в якій пари «ім'я – IP-адреса» реєструються в мережі БЧ та розподіляються по всіх вузлах. Це

забезпечить прозорість перетворень та і захищеність одночасно. Зловмисники не зможуть зруйнувати якусь одну певну інфраструктуру, атакувавши лише один із кластерів. Самі дані будуть захищені засобом застосування ЕП, ґешування та криптографічних протоколів. Застосування БЧ також суттєво зменшить мережеві витрати, пов'язані з читанням DNS.

3.4. Захист пристроїв «інтернету речей»

Згідно з результатами дослідження компанії F5 Networks, число атак на пристрої інтернет («інтернет речей», Internet of Things) та їх інфраструктуру зросло з початку 2018 р. на 280 % [3, 19]. Здебільшого це пов'язано з поширенням зловмисниками шкідливого програмного забезпечення. В цьому випадку Зловмисники атакують інтернет-пристрої та використовують їх для проведення DDoS-атак і хостингу інфраструктури різних вірусів [19, 22].

Вважається, що застосування БЧ дозволить захистити інтернет з тих же причин, за яких він використовується в криптовалютах – впевненість в легітимності даних і чіткий процес їх підтвердження. Однак необхідно враховувати, що простої реєстрації пристрою в БЧ – недостатньо. Необхідна ціла інфраструктура для управління пристроями і контролю доступу до даних. Проведений аналіз показав, що вже існують декілька рішень від публічних проектів, що реалізують цю систему. Наприклад, одним з рішень може стати проект ChainAnchor [9]. Це фреймворк, який будуть підтримувати розробники «розумних пристроїв», провайдери даних і незалежні розробники. Ідея полягає в тому, що учасники мережі, в обмін на підтримування безпеки, отримають можливість продавати анонімні дані з інтернет-пристроїв. Фреймворк має механізми, що дозволяють блокувати компрометовані пристрої, а також відключати від БЧ легітимні пристрої при зміни власника. Він також дозволяє працювати в умовах необмеженого зростання даних [5, 9].

3.5. Кібер та мережеві атаки

Нині ТБЧ [5, 9 – 12, 22, 23] рекламуються як дуже безпечні. Це можна пояснити в в силу захищеності від несанкціонованого доступу (НСД) – як тільки транзакція у блокчейні здійснена, її взагалі не можна змінити. Однак це справедливо тільки для транзакцій, які були включені в опублікований блок. Транзакції, які ще не були включені в опублікований блок в ланцюгу, вразливі для декількох типів атак. Так у мережах БЧ, які мають мітки часу для транзакцій, підроблення часу або зміна годин користувача могло б мати негативний або позитивний вплив на транзакцію, роблячи час і повідомлення часу вектором атаки. Відмова в обслуговуванні, коли атака може проводитися на платформі БЧ або на інтелектуальному контракті, реалізованому на платформі.

Мережі БЧ та їх застосування не захищені від зловмисних учасників, які можуть проводити мережеве сканування і розвідку, щоб виявляти і використовувати вразливості, а також запускати атаки нульового дня. Служби, засновані на БЧ, та розгорнуті в поспіху, або тільки закодовані додатки (такі, як інтелектуальні контракти) можуть містити нові, а також відомі вразливості і слабкості розгортання. Вони і будуть знайдені і потім атаковані через мережу, так само, як атакують сайти і додатки в теперішній час.

3.6. Огляд та класифікація атак спеціального типу

Атаки спеціального виду (side-chanel attacks) можна віднести до атак аналітичного типу. Реалізація цих атак направлена на пошук вразливостей у практичній реалізації криптосистеми, в першу чергу засобу криптозахисту інформації (КЗІ). У [23] запропоновано класифікацію спеціальних атак за такими ознаками:

- контроль над обчислювальним процесом;
- спосіб доступу до системи чи засобу;
- метод безпосереднього здійснення атаки.

На рис. 1. наведено модель, яка пояснює атаки спеціального виду [23].

Класифікація спеціальних атак за степенем впливу на обчислювальний процес. Аналіз показав [23], що за степенем впливу на обчислювальний процес спеціальні атаки можна поділити:

- на пасивні, коли зловмисник отримує необхідну інформацію без помітного впливу на систему, але система при цьому продовжує функціонувати як і раніше;

- активні, коли зловмисник реалізує деякий вплив на систему, у результаті якого змінюється поведінка системи, але зміни такого роду можуть бути «прозорими» для системи, на яку відбувається напад.

При цьому зловмисник у змозі визначати та використати інформацію про систему БЧ.

Класифікація спеціальних атак по способу доступу до системи. В залежності від можливості доступу до апаратно-програмного чи апаратного засобу КЗІ можна виділити такі класи атак [36]:

- агресивні (англ. invasive) – коли здійснюється спроба розкриття системи зловмисником та отримання прямого доступу до внутрішніх компонентів;

- напівагресивні (англ. semi-invasive) – коли вплив на внутрішні компоненти засобу КЗІ здійснюється без посереднього контакту;

- не агресивні (англ. non-invasive) – коли використовується тільки зовнішня інформація – наприклад, час обчислення чи споживання енергії. Тобто безпосереднього впливу на систему, що досліджується, немає.

Класифікація спеціальних атак за методом здійснення атаки. Спеціальні атаки, в залежності від методів, які використовуються для аналізу отриманої інформації, можна поділити [23]:

- на прості (simple side channel attack) – коли здійснюється дослідження прямої залежності між процесами в пристрої та отриманої зловмисником інформації, а результатом атаки є виділення корисної інформації, наприклад, від рівня шумів;

- диференційні (differential side channel attack) – коли використовуються статистичні методи дослідження залежностей між вхідними даними та інформацією, яка отримана під час спостереження.

Як правило, при цьому здійснюється велика кількість вимірювань, а також спеціальна обробка сигналу і корекція помилок. В процесі здійснення атак на реалізацію засобу КЗІ може здійснюватись аналіз усіх зовнішніх параметрів засобу, а також усі можливі методи порушення його нормального функціонування, аж до його руйнування з метою отримання секретного ключа.

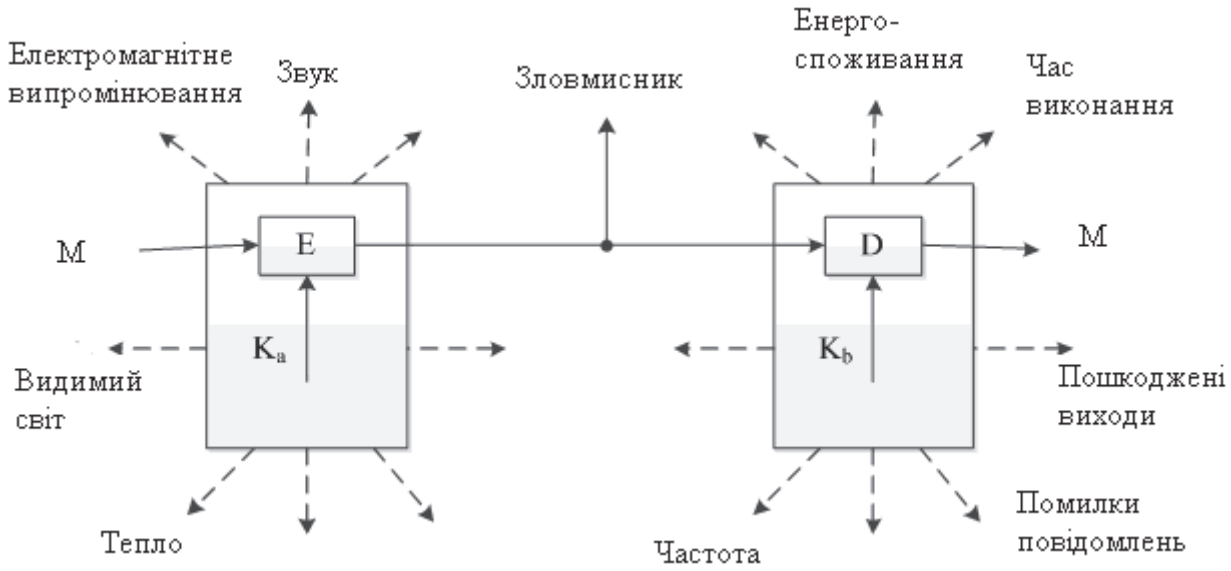


Рис. 1. Криптографічна модель відносно атак спеціального виду

При виконанні атак за часом [23] вимірюється час виконання алгоритму криптоперетворення. У реалізаціях асиметричних алгоритмів час виконання операцій також може залежати як від оброблюваних даних, так і від ключа криптоперетворення (ЕП, АСШ, ПІК). При використанні апаратного рішення у вигляді автомата з жорсткою логікою навіть час складання за деяким модулем може змінюватися у залежності від реалізації ланцюгів перенесення.

Атаки на реалізацію можуть ґрунтуватись на аналізі всіх споживаних потужностей сучасних обчислювальних пристроїв КЗІ, особливо таких, що побудовані на використанні елементів схемотехніки ТТЛ (TTL), ТТЛШ (TTL(S)), а також частково і КМОП (CMOS). Вона також залежить від оброблюваних даних. Тому у зловмисника з'являється можливість отримати інформацію про внутрішній стан автомату, у тому числі секретний ключ, наприклад шляхом аналізу енергоспоживання при АСШ чи ЦП. Так, атака, що описана у [23], дозволяє на основі аналізу енергоспоживання обчислити вагу Хемінга (кількість одиничних бітів) оброблюваного блоку. Ця інформація, а також знання виключно відкритих текстів (без знання шифротексту), дає зловмисникові можливість відтворити таємний ключ шифрування. Крім того, якщо у порушника є можливість порушувати нормальну роботу пристрою (наприклад, вносити збої), то за допомогою спеціальних методів можна відновити практично будь-який секретний параметр системи.

Основною метою фізичної атаки є дослідження особливостей реалізації пристрою КЗІ (мікросхеми), що потрібно для отримання інформації відносно особистого або таємного ключів, наприклад, шляхом дослідження області всередині кристалу ПЛІС. Як правило, такі атаки орієнтовані на специфічні області ПЛІС, які в режимі нормального функціонування є не доступними.

4. Сутність та пропозиції відносно протидії атакам спеціального виду

Наші попередні дослідження показали, що існуючі криптографічні механізми, що застосовуються в ТБЧ, не забезпечують захист від атак спеціального виду. Для забезпечення захисту потрібно застосовувати, як мінімум, постквантову криптографію – асиметричні перетворення типу ЕП, АСШ та криптографічні протоколи інкапсуляції ключів. Детальніше ці питання висвітлені в [9, 17 – 20, 23]. Відмітимо, що в основу захисту від атак спеціального виду можуть бути покладеними методи, що розглядаються нижче.

4.1. Фіксована кількість звернень до ґеш-функції

В [23] розглянуто атаку спеціального виду за часом, яка може розкрити секретний ключ NTRU. Ця атака можлива завдяки тому, що у розшифруванні різних шифротекстів використовується різна кількість звернень до ґеш-функції. Методом протидії таким атакам є використання механізму доповнення. Розмір доповнення повинен відповідати необхідному рівню криптостійкості. В цьому випадку використовується схема доповнення NAEP, а розмір доповнення дорівнює розміру ґеш-значення, яке задовольняє умові

$$Hlen = \begin{cases} 160 & k \le 112 \\ 256 & k > 112 \end{cases} \qquad (1)$$

де k – рівень криптостійкості.

За умови виконання (1) можна сподіватись, що криптоперетворення і, як наслідок, криптосистема, може бути захищеною від атак за часом.

4.2. Рандомізація даних

Метод рандомізації зводиться до «засліплення» даних [24 – 28]. По суті, воно зводиться до зміни вхідних даних в деякий непередбачуваний стан. Залежно від характеристик функції «засліплення», вона може виключити деякі або всі витоки корисної інформації. Основною властивістю вхідних даних є їх псевдовипадковість. У криптосистемі «NTRU Prime IIT

Ukraine»[25] застосовується засліплюючий поліном, що запобігає витоку інформації про секретний ключ.

4.3. Незалежність від значень

Якщо усі перетворення із особистим ключем та поліномом засліплення при зашифруванні та розшифруванні не залежать від значень засліплюючого поліному та особистого ключа, то про них неможливо по стороннім каналам дізнатися будь-яку інформацію. Також, якщо в операції множення не використовується значення секретного ключа, то неможливо отримати інформацію про секретний ключ, аналізуючи операцію множення по стороннім каналам.

4.4. Вплив заходів стійкості на кількість ключів NTRU-подібного алгоритму

Аналіз показав, що в будь-якому разі ключі криптоперетворення повинні задовольняти властивостям випадкових послідовностей. До таких властивостей належать: випадковість, рівноймовірність та незалежність. В «NTRU Prime IIT Ukraine» [25 – 27] це забезпечується за рахунок фіксованих значень кількостей ненульових елементів у секретних ключах f та g. Так, кількість 1, -1, 0 приблизно є рівною.

У табл. 1 – 3 у якості прикладу наведено конкретні значення параметрів для першого, середнього та останнього набору параметрів згідно [26 – 27]. У [25, 26] визначено наступне співвідношення (1, -1, 0) для секретних ключів f та g: для f кількість 1 та -1 позначається як df та дорівнює df=2t, для g кількість одиниць дорівнює dg1=n/3+1, кількість -1 дорівнює dg–1=n/3.

Таблиця 1

Приклади параметрів NTRU Prime IIT Ukraine

Параметри				
n	q	t	рівень стійкості k	
439	6833	142	112	1
727	5827	121	205	2
1021	8819	183	298	3

Таблиця 2

Стійкість NTRU Prime IIT Ukraine

Число ненульових елементів	Рівень стійкості		
	1	2	3
df=2t	184	242	366
dg1=n/3+1	147	243	341
dg–1=n/3	146	242	340

У табл. 3 наведено значення кількості можливих ключів, які отримані при застосуванні формули (1).

Таблиця 3

Кількість ключів NTRU Prime IIT Ukraine

Число ненульових елементів	Рівень стійкості		
	1	2	3
для f	$0,3*10^{193}$	$0,9*10^{344}$	$0,3*10^{482}$
для g	$0,5*10^{207}$	$0,9*10^{344}$	$0,1*10^{485}$

Якщо немає обмеження на кількість 1, -1 для ключів, наприклад, як для схеми Crystals-Kyber то для підрахунку треба використовувати формулу (2) розміщення з повторенням:

$$A'^{m}_{n} = n^{m}, \tag{2}$$

де n – для ключів це кількість елементів, тобто 3, а m – кількість позицій, тобто розмір ключа.

У табл. 4 наведено значення кількості секретних ключів при відсутності обмежень на кількість коефіцієнтів.

Таблиця 4

Кількості секретних ключів без обмеження на кількість коефіцієнтів

	Рівень стійкості		
	1	2	3
Кількість секретних ключів	$0,3*10^{210}$	$0,7*10^{347}$	$0,1*10^{488}$

Аналіз показав, що при введенні обмежень розмір простору ключів зменшується. У табл. 5 наведено значення, у скільки разів зменшується кількість ключів, якщо ввести обмеження на коефіцієнти згідно з наведеним вище.

Таблиця 5

Зменшення розміру ключового простору

Число ненульових елементів	Рівень стійкості		
	1	2	3
для f	10^{17}	$0,8*10^{3}$	$0,3*10^{6}$
для g	$0,6*10^{3}$	$0,8*10^{3}$	10^{3}

Таким чином, обмеження на кількість ненульових коефіцієнтів призводить до зменшення кількості ключів від 17-ти до 3-х десяткових порядків. Однак ця міра є необхідною задля захисту перспективних криптоперетворень постквантового періоду від атак по стороннім каналам. Але, для реалізації наведених пропозицій щодо захисту від спеціальних атак потрібно використовувати принципово нові криптографічні механізми, скоріше всього постквантового періоду [17, 18, 26 – 28].

Висновки

1. Впровадження у різноманітні інформаційні технології принципів децентралізації має суттєві перспективи. Основні з них закладені в технології БЧ. Але, незважаючи на появу та впровадження на основі ТБЧ перспективних широкомасштабних розробок, продовжують існувати певні сумніви відносно перспектив застосування ТБЧ. Вони, як і більшість нових інформаційних технологій, можуть бути обмеженими в застосуванні, а то і непотрібними .

2. Відносно застосування БЧ важливими є відповіді на питання – чи можна та яким чином удосконалити системи «клієнт – сервер». При цьому першим питання, на яке потрібно відповісти, це як оцінити покращення. Відповіді на дане питання можуть бути для звичайних користувачів та суспільства достатньо простими – нові технології повинні давати принципові та істотні покращення, в порівнянні з тими, що уже існують.

3. Важливим є такий фактор, як людська звичка. Суспільство та суб'єкти) переходять на нові технологій , тільки в разі, якщо вони дають суттєву перевагу, не просто на 5 – 10 %, а мінімум в 2-3 рази. При цьому, як підтверджено практикою, що для нових ІТ надважливими є такі критерії та показники оцінки та порівняння як вартість, складність (часова та просторова), швидкість, прибутковість, безпечність, анонімність, гнучкість, дизайн тощо.

4. Нова технологія, в нашому випадку ТБЧ, не повинна істотно програвати існуючим технологіям за іншими параметрами. Наприклад, якщо ТБЧ працює в три рази швидше, але якщо вона при цьому в два рази дорожче і в 1,5 рази складніше – вона, скоріше всього, не отримає визнання та застосування. В якості прийнятного порогу програшу можна взяти біля 30 – 50 %. У цілому нова технологія повинна бути в чомусь краще в 2-3 рази, а за всіма іншими параметрами програвати не більше ніж в 1,5 рази. Тобто, покращення повинне давати суттєві переваги, але ще і компенсувати побічні ефекти щодо програшу за іншими параметрами.

5. З метою реалізації системного підходу до аналізу та оцінки захищеності розглянемо визнані основні принципи та вимоги щодо побудування ТБЧ. При побудуванні ТБЧ повинні бути застосовані чи рекомендовані до застосування такі базові принципи як: мережевої цілісності; розподілення влади; цінності як стимулу для користувачів; захисту (безпеки) інформації та ресурсів; приватності інформації та ресурсів. Важливими також є принципи створення програмного забезпечення; послуг технологій; бізнес моделей та ринків; організація функціонування; при необхідності також управління БЧ тощо.

6. Серед особливостей, що стосуються безпеки ТБЧ, слід відмітити: складність системи, розміри мережі, швидкість і ефективність мережі, політика використання, зловмисні користувачі, відсутність довіри, ігнорування транзакцій конкретних користувачів, вузлів або навіть цілих держав, створення зміненого, альтернативного ланцюга щодо таємниці.

7. Атаки спеціального виду (side-chanel attacks) можна віднести до атак аналітичного типу. Реалізація цих атак направлена на пошук вразливостей у практичній реалізації криптосистеми, в першу чергу засобу криптозахисту інформації (КЗІ). У [23] запропоновано класифікацію спеціальних атак за такими ознаками:

- контроль над обчислювальним процесом;
- спосіб доступу до системи чи засобу;
- метод безпосереднього здійснення атаки.

8. Для захисту криптосистеми «NTRU Prime IIT Ukraine» від атак за часом пропонується під час шифрування здійснювати фіксовану кількість звернень до ґеш-функції, а також здійснювати засліплення даних (що вносить додаткову випадковість). Також усі перетворення, що здійснюються з секретними параметрами, не повинні залежати від конкретних значень цих параметрів.

9. Якщо усі перетворення із особистим ключем та поліномом засліплення при зашифруванні та розшифрування не залежать від значень засліплюючого поліному та особистого ключа, то про них неможливо по стороннім каналам дізнатися будь-яку інформацію. Також, якщо в операції множення не використовується значення секретного ключа, то неможливо отримати інформацію про секретний ключ аналізуючи операцію множення по стороннім каналам.

10. Для забезпечення захисту потрібно застосовувати як мінімум постквантову криптографію – асиметричні перетворення типу ЕП, АСШ та криптографічні протоколи інкапсуляції ключів. Детальніше ці питання висвітлені в [9, 17 – 20, 23].

Список літератури:

1. Andreas M. Antonopoulos Mastering Bitcoin: Unlocking Digital Cryptocurrencies / Andreas M. Antonopoulos Kyiv : NGITS, 2014. – С. 10 – 150.

2. Що таке децентралізований додаток? [Електронний ресурс]. Режим доступу: https://www.coindesk.com/information/what-is-a-decentralized-application-dapp.

3.Don Tapscott, Alex Tapscott Blockchain Revolution: How the Technology Behind Bitcoin is Changing Money, Business, and the World / Don Tapscott, Alex Tapscott Blockchain. Kyiv : Information Systems, 2016. С. 65 – 102.

4. 20 основних застосувань БЧ [Електронний ресурс]. Режим доступу: https://biznesmodeli.ru/blockhain-chto-eto-cases-crypto-top10/.

5. БЧ: атаки, безопасность и криптография [Електронний ресурс]. Режим доступу: https://www.securitylab.ru/blog/personal/ Informacionnaya_bezopasnost_v_detalyah/343072.php.

6. Распределённые реестры и информационная безопасность: от чего защищает БЧ [Електронний ресурс]. Режим доступу: https://habr.com/company/bitfury/blog/341902/.

7. Клиент-сервер: https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82_%E2%80%94_%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80.

8. Коржов В. Многоуровневые системы клиент-сервер. Издательство : Открытые системы. Дата обращения 31 января 2010. Архивировано 26 августа 2011 г.

9. NISTIR 8202 – Blockchain Technology Overview, 2018, 68 p. Access mode: https://doi.org/10.6028/NIST.IR.8202.

10. Nakamoto S. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008. https://bitcoin.org/bitcoin.pdf.

11. Pavan Duggal Blockchain Contracts and Cyberlaw / Pavan Dugga. Kyiv : Information Systems, 2015. С. 15 – 39.

12. Quantum attacks on Bitcoin, and how to protect against them / Divesh Aggarwal, Gavin K. Brennen, Troy Lee, Miklos Santha, Marco Tomamichel; National University of Singapore. Singapore, 2017.

13. Закон України «Про захист інформації в інформаційно-телекомунікаційних системах» від 31.05.2005, № 2594-IV.

14. Закон України «Про основні засади забезпечення кібербезпеки України» // Відомості Верховної Ради (ВВР). 2017. № 45, ст.403.

15. Регламент (ЄС) № 910/2014 Європейського Парламенту та Ради від 23 липня 2014 року «Про електронну ідентифікацію та довірчі послуги для електронних транзакцій у межах внутрішнього ринку та про скасування Директиви 1999/93/ЄС» (1) (COM (2012) 0238-C7-0133/2012 – 2012 / 0146 (COD)).

16. Прикладна криптологія. Теорія. Практика. Застосування / І.Д. Горбенко, Ю.І. Горбенко. Харків, 2012. С. 352 – 359.

17. Lily Chen Report on Post-Quatum Cryptography. NISTIR 8105 (DRAFT) .

18. «Status Report on the First Round of the NIST Post-Quantum Cryptography Standardization Process» Gorjan Alagic та ін. https://nvlpubs.nist.gov/nistpubs/ir/2019/NIST.IR.8240.pdf.

19. Алгоритмы шифрования – основа работы криптовалют [Електронний ресурс]. Режим доступу: https://tgraph.io/Algoritmy-shifrovaniya--osnova-raboty-kriptovalyut-09-27.

20. Blockchain 3.0 – 5 лучших проектов нового поколения: https://privatfinance.com/blockchain-3-0-5-luchshih-proektov-novogo-pokoleniya/.

21. Криптографические хэш-функции [Електронний ресурс]. Режим доступу: http://bit.nmu.org.ua/ua/student/metod/cryptology.pdf.

22. Возможные атаки на функции хэширования [Електронний ресурс]. Режим доступу: https://studfiles.net/preview/2157418/page:2/.

23. Горбенко Ю. І. Методи побудування та аналізу криптографічних систем. Харків : Форт, 2015. – 959 с.

24. American National Standard X9.98-2010. Lattice-based polynomial public key encryption algorithm, Part 1: key establishment, Part 2: data encryption. 2010.

25. Daniel J. Bernstein NTRU Prime / Daniel J. Bernstein, Chitchanok Chuengsatiansup, Tanja Lange, Christine van Vredendaal // Електронний ресурс. Режим доступу: https://ntruprime.cr.yp.to/ntruprime-20160511.pdf. https://bench.cr.yp.to/results-encrypt.html.

26. Горбенко И.Д. Общие положення и анализ алгоритма направленного шифрования NTRU Prime IIT Ukraine / І.Д. Горбенко, Е.Г. Качко, М.В. Есина // Радиотехника. 2018. Вып. 193. С. 5-16

27. Gorbenko I. D. Calculation of general parameters for NTRU Prime Ukraine of 6-7 levels of stability / I. D. Gorbenko, A. N. Alekseychuk, O. G. Kachko, M. V. Yesina, I. V. Stelnik, S. O. Kandy, V. A. Bobukh, V. A. Ponomar // Telecommunications and Radio Engineering, 2019. Volume 78, Issue 4. P.327-340. DOI: 10.1615/TelecomRadEng.v78.i4.40.

28. Gorbenko I.D. Methods of building general parameters and keys for NTRU Prime Ukraine of 5th-7th levels of stability. Product form / I.D. Gorbenko,O.G. Kachko, Yu.I. Gorbenko, I.V. Stelnik, S.O. Kandyi, M.V. Yesina // Telecommunications and Radio Engineering, 2019. Volume 78, Issue 7. P. 579-594. DOI: 10.1615/TelecomRadEng.v78.i7.30.98.

АТ «Інститут інформаційних технологій»;
Харківський національний університет імені В.Н. Каразіна;
Адміністрація Державної служби спеціального зв'язку та захисту інформації України.

Надійшла до редколегії 19.09.2019

УДК 004.056.5 **DOI:10.30837/rt.2019.3.198.11**

Н.А. ПОЛУЯНЕНКО, канд. техн. наук, А.А. КУЗНЕЦОВ, д-р техн. наук

МОДЕЛИРОВАНИЕ АТАКИ ДВОЙНОЙ ТРАТЫ НА ПРОТОКОЛ КОНСЕНСУСА «PROOF OF WORK»

1. Введение

Наиболее известной атакой на протоколы консенсуса блокчейн-систем является так называемая атака двойной траты (двойного расходования, англ. Double-spending), когда нечестный участник децентрализованной системы осуществляет повторное отчуждение (продажу) одних и тех же цифровых активов (единиц криптовалюты, токенов, монет и пр.), т.е. реализует несколько незаконных платежей из одного и того же стартового состояния [1, 2]. Если между заключением сделки и оформлением передачи права собственности проходит значительный промежуток времени, тогда продавец может попытаться продать один и тот же товар несколько раз разным покупателям, получая несколько раз оплату за один и тот же актив. Наиболее актуальной задача предотвращения двойной траты становится в системах электронных платежей. Цифровые активы легко копируются, и нечестный участник может передавать их копии большому количеству клиентов. Каждый получатель может убедиться, что полученный актив полностью соответствует заявленным характеристикам, однако не будет уверен, что такой же копией не расплатились с другим участником системы.

В традиционных (централизованных) системах задачу предотвращения двойной траты решают применением административных мер, когда централизованный (которому все подчинены) узел обеспечивает контроль допустимости той или иной операции. За предотвращение двойной траты в децентрализованной распределенной системе отвечают протоколы принятия консенсуса относительно того, какую транзакцию считать истинной [1 – 3]. Этот механизм позволяет (в идеальном случае) игнорировать попытки двойного расходования одних и тех же цифровых активов.

Первым и наиболее изученным протоколом консенсуса децентрализованных систем является алгоритм «Proof of work» [3, 4]. В его основе лежит решение сложной вычислительной задачи (как правило, поиск прообраза функции криптографического хеширования). И только тот, кто первым решит эту задачу (найдет подходящий прообраз), получит право внести изменение в состояние системы [4]. Фактически это означает возможность осуществлять транзакции с отчуждением (продажей, оплатой и пр.) цифровых активов. Таким образом, задача предотвращения двойной траты состоит в исключении (или, по крайней мере, снижении вероятности) возможного формирования прообраза одним и тем же участником системы. На практике это достигается вовлечением огромного числа участников с соответствующим распределением их вычислительных возможностей по поиску прообразов криптографической функции хеширования. Дополнительно каждый участник вправе передавать права на свои активы только после некоторого числа сформированных прообразов, кратно снижая тем самым вероятность двойной траты.

Первые результаты по оценке вероятности двойной траты в децентрализованной системе Биткоин были опубликованы в оригинальной статье Сатоши Накамото [4], а также в работе Мени Розенфельда [5]. Это самые популярные и цитируемые работы в данной области. Существуют также другие публикации, которые для разных случаев уточняют и дополняют результаты, полученные С. Накамото и М. Розенфельдом:

– результаты Карлоса Пинзон и Камило Роча [6], строящие модели атак двойной траты на основе не только хешрейта (вычислительных возможностей) злоумышленника и честной сети, но учитывающие также влияние временных параметров. Уравнения, управляющие этими моделями, используют распределение вероятностей Эрланга, в отличие от работы С. Накамото, который использует распределение вероятностей Пуассона, и работы

М. Розенфельда, который использует отрицательное биномиальное распределение вероятностей;

– результаты Ковальчук [7], которые обобщают и частично развивают известные оценки, также учитывающие время подтверждения транзакции;

– работу Аззолини [8], в которой используется вероятностное логическое программирование. Как утверждается, данный метод позволяет учитывать переменную во времени скорость хеширования и переменную сложность алгоритма «Proof of work»;

– результаты Кевин Ляо, представленные в работе [9] и рассматривающие китовую атаку, в которой злоумышленник из числа меньшинства увеличивает свои шансы на успешное проведение атаки двойной траты, стимулируя майнеров подорвать согласованный протокол и вступать в сговор посредством китовых транзакций или транзакций, несущих аномально большие сборы.

Следует отметить, что известные оценки получены в результате некоторых упрощений и допущений, т.е. используемые модели, как правило, дают приближенные значения, и основная критика этих оценок состоит в их нереалистичности, оторванности от реальных процессов, протекающих в децентрализованных системах. В частности, островными неточностями и ложными допущениями в работах С. Накамото и М. Розенфельда являются:

– вероятности сформировать блок честной сетью и злоумышленником в сумме должны быть равны единицы. Однако приведенные выражения не дают ответа, какой будет результат при независимых величинах этих вероятностей [10];

– не принимается во внимание экономическая возможность по формированию блоков злоумышленником, а также экономическая целесообразность. Ресурсы злоумышленника по поддержанию гонки между злоумышленником и честной сетью считаются безграничными, что не может соответствовать действительности [11];

– предполагается, что вероятность успеха сформировать блок не меняется во время эксперимента, хотя, в действительности, майнеры могут изменить свои вероятности поиска нужного прообраза и формирования блока, увеличивая или уменьшая свои вычислительные ресурсы [11];

– в работе М. Розенфельда теорема про вероятность успеха злоумышленником приведена без доказательства и получена с допущением о времени распространения блока в сети равным нулю, в [12] упомянуто о том, что нужно учитывать время синхронизации сети;

– допущение о формировании блоков в соответствии со средним временем ожидания блока, сделанное в работе С. Накамото, ошибочно [13].

К сожалению, существует не так много работ, в которых проводится попытка экспериментально подтвердить или опровергнуть полученные теоретические расчеты, т.е. эмпирическим путем обосновать адекватность выбранной математической модели. К таким работам можно отнести [8, 11].

Во всех упомянутых работах используется модель разорения игрока, проверяемая методами Монте-Карло. На основе данной модели и выводится формула для расчета вероятности успешного проведения атаки двойной траты.

Цель данной работы – критический анализ известных аналитических оценок вероятности успешной реализации атаки двойной траты на протокол консенсуса «Proof of work». В частности, мы рассматриваем «задачу о разорении игрока», лежащую в основе моделей С. Накамото и М. Розенфельда, и показываем, что базовые предположения о вероятностном пространстве (множество элементарных исходов и вероятности их наступления) не соответствуют реальным процессам, протекающим при установлении консенсуса «Proof of work» в блокчейн-системе. Далее, для теоретической оценки вероятности успешной атаки двойной траты мы предлагаем использовать модель «независимых игроков», которая, на наш взгляд, устраняет основные неточности и несоответствия. Эмпирическим путем мы показываем сходимость результатов теоретических расчетов с данными экспериментов по имитации «гонки» между честными игроками и злоумышленниками. Наиболее интересным, на наш взгляд,

является сопоставление результатов теоретических расчетов, полученных применением различных моделей, и эмпирических результатов, полученных имитацией «гонки».

2. «Задача о разорении игрока» применительно к атаке двойной траты

Рассмотрим «задачу о разорении игрока», а точнее ее небольшую модификацию, на которую ссылается С. Накамото, цитируя известный учебник Феллера 1968 г. [14], или М. Розенфельд, моделируя процесс гонки как процесс эквивалентный цепочке Маркова с дискретным временем, где каждый шаг определяется как поиск блока кем-либо.

Приведем сначала выдержку из раздела 11 работы С. Накамото [4], в которой излагаются рассуждения в отношении моделирования атаки двойной траты:

«Гонку между честными участниками и нападающим можно представить как биномиальное случайное блуждание. Успешное событие, когда «честная» цепь увеличивается на один блок, приводит к увеличению отрыва на единицу, увеличивая свое преимущество на $+1$, а неуспешное, когда очередной блок создает злоумышленник, – к его сокращению на один блок, уменьшая разрыв на -1. Вероятность атакующего наверстать разницу в несколько блоков такая же, как и в задаче о «разорении игрока». Представим, что игрок имеет неограниченный кредит, начинает с некоторым дефицитом и у него есть бесконечно много попыток, чтобы отыграться».

Далее приведем выдержку из раздела 3 работы М. Розенфельда [5]:

«Обозначим через $z = n - m$ количество блоков, в которых честная сеть имеет преимущество перед атакующим. Всякий раз, когда блок найден, значение z изменяется; если этот блок был обнаружен честной сетью, z увеличивается на 1, а если этот блок был обнаружен атакующим, z уменьшается на 1. Формально это цепочка Маркова с непрерывным временем и скоростью p/T_0 для продвижения вверх на шаг, и скорость q/T_0 для движения вниз на шаг».

Как видим, в этих работах используются модель, в которой в каждом испытании выигрывает злоумышленник (формируя очередной блок) или злоумышленник проигрывает и при этом считается, что выигрывает честная сеть (формируя очередной блок). Однако в статьях не приводится какого-либо обоснования выбранной модели. Авторы допускают, что если блок не сформировал злоумышленник, то, в таком случае, блок обязательно формирует честная сеть, никак не обосновывая это допущение.

Действительно, в определении задачи о разорении игрока используется вероятностное пространство с двумя элементарными событиями: «выиграл первый игрок»; «выиграл второй игрок». При моделировании атаки двойной траты С. Накамото и М. Розенфельд интерпретируют элементарные исходы этой задачи как «блок сформирован честной сетью» (по традиции вероятность такого исхода обозначается p) и «блок сформирован атакующим» (с вероятностью q), причем $p = 1 - q$. Однако в реальных блокчейн-системах вероятность формирования блока (нахождения прообраза функции хеширования) определяется исключительно хешрейтом (вычислительными возможностями) каждого участника, т.е. условие $p = 1 - q$ не обязано выполняться. Например, при хешрейте участников, превышающем сложность поиска прообраза за определенный интервал времени, каждый участник гарантированно найдет прообраз, т.е. сформирует блок и, в этом случае, $p = 1$ и $q = 1$. В реальных системах сложность поиска прообраза корректируется исходя из вычислительных возможностей участников, причем так, чтобы прообраз был найден за определенный временной интервал (например, в криптовалюте биткоин это 10 мин.). Если предположить, что такая корректировка выполняется над двумя игроками: «честная сеть» и «атакующий», а p и q – соответствующие вероятности формирования блока за определенный временной интервал, тогда предположение $p = 1 - q$ оправданно. Однако в реальной ситуации злоумышленник атакует систему, не оглашая своих вычислительных возможностей и, вероятнее всего, скрывая сам факт предполагаемой атаки, т.е. предположение $p = 1 - q$ не имеет оснований.

Если оставить введенные обозначения (вероятности p и q) и отказаться от обязательного выполнения условия $p = 1 - q$, тогда в результате каждой попытки (или серии попыток в течение заданного интервала времени) пространство элементарных исходов содержит такие элементарные события:

- «блок сформирован честной сетью и атакующий не сформировал блок» с вероятностью $p(1-q)$;
- «блок не сформирован честной сетью и атакующий сформировал блок» с вероятностью $(1-p)q$;
- «блок не сформирован честной сетью и атакующий не сформировал блок» с вероятностью $(1-p)(1-q)$;
- «блок сформирован честной сетью и атакующий сформировал блок» pq.

Множество всех элементарных исходов составляет полную группу событий:

$$p(1-q) + (1-p)q + (1-p)(1-q) + pq = 1.$$

Эта модель с четырьмя элементарными исходами (будем называть ее в дальнейшем «модель с независимыми игроками») описывает реальный вероятностный процесс в блокчейн-системе при установлении консенсуса на основе алгоритма «Proof of work».

3. Сравнение вероятностных событий в двух исследуемых моделях

В модели независимых игроков формирование очередного блока у злоумышленника и честной сети происходит независимо друг от друга, вероятности поиска прообраза хеш-функции (для формирования блока) определяются их хешрейтами (вычислительными возможностями). Для сравнения с результатами, полученными в работах С. Накамото и М. Розенфельдом (для модели разорения игрока), будем использовать общепринятые упрощения:

– время распространения блока по сети пренебрежимо мало, т.е. обмен информацией между узлами происходит практически мгновенно (время синхронизации равно нулю);

– хешрейт злоумышленника, хешрейт честной сети и сложность майнинга не меняются со временем на протяжении всей гонки;

– возможности злоумышленника по поддержанию состояния гонки достаточно велики, но не бесконечны;

– кроме злоумышленника все остальные пользователи сети действуют строго в соответствии с правилами протокола блокчейн-сети;

– победой злоумышленника будем считать формирование необходимого количества блоков подтверждения раньше или одновременно (считается, что один блок злоумышленник сформировал заранее) или в противном случае – последующего формирования цепочки блоков равной длины с честной сетью.

В задаче двойной траты злоумышленник выигрывает, если сформирует равное с честной сетью количество блоков, при условии, что честная сеть уже сформировала N блоков. Здесь используем ту же формулировку, что и в работе М. Розенфельда [5], предполагая, что один блок был предварительно добыт атакующим до начала атаки и, следовательно, общая длина сформированной злоумышленником цепочки будет на единицу больше, что является достаточным условием для принятия ее честной сетью как основной блокчейн.

Если предполагать, что ресурсы у злоумышленника конечны или выигрыш злоумышленником не покроет его финансовых затрат на поддержание дальнейшей гонки, то логично предположить об ограничении на формирование максимального количества блоков в состязании [11]. Предположим, что злоумышленник отказывается от продолжения гонки в случае, если честная сеть сформировала $N + n_{\max}$ блоков. Все состояния, в которых злоумышленник не выиграл, будут для него проигрышными.

Необходимо обратить внимание, на два момента в модели разорения игрока:

1) злоумышленник не может выиграть гонку раньше, чем за $2 \cdot N$ попыток (необходимо не менее N попыток для формирования N блоков честной сетью и столько же попыток для формирования N блоков злоумышленником);

2) злоумышленник может выиграть при нечетном количестве попыток, только если он опередил честную сеть до того, как она сформировала N блоков (вероятность чего значительно меньше при меньших мощностях майнинга злоумышленником).

В отличие от модели разорения игрока, в модели независимых игроков злоумышленник может выиграть, начиная с N попыток и так как события формирования блоков обоими участниками независимы и нет зависимости вероятности выигрыша от четности или нечетности текущей попытки.

Рассмотрим пример расчета вероятности наступления какого-либо события для двух рассмотренных моделей. Для определенности положим вероятность формирования блока злоумышленником за каждую попытку $q = 0,3$ (вероятность не сформировать блок будет $(1-q) = 0,7$). Для согласования с моделью разорения игрока положим $p = 0,7$ (вероятность не сформировать блок честной сетью будет $(1-p) = 0,3$). Необходимое количество подтверждений $N = 1$. Ограничение на формирование максимального количества блоков в состязании $N + n_{\max} = 2$ попытки.

Проанализируем вероятности различных исходов для различных моделей.

Рассмотрим модель разорения игрока:

1) первая попытка (два возможных исхода):

– формирование блока честной сетью, злоумышленник отстает на один блок, гонка продолжается, вероятность наступления такого события равна $p = 0,7$;

– формирование блока злоумышленником, честная сеть отстает на один блок, победа злоумышленника[1], вероятность наступления события равна $q = 0,3$;

2) вторая попытка (четыре возможных исхода, рассматриваем только случай формирования блока честной сетью в первой попытке, т.е. когда гонка продолжается):

– и в первой, и во второй попытке сформирован блок честной сетью, злоумышленник проиграл в гонке, гонка завершена, вероятность наступления события $p \cdot p = 0,49$;

– в первой попытке сформирован блок честной сетью, но во второй попытке блок сформирован злоумышленником, злоумышленник выиграл, гонка завершена, вероятность наступления события $p \cdot p = 0,21$.

Таким образом, в модели разорения игрока злоумышленнику удастся победить с вероятностью $0,3 + 0,21 = 0,51$.

Для модели независимых игроков:

1) первая попытка (четыре возможных исхода):

– блок сформирован честной сетью и атакующий не сформировал блок, злоумышленник отстает на один блок, гонка продолжается, вероятность наступления события $p \cdot (1-q) = 0,49$;

– блок не сформирован честной сетью и атакующий сформировал блок, победа злоумышленника[1], вероятность наступления события $(1-p) \cdot q = 0,09$;

– блок не сформирован честной сетью и атакующий не сформировал блок, гонка продолжается, вероятность наступления события $(1-p) \cdot (1-q) = 0,21$;

– блок сформирован честной сетью и атакующий сформировал блок, победа злоумышленника, гонка завершена, вероятность наступления события $p \cdot q = 0,21$;

[1] В этом случае победа будет засчитана только после формирования $N = 1$ блока честной сетью

2) вторая попытка (шестнадцать возможных исходов, рассматриваем только те случаи, когда после первой попытки исход гонки не определен)

- (в первой попытке блок сформирован честной сетью и атакующий не сформировал блок):
  - во второй попытке блок сформирован честной сетью и атакующий не сформировал блок, злоумышленник проиграл в гонке, гонка завершена, вероятность наступления события $p \cdot (1-q) \cdot p \cdot (1-q) = 0{,}2401$;
  - во второй попытке блок не сформирован честной сетью и атакующий сформировал блок, победа злоумышленника, гонка завершена, вероятность наступления события $p \cdot (1-q) \cdot (1-p) \cdot q = 0{,}0441$;
  - во второй попытке блок не сформирован честной сетью и атакующий не сформировал блок, гонка продолжается, вероятность наступления события $p \cdot (1-q) \cdot (1-p) \cdot (1-q) = 0{,}1029$;
  - во второй попытке блок сформирован честной сетью и атакующий сформировал блок, злоумышленник проиграл в гонке, гонка завершена, вероятность наступления события $p \cdot (1-q) \cdot p \cdot q = 0{,}1029$;
- (в первой попытке блок не сформирован честной сетью и атакующий не сформировал блок):
  - во второй попытке блок сформирован честной сетью и атакующий не сформировал блок, гонка продолжается, вероятность наступления события $(1-p) \cdot (1-q) \cdot p \cdot (1-q) = 0{,}1029$;
  - во второй попытке блок не сформирован честной сетью и атакующий сформировал блок, победа злоумышленника[2], вероятность наступления события $(1-p) \cdot (1-q) \cdot (1-p) \cdot q = 0{,}0189$;
  - во второй попытке блок не сформирован честной сетью и атакующий не сформировал блок, гонка продолжается, вероятность наступления события $(1-p) \cdot (1-q) \cdot (1-p) \cdot (1-q) = 0{,}0441$;
  - во второй попытке блок сформирован честной сетью и атакующий сформировал блок, победа злоумышленника, гонка завершена, вероятность наступления события $(1-p) \cdot (1-q) \cdot p \cdot q = 0{,}0441$.

Таким образом, в модели независимых игроков злоумышленнику удастся победить за две попытки с вероятностью $0{,}09 + 0{,}21 + 0{,}0441 + 0{,}0189 + 0{,}0441 = 0{,}4071$, что отличается от вероятности, рассчитанной для модели разорения игрока.

С помощью моделирования проведем вычислительный эксперимент и эмпирически оценим вероятности выигрыша злоумышленника у честной сети при различных моделях формирования цепочки блоков.

4. Моделирование вычислительного эксперимента

На первом этапе протестируем вероятности формирования N блоков ровно за t попыток и вероятность формирования N блоков при проведении t испытаний.

4.1. Вероятность формирования цепочки блоков заданной длины

На первом этапе протестируем вероятность формирования блока ровно за t попыток.

В качестве входных параметров будем использовать:

q – вероятность успешно сформировать блок злоумышленником при каждом испытании. Вероятность зависит от имеющихся у злоумышленника вычислительных возможностей (т.е. пропорциональна хешрейту злоумышленника);

p – вероятность успешно сформировать блок честными участниками при каждом испытании (пропорционально хешрейту честной сети). При моделировании будем полагать, что

[2] В этом случае победа будет засчитана только после формирования $N = 1$ блока честной сетью

$p = 1 - q$, так как такая взаимосвязь лежит в основе модели разорения игрока. В общем случае для модели независимых игроков условие $p = 1 - q$ может не выполняться;

N – количество блоков в сети, после которых сделка считается подтвержденной;

t – номер текущей попытки.

В программной среде создадим процесс, итеративно пытающийся сформировать блоки. Каждое испытание происходит по следующему правилу:

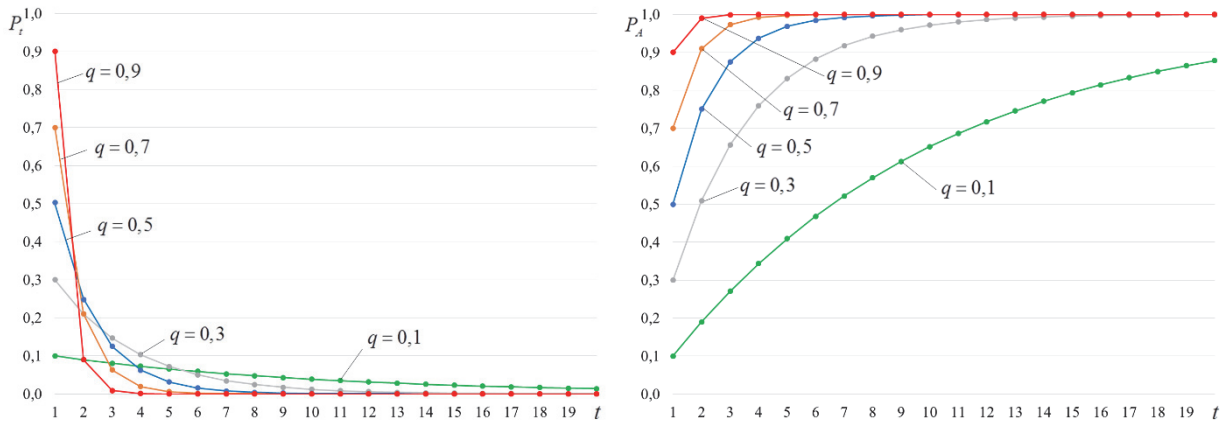
– генерируем псевдослучайное число (используем реализацию на основе Вихря Мерсенна) в интервале $[0,1]$ (в программной реализации используется минимальный шаг генерации $5.4 \cdot 10^{-20}$, что позволяет тестировать $q \geq 10^{-19}$);

– сравниваем сгенерированное случайное число со значением q;

– если сгенерированное число $\leq q$, то считаем, что генерация блока прошла успешной и увеличиваем счетчик сформированных потоком блоков ($k\_блок1$) на единицу. Проверяем $k\_блок1 = N$, если сформировали необходимо количество блоков, то увеличиваем массив $Mass1[t]$ на единицу, что соответствует удачной попытке сформировать цепочку блоков нужной длины на t-й попытке;

Для достижения заданной точности испытания проводим N_{test} раз (выбор N_{test} описан ниже). По окончании всех испытаний нормируется результат (массив $Mass1[t]$) по общему количеству испытаний и таким образом получаем эмпирическое распределение вероятности формирования N блоков от количества проведенных попыток P_t. Суммируя все полученные вероятности от 1 до заданного t получаем эмпирическую функцию распределения вероятности формирования блока P_A.

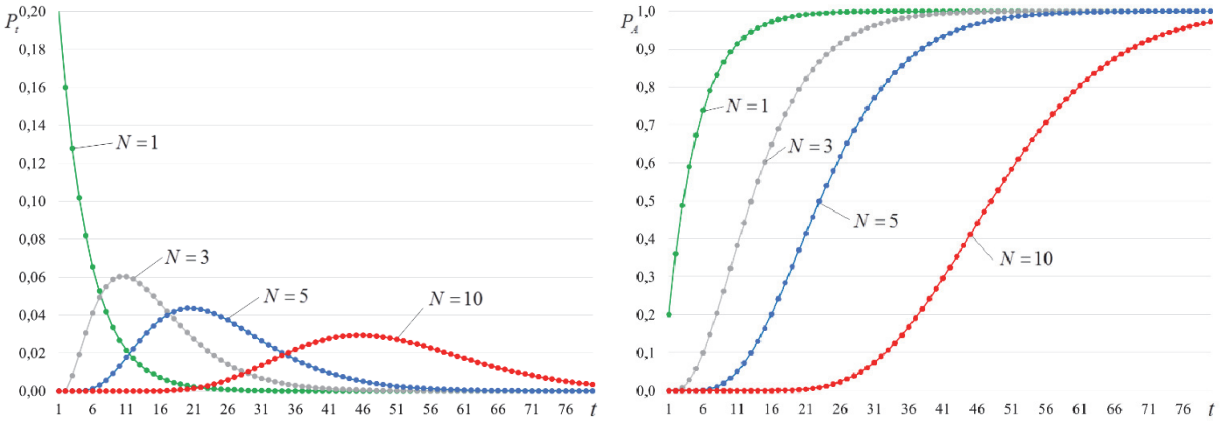
Результаты испытаний приведены на рис. 1 (точки). Сплошные линии соответствуют отрицательному биномиальному распределению и его функции распределения для тех же вероятностей.



а *б*

Рис. 1. Функция вероятности (*а*) и функция распределения вероятности (*б*) формирования блока при каждом испытании (линиями отображены расчетные значения, соответствующие отрицательному биномиальному распределению, точки – экспериментальные данные)

Построим вероятность формирования цепочки из N блоков. На рис. 2 представлены полученные аналогичные вероятности, но для фиксированного значения q и разного числа N.



а *б*

Рис. 2. Функция вероятности (*а*) и функция распределения вероятности (*б*) формирования цепочки из N блоков при $q = 0,2$ (линиями отображены расчетные значения, точки соответствуют экспериментальным данным)

Как видим, значения, полученные вычислительным моделированием, хорошо согласуются с отрицательным биномиальным распределением.

На втором этапе будем моделировать двух конкурирующих участников.

Будем исследовать две модели соревнования (гонки) злоумышленника с честной сетью по формированию цепочки блоков:

– модель разорения игрока;

– модель независимых игроков.

4.2. Модель разорения игрока

В программной среде создадим процесс (соответствующий злоумышленнику), итеративно пытающийся сформировать блоки. Каждое испытание происходит по следующему правилу:

– генерируем случайное число в интервале $[0,1]$;

– сравниваем полученное число с q;

– если сгенерированное число $\le q$, то считаем, что генерация блока прошла успешной и увеличиваем счетчик сформированных злоумышленником блоков ($k\_блок1$) на единицу;

– если сгенерированное число $> q$, то считаем, что блок сгенерирован честной сетью и увеличиваем счетчик сформированных ею блоков ($k\_блок2$) на единицу. Проверяем $k\_блок2 \ge N$, если да, то проверяем: сформировал ли злоумышленник необходимой длины цепочку:

- если злоумышленнику также удалось сформировать необходимое количество блоков (т.е. $k\_блок1 \ge k\_блок2$) то увеличиваем $Mass1[t]$ на единицу, что соответствует удачной попытке злоумышленника сформировать цепочку блоков нужной длины за t попыток (злоумышленник выиграл гонку). Завершаем испытание;
- если злоумышленник еще не сформировал необходимое количество блоков (т.е. $k\_блок1 < k\_блок2$), то продолжаем испытание;

– если $k\_блок2 = N + n_{\max}$, заканчиваем испытание, присваивая победу честной сети (увеличиваем массив $Mass2[t]$ на единицу).

В вычисленных экспериментах положим $n_{\max} = 1000$ (что соответствует практически неограниченным ресурсам злоумышленника). При выборе $n_{\max}$ мы упомянем работу [11],

в которой утверждается, что для $q < 0.45$ выбор значение $n_{\max} = 35$ практически не влияет на результат, кроме того, данный вопрос будет рассмотрен ниже.

4.3. Модель независимых игроков

В программной среде создадим два независимых процесса (первый процесс, соответствующий злоумышленнику, второй – честным пользователям), итеративно пытающихся сформировать блоки. Каждое испытание происходит по следующему правилу:

– генерируем случайное число в интервале $[0,1]$;

– сравниваем полученное число с q;

– если сгенерированное число $\le q$, то считаем, что генерация блока прошла успешной и увеличиваем счетчик сформированных первым потоком блоков ($k\_блок1$) на единицу;

– генерируем случайное число в интервале $[0,1]$;

– сравниваем полученное число с p;

– если сгенерированное число $\le p$, то считаем, что генерация блока прошла успешной и увеличиваем счетчик сформированных вторым потоком блоков ($k\_блок2$) на единицу. Проверяем $k\_блок2 \ge N$, если да, то проверяем: сформировал ли первый поток необходимой длины цепочку:

- если и злоумышленнику также удалось сформировать необходимое количество блоков (т.е. $k\_блок1 \ge k\_блок2$) то увеличиваем $Mass1[t]$ на единицу, что соответствует удачной попытке сформировать цепочку блоков нужной длины злоумышленником за t попыток (злоумышленник выиграл гонку). Завершаем испытание;
- если злоумышленник еще не сформировал необходимое количество блоков, то продолжаем испытание;

– если $k\_блок2 = N + n_{\max}$ заканчиваем испытание присваивая победу честной сети (увеличиваем массив $Mass2[t]$ на единицу).

4.4. Обеспечение точности и достоверности результатов моделирования

С помощью имитационного моделирования точное значение случайной величины (обозначим ее Θ) определить нельзя, так как число реализаций модели ограничено. При конечном числе реализаций модели определяется приблизительное значение заданной характеристики. Обозначим это приближение Θ^*. Приблизительное значение называют оценкой соответствующей характеристики [15, 16].

Точностью оценки характеристики Θ^* называют величину ε относительно

$$\left|\Theta^* - M\left[\Theta\right]\right| < \varepsilon,$$

где $M\left[\Theta\right]$ – математическое ожидание случайной величины [15, 16].

Величина ε представляет собой абсолютное значение ошибки в определении значения искомой характеристики.

Достоверностью оценки характеристики Θ^* называют вероятность α того, что заданная точность достигается [15, 16]:

$$P\left(\left|\Theta^* - M\left[\Theta\right]\right| < \varepsilon\right) = \alpha.$$

Достоверность характеризует повторяемость, устойчивость эксперимента и трактуется так: если для оценки $M\left[\Theta\right]$ использовать величину Θ^*, то в среднем на каждые 1000 использований данного правила в $1000 \cdot \alpha$ случаев величина Θ^* будет отличаться от $M\left[\Theta\right]$ на величину меньше ε.

В ряде случаев целесообразно использовать относительную точность

$$d = \varepsilon / M\left[\Theta\right].$$

В этом случае достоверность оценки имеет вид

$$P\left(\left|\frac{\Theta^* - M\left[\Theta\right]}{M\left[\Theta\right]}\right| < d\right) = \alpha .$$

Если принять предположение относительно нормального распределения случайной величины[3], тогда функциональная связь между относительной точностью и достоверностью с количеством реализаций N_{test} имеет вид [15]:

$$N_{test} = \frac{t_\alpha{}^2 (1-P)}{Pd^2},$$

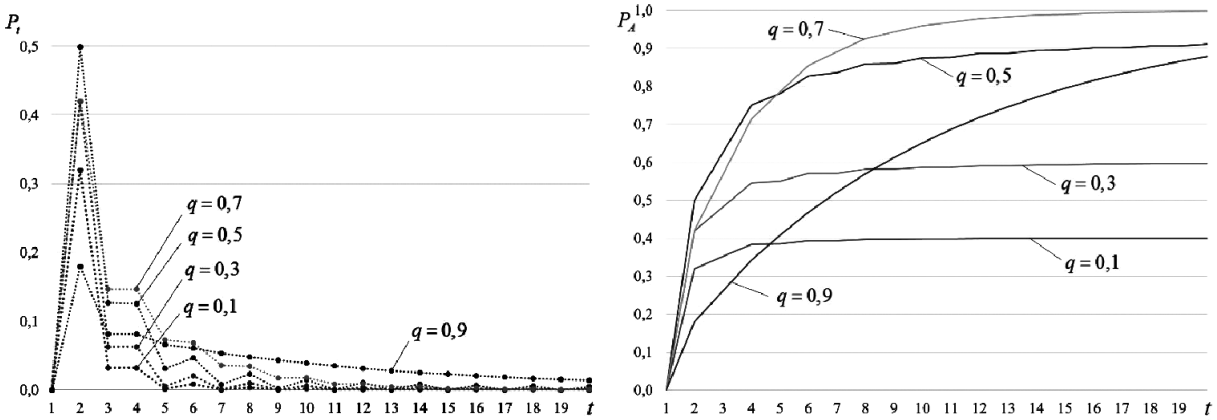
где t_α – аргумент функции Лапласа $t_\alpha = \Phi_0^{-1}\left(\frac{\alpha}{2}\right)$, интеграл Лапласа табулированный, следовательно, задаваясь значением достоверности α, можем определить t_α.

Из последней формулы следует, что при определении оценок малых вероятностей с приемлемой точностью необходимо выполнить очень большое число реализаций модели. При отсутствии высокопроизводительного компьютера применения статистического моделирования становится проблематичным.

Для проведения экспериментальных исследований были выбраны $\alpha = 0.99$ и $d = 0.01$, значение N_{test} рассчитывалось по приведенной выше формуле.

5. Результаты вычислений

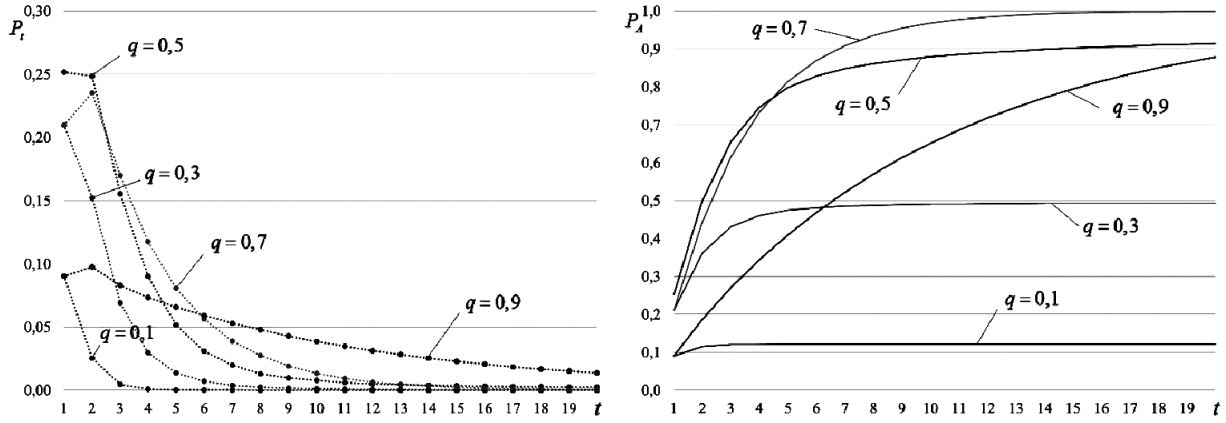
С использованием рассмотренных моделей были получены эмпирические оценки вероятности удачного формирования злоумышленником цепочки блоков при разных значениях q и N. На рис. 3 – 8 приведены полученные результаты в зависимости от номера попытки для каждого испытания, а также представлены соответствующие функции распределения вероятности в зависимости от количества попыток для каждого испытания.



а *б*

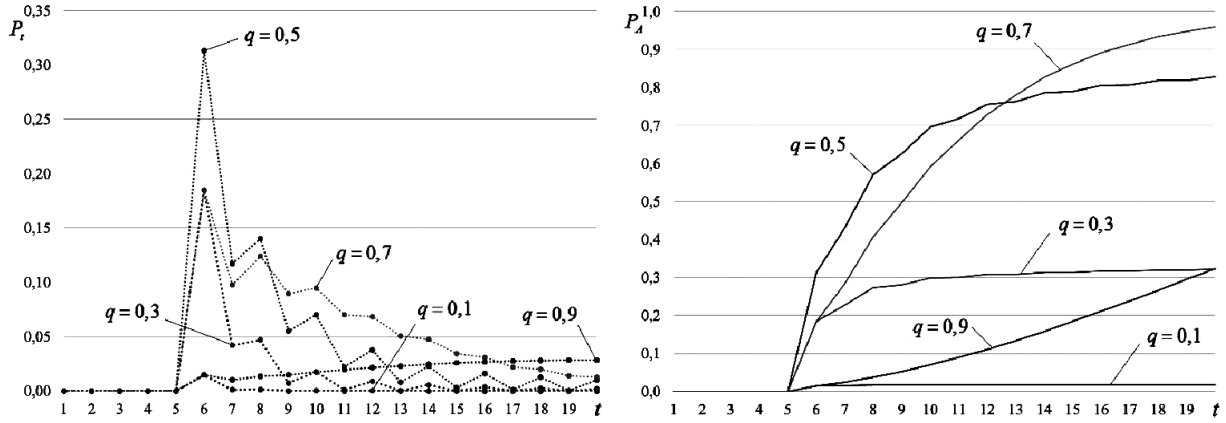
Рис. 3. Функция вероятности (*а*) и функция распределения вероятности (*б*) формирования цепочки для $N = 1$ подтверждений злоумышленником при участии двух конкурирующих субъектах (модель разорения игрока)

[3] В силу центральной предельной теоремы для большого числа испытаний биномиальное распределение хорошо аппроксимируется нормальным распределением [15, 16]



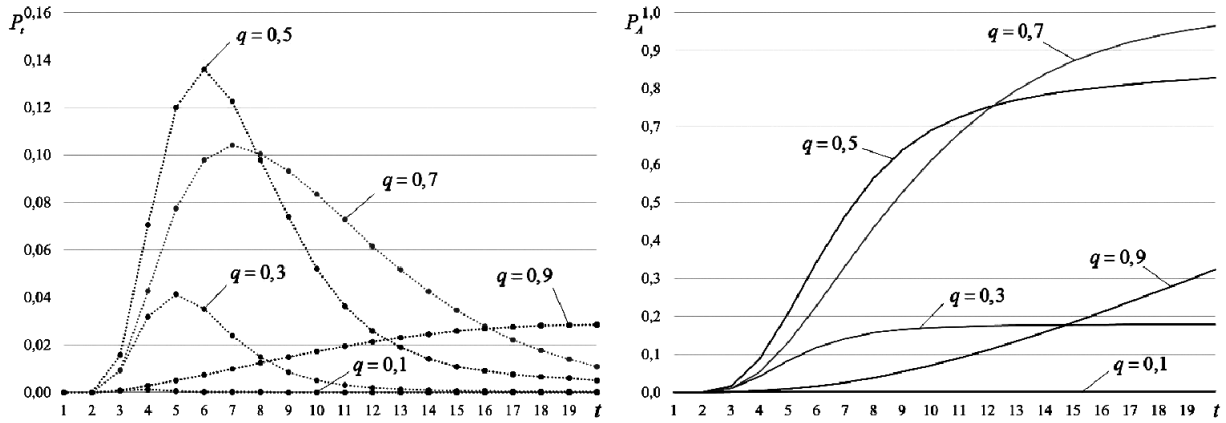
а *б*

Рис. 4. Функция вероятности (*а*) и функция распределения вероятности (*б*) формирования цепочки для $N = 1$ подтверждений злоумышленником при участии двух конкурирующих субъектах (модель независимых игроков)



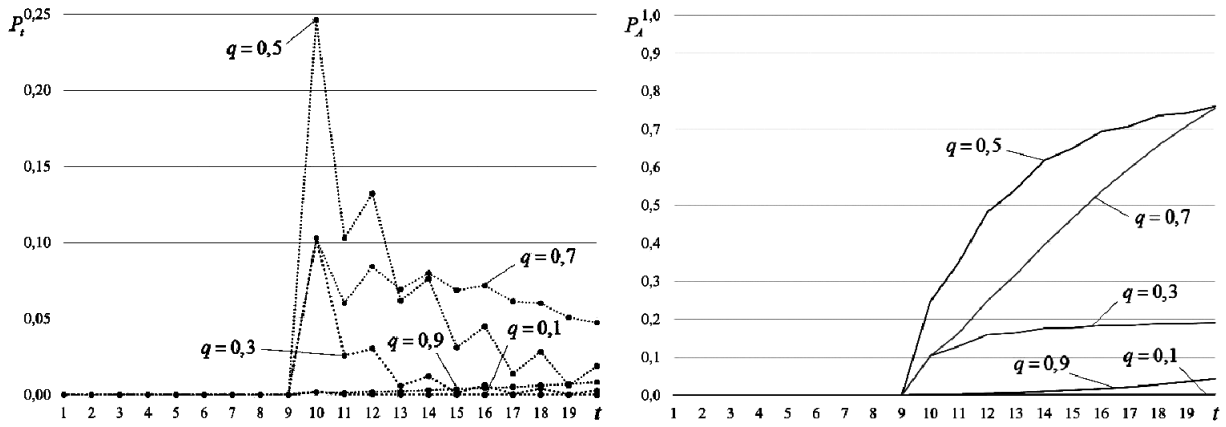
а *б*

Рис. 5. Функция вероятности (*а*) и функция распределения вероятности (*б*) формирования цепочки для $N = 3$ подтверждений злоумышленником при участии двух конкурирующих субъектах (модель разорения игрока)



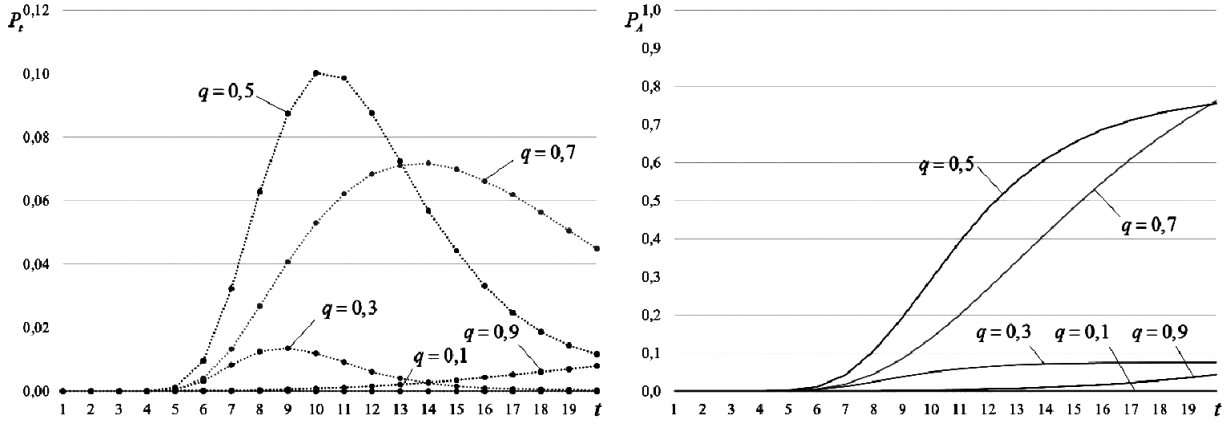
а *б*

Рис. 6. Функция вероятности (*а*) и функция распределения вероятности (*б)* формирования цепочки для $N = 3$ подтверждений злоумышленником при участии двух конкурирующих субъектах (модель независимых игроков)



а *б*

Рис. 7. Функция вероятности (*а*) и функция распределения вероятности (*б*) формирования цепочки для $N = 5$ подтверждений злоумышленником при участии двух конкурирующих субъектах (модель разорения игрока)

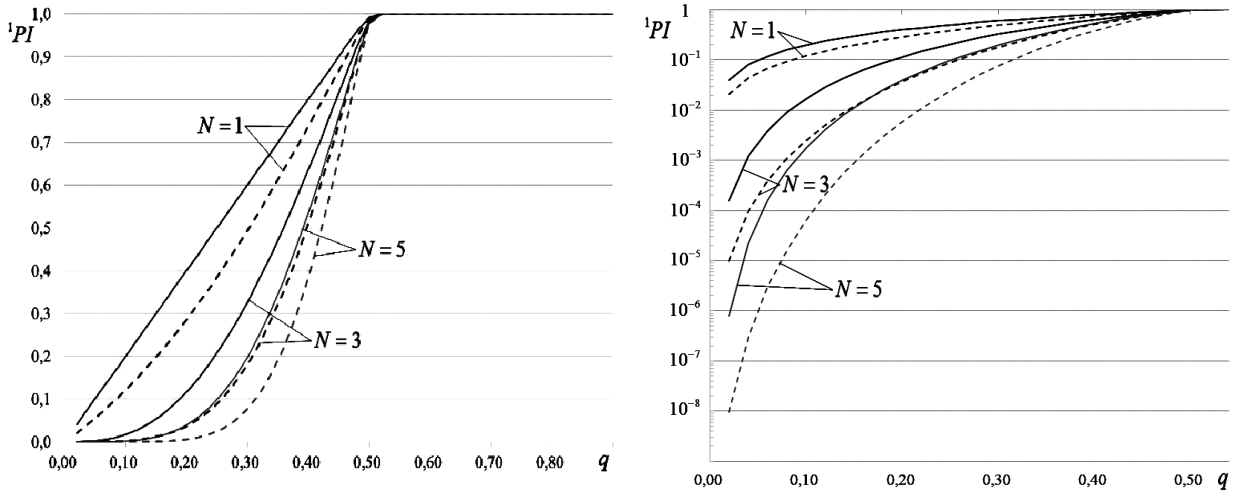


а *б*

Рис. 8. Функция вероятности (*а*) и функция распределения вероятности (*б*) формирования цепочки для $N = 5$ подтверждений злоумышленником при участии двух конкурирующих субъектах (модель независимых игроков)

Просуммировав описанные вероятности по всем возможным испытаниям, то есть для всех $t = 1, 2, 3, \ldots$, получим интегральную (или общую) вероятность успешного формирования альтернативной цепочки блоков для N подтверждений злоумышленником (^1PI).

Для приведенных примеров интегральная вероятность успешного формирования злоумышленником цепочки для N подтверждений приведена на рис. 9. Для удобства анализа полученных данных один и тот же результат приведен в обычной шкале (хорошо иллюстрирует поведение кривых при $q > 0,2$) и в логарифмическом масштабе (для иллюстрации кривых при $q < 0,2$). В этих и последующих графиках изменение значения q проводилось с шагом 0,02.



а *б*

Рис. 9. Интегральная вероятность успешного формирования злоумышленником цепочки блоков для N подтверждений при участии двух конкурирующих субъектах (экспериментальные данные). Сплошная линия – модель разорения игрока, пунктиром – модель независимых игроков. *а* – обычная шкала, *б* – логарифмическая шкала

Как видим, результаты различных моделей значительно отличаются друг от друга. Рассмотрим относительную ошибку моделирования (для двух рассмотренных моделей), определяемую как

$$\frac{^1PI_{мри} - {}^1PI_{мни}}{^1PI_{мри}} \cdot 100\%,$$

где $^1PI_{мри}$ – интегральная вероятность, рассчитанная на основе модели разорения игрока; $^1PI_{мни}$ – интегральная вероятность, рассчитанная на основе модели независимых игроков,

При веденном обозначении значения относительной ошибки моделирования приведены в табл. 1.

Таблица 1

Значение относительной ошибки в результате применения различных моделей (на основе разорения игрока и независимых игроков)

	$q = 0,01$	$q = 0,2$	$q = 0,4$
$N = 1$	48 %	29 %	8 %
$N = 3$	94 %	68 %	20 %
$N = 5$	99 %	85 %	28 %

Как видим из таблицы, две рассмотренные модели атаки двойной траты (модель разорения игроков и модель независимых игроков) дают различные оценки вероятности выигрыша гонки злоумышленником (успеха атаки). По мере увеличения длины цепочки блоков N расхождение увеличивается (относительная ошибка моделирования достигает 100 %). Это наблюдается для различных вероятностей q (т.е. для различных соотношений хешрейтов злоумышленника и честной сети).

Следует отметить, что результаты, полученные на основе модели разорения игрока, соответствуют (в пределах заданной достоверности и выбранной относительной точности) аналитическим результатам, полученным на основе формул М. Розенфельда (см. выражение 1 и рис. 4 из [5]. Отличие наблюдается только в точке $q = 0,5$, где относительная ошибка между

экспериментальными и аналитическими результатами составила 1,7 % для $N = 3$ и 2,2 % для $N = 5$, что связанно с ограничением в $n_{\max} = 1000$ блоков.

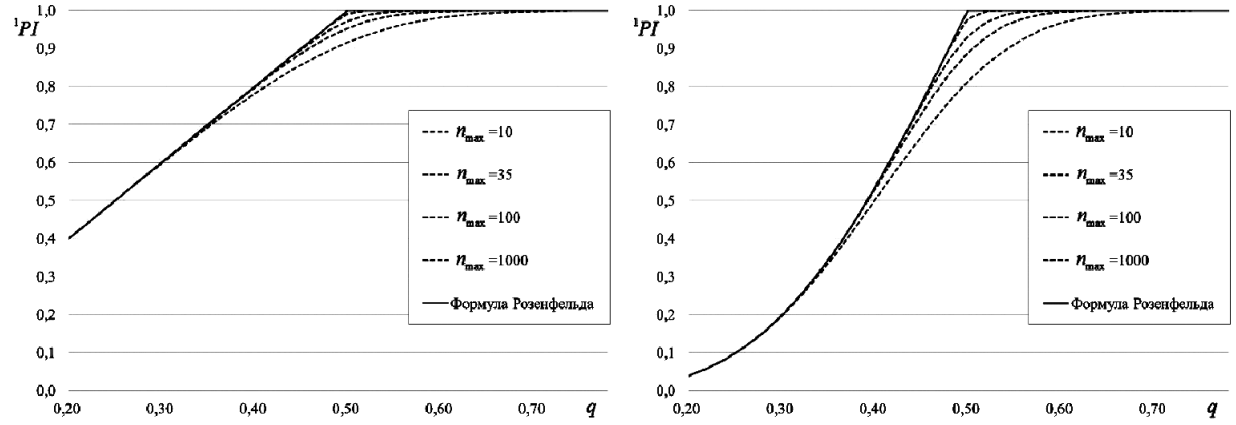
Как было показано в работе [11], результат имеет отличия при разных значениях $n_{\max}$. Проанализируем данный вопрос подробнее.

6. Влияние $n_{\max}$ на вероятность победы злоумышленника

Учитывая, что поддержка гонки злоумышленником постоянно требует определенных финансовых затрат от злоумышленника, то гонка только теоретически может продолжаться бесконечно. В реальных обстоятельствах злоумышленнику будет невыгодно продолжать гонку и затрачивать на ее поддержание больше ресурсов, чем он сможет себе вернуть, удачно проведя атаку двойной траты, или имеет в своем распоряжении. Другой вариант, если злоумышленник в состоянии формировать определенное количество блоков на протяжении большого промежутка времени, то ему может быть экономически выгодней их публиковать по правилам сети, получая за это награду, чем пытаться извлечь выгоду из нечестного (не соответствующего правилам сети) поведения. Если злоумышленник отстал в гонке с честной сетью на значительное количество блоков, то, как показано выше, его шансы победить значительно снижаются, и ему уже нет смысла продолжать попытки до бесконечности.

При всех рассмотренных вариантах значение $n_{\max}$ есть число конечное. Рассмотрим его влияние на вероятность победы злоумышленника.

В качестве иллюстрации на рис. 10 приведены графики экспериментальных значений, полученных в соответствии с моделью разорения игрока, для $n_{\max} = 10, 35, 100, 1000$ и разных $N = 1$ и 5, а также проводится сравнение с теоретическими результатами, полученными М. Розенфельдом.



а *б*

Рис. 10. Интегральная вероятность успешного формирования злоумышленником цепочки блоков для $N = 1$ (*а*) и $N = 5$ (*б*) подтверждений при участии двух конкурирующих субъектов (пунктир – экспериментальные данные). Модель разорения игрока

Как видно из приведенных результатов, увеличение $n_{\max}$ приближает полученные эмпирические данные к аналитическим результатам М. Розенфельда [5]. С уменьшением вероятности теоретические результаты, полученные М. Розенфельдом, хорошо аппроксимируются при небольших $n_{\max}$.

Относительная ошибка между теоретическими и экспериментальными результатами близка к значению $q = 0,5$ и составляет более 0,1 % в следующих диапазонах:

для $N = 1$:

– от $0,28 \le q \le 0,72$ при $n_{\max} = 10$;

– от $0,4 \le q \le 0,6$ при $n_{\max} = 35$;

– от $0,44 \le q \le 0,56$ при $n_{\max} = 100$;

– $q = 0,5$ при $n_{\max} = 1000$;

для $N = 5$:

– от $0,26 \le q \le 0,70$ при $n_{\max} = 10$;

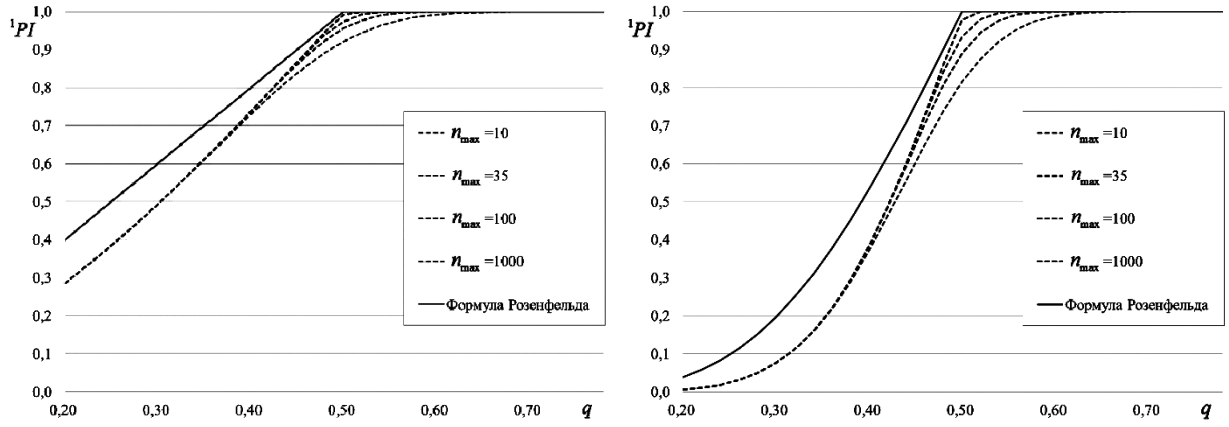
– от $0,38 \le q \le 0,62$ при $n_{\max} = 35$;

– от $0,42 \le q \le 0,54$ при $n_{\max} = 100$;

– от $0,48 \le q \le 0,5$ при $n_{\max} = 1000$;

Сходимость результатов экспериментов с теоретическими расчетами по известным аналитическим выражениям подтверждает адекватность и обоснованность результатов исследований.

На рис. 11 приведены экспериментальные результаты, полученные в соответствии с моделью независимых игроков при тех же параметрах ($n_{\max} = 10, 35, 100, 1000$; $N = 1, 5$). Для наглядности оставлен теоретический результат, полученный М. Розенфельдом.



а *б*

Рис. 11. Интегральную вероятность успешного формирования злоумышленником цепочки блоков для $N = 1$ (*а*) и $N = 5$ (*б*) подтверждений при участии двух конкурирующих субъектах (пунктир – экспериментальные данные). Модель независимых игроков.

Как видим, характер влияния $n_{\max}$ на полученный результат сохраняется и для модели независимых игроков. Однако сопоставление полученных результатов для различных моделей подтверждает тезис о расхождении оценок вероятностей успешной атаки двойной траты.

7. Выводы

Проведен критический анализ известных работ по оценке вероятностей двойной траты в протоколе консенсуса «Proof of work». Показано наличие неточностей и необоснованных допущений в известных работах С. Накамото [4] и М. Розенфельда [5]. В частности, показано, что базовые предположения о вероятностном пространстве (множество элементарных исходов и вероятности их наступления) в используемой модели разорения игроков (с двумя элементарными исходами) не соответствуют реальным процессам, протекающим при установлении консенсуса «Proof of work».

Для теоретической оценки вероятности успешной атаки двойной траты предложено использовать модель независимых игроков с четырьмя элементарными исходами. Эта модель описывает реальный вероятностный процесс в блокчейн-системе при установлении консенсуса на основе алгоритма «Proof of work», когда каждый участник (злоумышленник и честная сеть) независимо формируют блоки с вероятностями, пропорциональными своему хешрейту (своим вычислительным возможностям).

Проведено сравнение результатов, полученных с помощью вычислительного моделирования атаки двойной траты на основе модели разорения игрока и модели независимых игроков. Сравнение проведено для разных возможностей злоумышленника (вероятности сформировать блок), различного количества сформированных блоков, после которых сделка считается подтвержденной, различной продолжительности гонки (количества блоков, на протяжении которых злоумышленник продолжает попытки догнать честную сеть). Показано значительное отличие (относительная ошибка модели до 99 %) результатов, полученных в вычислительном моделировании при использовании модели независимых игроков от модели разорения игрока.

Все эмпирические оценки получены для высокой точности (относительная ошибка не более 1 %) и достоверности (доверительная вероятность не менее 99 %).

Для подтверждения адекватности полученных результатов приведено сравнение эмпирических результатов с теоретическими расчетами по известным аналитическим соотношениям. Показано, что результаты вычислительного эксперимента для модели разорения игрока полностью совпадают (в пределах заданной достоверности и относительной точности) с аналитическим результатом, приведенным в работе М. Розенфельда [5].

На основе полученных результатов можно утверждать об ошибочности использования модели разорения игрока для оценки вероятности успешной атаки двойной траты на протокол консенсуса «Proof of work».

Список литературы:

1. The Double Spending Problem and Cryptocurrencies. Banking & Insurance Journal. Social Science Research Network (SSRN). Accessed 24 December 2017. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3090174
2. Mark Ryan. Digital Cash // School of Computer Science, University of Birmingham. Retrieved 2017-05-27. https://www.cs.bham.ac.uk/~mdr/teaching/modules06/netsec/lectures/DigitalCash.html
3. Varshney, Neer (2018-05-24). Why Proof-of-work isn't suitable for small cryptocurrencies // Hard Fork. Retrieved 2018-05-25. https://thenextweb.com/hardfork/2018/05/24/proof-work-51-percent-attacks/
4. Nakamoto S. Bitcoin: A Peer-to-Peer Electronic Cash System / Satoshi Nakamoto, 2009. 9 с.
5. Rosenfeld M. Analysis of hashrate-based double-spending / Meni Rosenfeld, 2014. 13 с.
6. Carlos Pinzón, Camilo Rocha. Double-spend Attack Models with Time Advantange for Bitcoin // Electronic Notes in Theoretical Computer Science. Volume 329, 9 December 2016, Pages 79-103 https://doi.org/10.1016/j.entcs.2016.12.006
7. Kaidalov D.S., Kovalchuk L.V., Nastenko A.O., Rodinko M.Yu., Shevtsov O.V., Oliynykov R.V. Comparison of block expectation time for various consensus algorithms // Radio Electronics, Computer Science, Control. 2018. № 4. PP. 159- 171 DOI 10.15588/1607-3274-2018-4-15
8. Azzolini D., Riguzzi F., Lamma E., Bellodi E., Zese R. Modeling Bitcoin Protocols with Probabilistic Logic Programming http://ceur-ws.org/Vol-2219/paper6.pdf
9. Kevin Liao, Jonathan Katz. Incentivizing Double-Spend Collusion in Bitcoin. 2017. https://www.cs.umd.edu/~gasarch/reupapers/katzbitcoin16.pdf
10. Ковальчук Л.В. Основні визначення у галузі блокчейну та детальний аналіз результатів Накамото-Розенфельда-Грунспана про імовірність атаки подвійної витрати. Звіт про НДР (проміжний). Харків : АТ ІІТ. 36 с.
11. Pinar Ozisik., Brian Neil Levine. An Explanation of Nakamoto's Analysis of Double-spend Attacks https://arxiv.org/pdf/1701.03977.pdf
12. Apostolaki M. Hijacking Bitcoin: Routing Attacks on Cryptocurrencies / M. Apostolaki, A. Zohar, L. Vanbever. San Jose, CA , USA, 2017. 18 с.
13. Grunspan C., Pérez-Marco R. Double spend races. 2017. hal-01456773 https://hal.archives-ouvertes.fr/hal-01456773
14. W. Feller. An Introduction to Probability Theory and its Applications: Volume I, volume 3. John Wiley & Sons London-New York-Sydney-Toronto, 1968
15. Смирнов Н.В., Дунин-Барковский И.В. Курс теории вероятностей и математической статистики для технических приложений. Москва : Наука, 1969. 512 с.
16. Вентцель Е.С. Теория вероятности. Москва : Наука, 1969. 576с.

Харьковский национальный
университет имени В.Н. Каразина;
АО «Институт информационных технологий», Харьков

Поступила в редколлегию 27.08.2019

УДК 004.056.55 **DOI:10.30837/rt.2019.3.198.12**

І.Д. ГОРБЕНКО, д-р техн. наук, О.В. ПОТІЙ, д-р техн. наук,
Ю.І. ГОРБЕНКО, канд. техн. наук, А.І. ПУШКАРЬОВ, М. В. ЄСІНА, канд. техн. наук

ПРИНЦИПИ ПОБУДУВАННЯ ТА АНАЛІЗУ ІНФРАСТРУКТУР ВІДКРИТОГО КЛЮЧА НА ОСНОВІ ЗАСТОСУВАННЯ ТЕХНОЛОГІЇ БЛОКЧЕЙН

Вступ

Результати теоретичних та практичних досліджень асистемних підходів та досвіду застосування нових інформаційних технологій (ІТ), що наведені у [1 – 9] дозволяють зробити висновки про можливості інтенсивного розроблення, активного розповсюдження та застосування децентралізованих інформаційних технологій. Для досягнення вказаного в тій чи іншій мірі необхідно виконати наступні вимоги [10, 1 – 9].

1. Застосування принципу децентралізації існуючих ІТ повинне поліпшити хоча б один із важливих для цільового застосування параметр: вартість, складність (часова та просторова), швидкість, прибутковість, безпека (загальна та інформаційна), анонімність, прозорість, гнучкість тощо. Причому важливо, щоби цільовий параметр, за яким здійснюється оцінка, пропонувався самим замовником (користувачами).

2. Покращення ІТ, що досягається, має бути суттєвим, причому децентралізація повинна покращувати хоча б один важливий для користувачів параметр, як мінімум в 2-3 рази.

3. Нова, в даному випадку технологія блокчейн (ТБЧ), не повинна істотно програвати існуючим ІТ за іншими важливими параметрами. Наприклад [10], якщо ТБЧ працює в три рази швидше, але, якщо вона при цьому в два рази дорожче і в 1,5 рази складніше – вона, скоріше за все, не буде застосовуватись. Вважається, що у цілому нова ІТ повинна бути в чомусь краще в 2-3 рази, а за всіма іншими параметрами програвати не більше, ніж у 1,5 рази. Тобто, покращення повинне не тільки давати суттєві переваги, але ще й компенсувати побічні ефекти програшу.

Таким чином, «кращість» і перевага нової ІТ носить суб'єктивний характер, вони повинні формуватись скоріше всього користувачами, і в меншій мірі розробниками. У більшості випадків кращість може визначатись рівнем продаж таких ІТ. Відсутність певних продаж показує, що явної переваги цільовою аудиторії відносно нової чи удосконаленої ІТ не визнано.

Аналіз показав, що серед нових технологій суттєвий розвиток та «кращість» досягають ІТ, що розробляються чи удосконалюються на основі використання ТБЧ, що ґрунтується на децентралізації. Таким чином, існує проблема, сутність якої в тому, що децентралізація, в тому числі у вигляді ТБЧ, є «начебто» перспективною інформаційною технологією. В той же час, де саме та як її краще використовувати, де вона буде, в порівнянні з існуючими технологіями типу «Клієнт–сервер», кращою, на наш погляд, є не вирішеним питанням.

Одним із важливих та необхідних додатків ТБЧ є, по суті, удосконалення інфраструктури відкритого ключа (ІВК) на основі використання при її побудови принципів децентралізації та прозорості тощо. У явному вигляді проблема вже викладена та обговорюється в значному числі робіт [1 – 15].

Мета статті:

- обґрунтування можливостей та необхідності створення ІВК на основі ТБЧ;
- розробка структури ІВК на основі БЧ та оцінка складності впровадження;
- аналіз удосконаленої моделі ІВК з прозорістю сертифікатів на основі БЧ;
- аналіз основних проблемних питань перспективних ІВК на базі БЧ;
- загальна оцінка стійкості ІВК на основі БЧ до відомих атак.

Автори розуміють, що стаття носить характер первинного загальносистемного аналітичного огляду та відображає погляди авторів на можливості та необхідність створення ІВК з використанням ТБЧ. Ця наша впевненість ґрунтується на тому, що діюча ІВК України роз-

роблена в суттєвій мірі за участі авторів цієї статті, в тому числі практично реалізована та підтримується при експлуатації [16 – 18].

Загальні положення щодо побудування та аналізу ІВК на основі застосування БЧ

Особливістю асиметричних криптографічних перетворень є те, що при їх виконанні використовується одна або декілька асиметричних пар ключів. Наприклад, для ЕП та АСШ використовуються дві різні асиметричні ключові пари [16 – 19]. Наприклад, для криптографічних перетворень у групі точок еліптичної кривої [16, 23] кожна асиметрична ключова пара (d_A, Q_A), де $1 \le d_A < n$ є випадкове число – особистий (закритий) ключ, а Q_A – точка на еліптичній кривій – відкритий ключ, що обчислюється способом використання скалярного множення:

$$Q_A = d_A \cdot G (\mathrm{mod}\, q),$$

де G – базова точка на еліптичній кривій порядку n, q – модуль перетворення.

Згідно з вимогами до таких асиметричних криптосистем щодо застосування особистих ключів повинні безумовно бути виконаними вимоги забезпечення їх конфіденційності, цілісності, справжності, доступності та неспростовності. Вказані вимоги можуть бути забезпеченими кожним із користувачів, оскільки особистий ключ доступний тільки його власнику і він повинен і може зберігати його в таємниці. В той же час відкритий ключ повинен бути доступним, як мінімум, усім користувачам домену, а то і усьому цифровому світу. При застосуванні відкритих ключів з високим рівнем гарантій повинні бути забезпеченими щодо них послуги цілісності, справжності, доступності та неспростовності, незалежно від математичного методу, що використовується для побудови (генерування) асиметричної пари. Вказані асиметричні пари ключів застосовуються для електронного підпису (ЕП), асиметричного шифрування (АСШ) та різних криптографічних протоколів (КРП).

В системах БЧ ЕП є основним механізмом забезпечення цілісності, справжності та неспростовності транзакцій. У [16 – 23] наведено основні методи асиметричних криптоперетворень, стандартизованих щодо ЕП, які на нинішній час знайшли широке застосування, в тому числі для захисту транзакцій. Причому відкриті ключі перевірки ЕП повинні бути доступними всім користувачам, що виконують, наприклад, перевірку підписаних електронних документів, даних тощо. За таких умов необхідно забезпечити їх цілісність, справжність і доступність та їх неспростовність для надання користувачам електронних довірчих послуг [19, 22].

Факти та приклади здійснення атак на існуючі ІВК

Нині ІВК є третьою довіреною стороною, яка надає, в тому числі, послуги автентифікації та ідентифікації особистостей в Інтернеті та інших мережах. У загальному випадку ІВК визначає політику та процедури, що необхідні для видачі, управління, перевірки та розповсюдження цифрових сертифікатів для безпечного використання підпису (ЕП), асиметричного шифрування (АСШ) та різних криптографічних протоколів (КРП) [22, 24]. Зазвичай управління ІВК ґрунтується на стандарті сертифіката [24] ДСТУ ITU-TRec.X.509|ISO/IEC 9594-8:2006 (2015), який забезпечує перевірку права власності на особистий ключ деяким зовнішнім об'єктом. Сертифікат X.509 визначає структуру даних, яка пов'язує значення відкритого ключа з суб'єктами (наприклад, доменними іменами) тощо. Причому прив'язка користувачів затверджується центром сертифікації ключів (ЦСК), які з використанням особистого ключа підписують кожен відкритий ключ, виготовляючи таким чином сертифікат відповідного відкритого ключа.

У процесі широкого застосування ІВК виявлено суттєвий недолік, що пов’язаний з можливою компрометацією особистих ключів чи особистого ключа ЦСК. У цьому випадку ЦСК є точками відмови в ІВК, оскільки це приводить до компрометації ключів усіх користувачів, що обслуговуються цим ЦСК [36 – 39]. Нижче наводяться приклади таких компрометацій.

Їх критичність в тому, що компрометація особистого ключа ЦСК приводить до компрометації усіх користувачів. Тому потрібно провести повне відновлення засобом блокування компрометованих особистих ключів як ЦСК, так і користувачів, що є надскладною проблемою [16 – 18].

У ряді джерел наведено приклади таких компрометацій, в тому числі [25]. Так ряд таких веб-додатків, як інтернет-банкінг, розсилка повідомлень, електронна торгівля тощо стали невід'ємною частиною нинішнього життя. У світі як стандарт де-факто, для забезпечення послуг автентичності, цілісності та конфіденційності у вказаних додатках використовуються сертифікати SSL/TLS. Ці сертифікати видаються ЦСК, які вважаються третіми довіреними організаціями. Зокрема, очікується, що ЦСК діють згідно з деякими правилами, які позначені як документи про сертифікаційну політику (Certificate Policy – CP) та Заяви про практики сертифікатів (Cate Practice Statement – CPS). У такій моделі довіри ЦСК мають абсолютну відповідальність за видачу дійсних сертифікатів кожному суб'єкту (користувачу). Проте ЦСК можуть бути скомпрометовані та підроблені, причому чинні сертифікати можуть бути видані через неналежну практику безпеки або невідповідність CP та CPS. Протягом останнього десятиліття відбулися серйозні інциденти через вищезгадані причини, які коротко наводяться нижче [25, 26].

1) Шкідливе програмне забезпечення Stuxnet підписано особистими ключами двох скомпрометованих тайванських ЦСК, мета яких контролювати специфічну промислову систему, яка, ймовірно, є в Ірані, наприклад, газопровід або електростанція.

2) Comodo CA (ЦСК), що має велику частку на ринку SSL, зламано в березні 2011 р. [53]. Один з центрів реєстрації (ЦР) піддався атаці, щоб видати 9 сертифікатів, де зловмисник простежується назад до Ірану.

3) Голландський CA DigiNotar започатковано в липні 2011 р., було видано 531 зловмисний сертифікат для цінних доменів, таких як *.google.com, *windowsupdate.com та *.mozilla.com. Ці сертифікати можуть бути легко використані для поширення зловмисних оновлень Windows або плагінів Firefox без привернення уваги. Щонайменше 300000 унікальних IP-адрес виявлено за допомогою служб Google через ці сертифікати, 99 % трафіку яких надходить з Ірану.

4) Компанія Trustwave Center Authority продала сертифікат для підлеглого CA. Цей під-CA випустив зловмисні сертифікати TLS, якими вони користувалися у внутрішньому трафіку TLS.

5) Турецька компанія CA Turktrust помилково видала сертифікати ЦСК замість сертифікатів TLS у грудні 2012 р. Ці сертифікати використовувались для створення сертифікатів TLS для внутрішнього трафіку. Google визначив зловмисний сертифікат Google через Chrome.

6) Під-ЦСК китайської компанії CNNIC, що знаходиться в Єгипті, видала зловмисні сертифікати TLS для інспекції дорожнього руху в березні 2015 р. Пізніше визначено, що CNNIC експлуатується без документованого CPS.

7) Lenovo Superfish розгорнув місцеві ЦС у своїх продуктах у 2015 р. Цей сертифікат використовується для вставки реклам у веб-сайти, які захищені TLS. Оскільки закриті ключі ЦС знаходяться в оперативній пам'яті комп'ютера, вони можуть бути легко використані для внутрішнього трафіку.

8) У вересні 2015 р. компанія Symantec видала неавторизовані сертифікати для доменів Google. Пізніше Symantec стверджував, що ці сертифікати виготовляються з метою тестування.

9) Symantec придбав Blue Coat у травні 2016 р. Blue Coat має пристрої для перехоплення зашифрованого Інтернет-трафіку. Blue Coat став під-ЦС при Symantec. Ця уніфікація посилила скептицизм.

10) Малайзійський ЦСК DigiCert Sdn. Bhd. помилково випустив 22 слабкі SSL-сертифікати, які можна було використовувати для видавання веб-сайтів і підписання шкідли-

вого програмного забезпечення. У результаті, основним браузерам довелося відкликати свою довіру до всіх сертифікатів, виданих DigiCert Sdn. Bhd.

Існували також проблеми з порушеннями сертифікату TrustWave, великим американським центром сертифікації. TrustWave визнав, що він видав підпорядковані кореневі сертифікати одному з клієнтів, а клієнт зміг контролювати трафік на їх внутрішній мережі. Загроза цього в тому, що підпорядковані кореневі сертифікати дозволяють їх власникам створювати SSL-сертифікати для майже будь-якого домену в Інтернеті. Хоча TrustWave скасував сертифікат і заявив, що він більше не буде видавати підпорядковані кореневі сертифікати клієнтам, він показує, наскільки легко ЦСК може робити помилки і наскільки серйозними можуть бути наслідки цих помилок.

Наведені фатальні випадки призводять до того, що багато досліджень розподіляють абсолютну довіру до ЦС на декілька органів. Для виявлення підроблених, але дійсних сертифікатів TLS [25, 26], застосовується закріплення ключа, краудсорсингу та доведення до браузерів інформації про відкликання тощо. Вказані початкові рішення, які частково реалізовані, на жаль зазнали невдачі через проблеми масштабування.

Основні підходи класичного вирішення проблеми захисту ІВК

У процесі досліджень та удосконалення ІВК було запропоновано два підходи класичного вирішення проблеми ІВК SSL/TLS. Це удосконалення існуючої ІВК на основі системного журналу [26] та застосування децентралізованої мережі однорангової сертифікації, яка отримала назву Мережі довіри (Web of Trust, WoT) [26].

Підхід щодо ІВК, заснований на журналах (лог) [26]. Такий підхід був запропонований в якості нового вирішення проблеми удосконалення традиційних ІВК – його ЦСК. Ідея підходу полягає у використанні загальнодоступних серверів журналів, які контролюють та публікують сертифікати, що видані ЦСК. Такі загальнодоступні журнали забезпечують прозорість, гарантуючи, що лише загальнодоступні сертифікати приймаються та довіряються кінцевим клієнтам. Отже, будь-яка неправильна поведінка ЦСК буде виявлена користувачами та серверами. Прикладом реалізації такого підходу є сертифікати Google Transparency [26], що є найбільш поширенішою ІВК на основі журналу. Зараз такий підхід доступний як у системах Chrome, так і у Firefox. Також відомо багато пропозицій, що дозволяють розширити можливості удосконалених ІВК на основі журналу, в основному це може бути досягнуто за рахунок підтримки відкликання та обробки помилок. Але, на жаль, незважаючи на ці переваги такого ІВК, вони все ще мають кілька проблем, що пов'язані, наприклад, з відкликанням сертифікатів, як пояснено у [25, 26].

Підхід щодо ІВК на основі мережі довіри (Web of Trust – WoT). Підхід заснований на основі децентралізації, при його використанні користувачі можуть визнавати як надійних інших підписувачів, підписуючи у них сертифікати відкритих ключів. У даному випадку кожен користувач має сертифікат, що містить його відкритий ключ і електронні (цифрові) підписи від осіб, які вважають його таким, що заслуговує на довіру. Потім сертифікат завіряється третьою довіреною стороною, наприклад ЦСК, якщо можливо перевірити, що сертифікат містить підпис того, кому є довіра. Такий підхід має перевагу над розподіленим характером довіри, оскільки в цьому випадку усувається будь-яка центральна точка відмови ЦСК (компрометації його особистого ключа). Але такий підхід має недолік, що пов'язаний з ускладненням приєднання нових або віддалених користувачів до мережі. Скоріше всього це пов’язане з тим, що деякі існуючі члени WoT зазвичай повинні особисто зустрітися з новим користувачем, щоб вперше підтвердити свою особистість і підписати відкритий ключ. Крім того, на відміну від підходу, заснованого на ЦСК, при застосуванні WoT виникають проблеми з відкликанням компрометованого ключа. Тобто користувач, який обмежений вибором користувача, на довіру якого він спирається, не може відкликати особистий ключ в разі його втрати або компрометації. Практичні можливості такого підходу залежать від можливостей періо-

дичної відправки у браузери списків відкликаних сертифікатів. В такому разі виникає довіра до недійсного (компрометованого) сертифікату [24 – 26].

Підхід до побудови ІВК, заснованого на БЧ

Аналіз показав, що застосування ТБЧ для створення захищених ІВК надихнуло багатьох дослідників, внаслідок появилось значне число робіт, в першу чергу [26 – 28]. Основний аргумент при обґрунтуванні застосування БЧ полягає в тому, що рішення, засновані на ТБЧ, можуть об'єднати переваги ІВК засновані на журналах, та підході WoT, а також вирішити деякі проблеми зі звичайною системою ІВК. Так, з одного боку, БЧ усуває потенційні точки відмови підходу заснованого на ІВК на основі журналу і проблеми розгортання, які розглянемо нижче. З іншого боку, підхід, що заснований на БЧ, пом'якшує потреби WoT у нових власниках сертифікатів, щоб довести достовірність існуючих членів мережі, а також пом'якшує вимоги WoT для нових власників сертифікатів, щоб довести достовірність існуючих членів мережі.

Згідно з [25 – 28] удосконалення може засновуватись на ТБЧ і інфраструктурі ІВК для управління сертифікатами X.509. Вказане може досягатись на розширенні формату стандартного сертифіката X.509, так щоби він був сумісний з підходом ІВК на основі ТБЧ. Це, по суті, досягається завдяки полям розширення X.509, які використовуються для вбудовування метаданих ТБЧ. Також ІВК на основі БЧ забезпечує надійне керування цифровими сертифікатами.

У першу чергу при удосконаленні ІВК на основі БЧ необхідно обґрунтувати ланцюг довіри.

Як уже розглядалось [24, 16], класичні системи ІВК ґрунтуються на основі ЦСК, які виготовляють та обслуговують сертифікати, що відповідають стандарту X.509. Кожен сертифікат засвідчує право власності на відкритий ключ. Наприклад, коли користувач входить в Twitter через веб-браузер, спочатку веб-браузер перевіряє заявлений сертифікат, який містить відкритий ключ Twitter, перевіряючи ЦСК даного сертифікату. Зазвичай веб-браузери попередньо налаштовані на прийом сертифікатів від певних відомих ЦСК. Для того, щоб сертифікат був довіреним, він повинен бути виданий кореневим центром сертифікації, який існує в довіреному сховищі браузера або пристрої користувача, або допоміжним центром сертифікації, якому довіряють за допомогою підпису кореневого ЦСК. Так нині, як правило, продукти Mozilla постачаються з 154 кореневими сертифікатами [29]. Крім того, фірми Apple, Microsoft і Google мають своє власне сховище довірених кореневих сертифікатів, що вбудовані у їхні продукти.

Зв'язок між конкретним даним сертифікатом і кореневим сертифікатом відомий як ланцюг довіри. При цьому важливо, що ланцюг довіри може включати будь-яку кількість сертифікатів підлеглих ЦСК, тобто ЦСК нижчого рівню. Тому між даним сертифікатом і сертифікатом кореневого СА є зв’язок, а X.509v3 [24] має розширення під назвою Основні обмеження, що може обмежити максимальну глибину дійсного ланцюга сертифікатів (ланцюга довіри) [30].

На рис. 1 [26] наведено шлях сертифікації від сертифіката кінцевого суб'єкта до кореневого ЦСК, де починається ланцюг довіри. Отже, якщо сертифікат кінцевого об'єкта не був виданий довіреним ЦСК, веб-браузер потім перевірить, чи був сертифікат ЦСК випущений довіреним ЦСК, і т.д. поки не буде знайдено довірений ЦСК. За цієї умови браузер зазвичай відображає помилку.

Особливості застосування технології БЧ. Розподілений реєстр, тобто БЧ, позитивно розглядається завдяки успіху в застосуванні у Bitcoin. Нині більшість БЧ-платформ використовуються у фінансових додатках, однак починають з'являтися все більше нових додатків для різних сфер. Звичайно це додатки, що вимагають високої надійності і повного усунення ризиків маніпулювання даними. Також БЧ є розподіленим, тому він не має вразливостей, що пов’язані з одиничною точкою відмови.

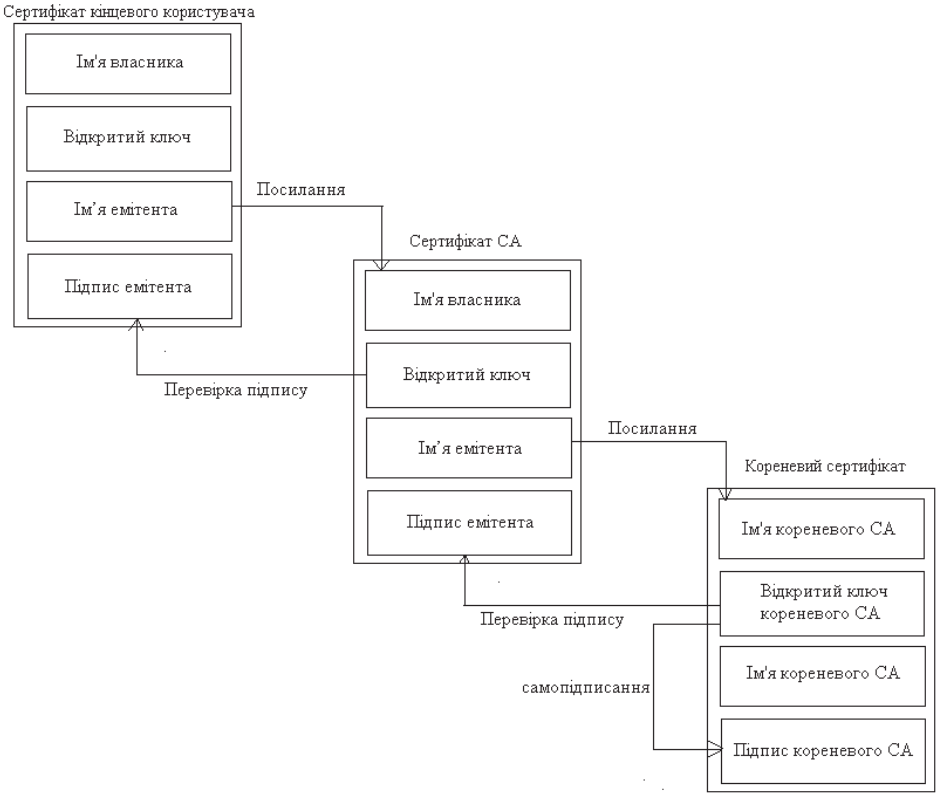


Рис. 1. Ланцюг традиційної довіри

Наразі розроблено БЧ, що дозволяють виконувати довільну логіку, відому як смарт-контракти. У загальному смарт-контракт – це програма, яка виконується поверх БЧ і використовує базовий порядок транзакцій для забезпечення узгодженості результатів виконання смарт-контракту між партнерами (peers) [32]. Так, наприклад, БЧ Ethereum підтримує складну та повну за Тьюрингом мову Solidity (http://solidity.readthedocs.io/en/develop/), яка може бути використана для програмування та визначення широкого кола сценаріїв застосування [32].

Також, щодо ІВК технологія БЧ надає такі важливі засоби безпеки як відкликання сертифікатів, усунення центральних точок збою і надійний запис транзакцій. Наприклад, при швидкому відкликанні сертифікатів ІВК, що засновані на БЧ, можна миттєво ізолювати інфікований ЦСК і відповідні сертифікати без очікування наступного оновлення списків відкликаних сертифікатів (CRL). Крім того, ІВК на основі ТБЧ, як відкритий журнал тільки для додавання (запису), природно надає властивість прозорості сертифікату, що запропоновано в [27]. Тому при подальших дослідженнях необхідно вибрати платформу, наприклад, Ethereum і мову програмування смарт-контрактів, наприклад, Solidity. Це можна пояснити тим, що вони мають велике співтовариство розробників програмного забезпечення з відкритим вихідним кодом. Це робить процес розробки програмного забезпечення набагато більш ефективним.

Основні дослідження щодо реалізації БЧ для побудови системи ІВК. Реалізація БЧ для побудови системи ІВК ретельно вивчалась дослідниками та розробниками. Так, в [33] автори пропонують Blockstack, який використовує для надання системи реєстрації імен, реалізацію БЧ біткойну, де імена пов'язані з відкритими ключами.

Подібно до Blockstark, ІВК на основі БЧ реалізується в Emercoin (https://emercoin.com/en/tech-solutions/) (проект EmerSSH). Emercoin – це загальнодоступний ТБЧ, досить близький до біткойну з точки зору архітектури, що включає в себе гібридний консенсус доказу роботи і доказу ставки (в залежності від доступності майнингових потужностей). Emercoin не має смарт-контрактів і зберігає тільки ґеш-значення сертифікатів у ТБЧ. Сам EmerSSH зберігає тільки ґеш-значення сертифіката для ТБЧ, що на думку авторів, зме-

ншиться ризик «людина посередині». Це досягається тим, що як тільки ґеш-значення сертифіката завантажується в БЧ, встановлюється безпечне з'єднання за допомогою відкритого ключа сертифікату і обмін для кожного з'єднання проводиться абсолютно новими ключами.

Спеціалісти Fromknecht та інші [28] пропонують для реалізації ІВК для зберігання доменів та пов'язаних з ними відкритих ключів використовувати БЧ Certcoin.

Аналіз показав, що усі згадані дослідження пропонують підходи, що засновані виключно на БЧ. У роботі [26] пропонується використовувати загальний стандартний сертифікат X.509v3 з незначним доповненням до полів розширення з інформацією, що пов'язана з ТБЧ [30]. При цьому розширений сертифікат X.509 може бути перевірений класичним ланцюгом довіри на основі ЦСК або з використанням структури ІВК на основі ТБЧ. Вказані автори, на наш погляд, першими запропонували такий гібридний сертифікат.

Структура ІВК на основі БЧ

У цьому підрозділі аналізується структура для управління ІВК на платформі БЧ [26].

Основні можливості захисту. Структура ІВК на основі БЧ підтримує відкликання сертифікату, що є суттєвою проблемою в традиційних системах ІВК. Також, оскільки неможливо видалити інформацію з БЧ, то тільки батьківський ЦСК може позначати виданий ним сертифікат як відкликаний. Тобто, будь-яка неправильна поведінка ЦСК стосовно відкликання сертифікату буде також простежена і помічена всіма іншими суб'єктами.

Особливості проектування та застосування (методологія проектування). Структура ІВК [26] на основі БЧ базується на гібридних сертифікатах X.509, як це показано на рис. 2. Сертифікат містить певну інформацію про середовище ІВК в полях розширення. Значення полів розширення наступні:

- Ідентифікатор ключа суб'єкта: зберігає особистість власника сертифіката.

- Ім'я БЧ: містить назву платформи ТБЧ. Наразі використовується загальнодоступний ТБЧ Ethereum, але потрібно охопити більше платформ.

- Ідентифікатор ключа ІВК: містить адресу смарт-контракту поточного ЦСК, якщо це сертифікат ЦСК. Для сертифікатів не ЦСК поле порожнє.

- Ідентифікатор ЦСК емітента: має адресу смарт-контракту ЦСК, що видав цей сертифікат. Дозволяє валідатору знайти смарт-контракт батьківського ЦСК в ТБЧ і перевірити, чи сертифікат з відповідним ґеш-значенням був виданий і не був відкликаний.

Для кореневих сертифікатів це поле порожнє.

- Алгоритм ґешування: містить інформацію про алгоритм ґешування, який використовувався при обчисленні ґеш-значення сертифікату, завантаженого в ТБЧ.

Таким чином, згідно [26] структура передбачає три типи сертифікатів: сертифікати кореневого ЦСК (Root-CA), під-ЦСК (Sub-CA) і кінцевого користувача (Enduser). У табл. 1 наведено ієрархію гібридних сертифікатів БЧ. У першому рядку представлений кореневий ЦСК, в якому сертифікат випущений і підписаний ним самим (самопідписаний). ЦСК емітента відсутній, що підтверджено ID емітента ЦСК (в п'ятому стовпці 0x00000000). Сертифікат під-ЦСК подається у другому рядку – він був виданий кореневим ЦСК, а ID емітента ЦСК вказує на кореневий ЦСК. Останній рядок містить сертифікат кінцевого користувача, що виданий під-ЦСК. У наступному розділі ми пояснюємо, як ми реалізували цю структуру на платформі ТБЧ.

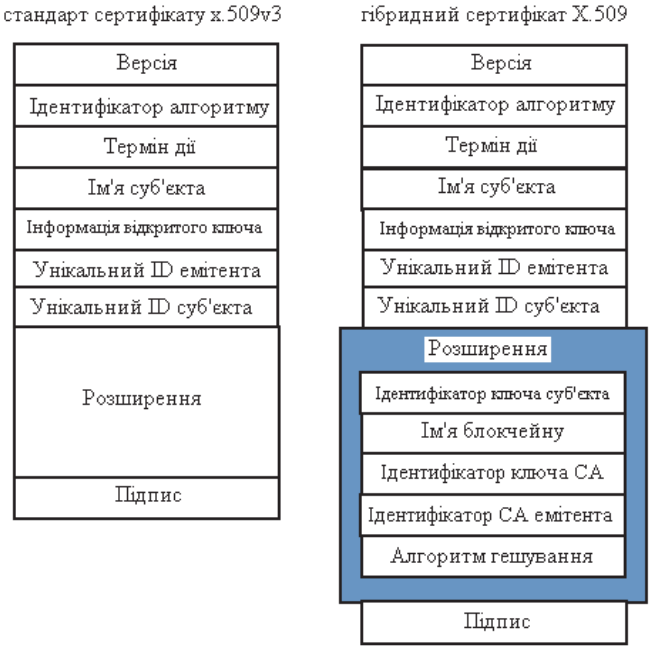


Рис. 2. Стандартний і гібридний сертифікат X.509

Таблиця 1

Ієрархія гібридних сертифікатів ТБЧ

Cert.	Issued By	Issued To	CA Contract ID	Issuer CA ID
RootCA	RootCA	RootCA	0x1234xxxx	0x00000000
SubCA	RootCA	SubCA	0x5631xxxx	0x1234xxxx
EndUser	SubCA	End user	-	0x5631xxxx

Архітектура ІВК БЧ. Основна ідея структури згідно [26] полягає в тому, що кожен WCR має спеціальний смарт-контракт, який містить наступну інформацію:

- Масив з ґеш-значеннями виданих сертифікатів, а також може містити дату закінчення терміну дії кожного сертифікату та іншу технічну інформацію.

- Відображення даних про відкликання, на які посилається ґеш-значення сертифікату. Якщо сертифікат відкликано, ЦСК, який видав цей сертифікат, додає дані відкликаного сертифікату.

Крім того, якщо сертифікат є сертифікатом ЦСК, то він також завантажується у відповідний смарт-контракт ЦСК. Далі, оскільки сертифікат містить адресу смарт-контрактів батьківського CA, то він дозволяє перевіряти все дерево центру сертифікації, тобто ланцюг довіри від користувача до кореневого сертифіката.

Реалізація ІВК заснованого на БЧ. Запропонована структура ІВК, заснована на БЧ, включає три основні частини – для тестування в основному використовувалися служба звільнення, перевірки сертифікатів та веб-інтерфейс користувача. Їх призначення у наступному [26].

1) Служба звільнення (Restful service). Розроблена разом з Golang як окремий веб-сервер, що забезпечує доступ до загальнодоступного БЧ Ethereum. Вона надає всі можливості видачі, відкликання та перевірки сертифікатів. Важливо відзначити, що перевірка проводиться «безкоштовно» з точки зору витрат криптовалюти загальнодоступного БЧ, оскільки перевірка не додає або не змінює дані в БЧ.

Служба звільнення (Restful service) надає такі основні функції:

- Реєстрація користувача. Додає ґеш-значення сертифікату до смарт-контракту даного ЦСК. У якості альтернативи надання ґеш-значення як параметру, сертифікат може бути

завантажений в службу Restful. У цьому випадку ґеш-значення обчислюється на основі завантаженого сертифіката.

- Чорний список користувачів: відкликає сертифікат, тобто переміщує сертифікат (звичайний або ЦСК) з білого списку до чорного списку. Технічно це досягається шляхом додавання посилання на ґеш-значення сертифікату до відображення відкликання у смарт-контракті.

- Створення контракту. Створює порожній контракт для нового ЦСК. Викликається батьківським ЦСК при видачі сертифіката для його під-ЦСК.

- Заповнення договору. Після створення порожнього смарт-контракту для під-ЦСК батьківський ЦСК повинен завантажити до нього сертифікат під-ЦСК, що містить адресу батьківського смарт-контракту та адресу смарт-контракту під-ЦСК у полях розширення. Після заповнення смарт-контракту під-ЦСК з сертифікатом його батьківського ЦСК, адреса облікового запису Ethereum під-ЦСК записується в змінну власника смарт-контракту, забезпечуючи таким чином доступ на запис тільки для під-СА.

- Перевірка-cert (перегляд/постійна функція). Перевірка сертифікату з листа до кореня дерева ЦСК. Важливо, що перевірка сертифікатів може бути проведена службою Restful, яка базується на коді Golang або на основі виклику окремого смарт-контракту перевірки. При цьому перевірка проводиться при нульовій вартості.

Перші чотири функції функціональності служби Restful передбачають авторизацію користувача Ethereum, що відповідає батьківському ЦСК. Важливою особливістю функціональності служби Restful є навмисне багатоступеневе ініціювання реєстрації під-ЦСК (додавання сертифіката під-ЦСК до списку затверджених сертифікатів в смарт-контракті батьківського ЦСК). На відміну від традиційного сертифіката кінцевого користувача, який ініціюється тільки за допомогою функції «Зареєструвати користувача служби Restful», сертифікат під-ЦСК запускається за допомогою наступних кроків.

- Батьківський ЦСК повинен створити порожній смарт-контракт для під-ЦСК з функцією Створити-контракт. Тепер батьківський ЦСК може генерувати гібридний сертифікат, що вводить нову адресу смарт-контракту у відповідне поле розширень сертифікату.

- Батьківський ЦСК заповнює новий смарт-контракт під-ЦСК згенерованим сертифікатом, використовуючи функцію «Заповнити контракт служби Restful». Після виконання функції «Заповнити контракт» права на написання нового смарт-контракту передаються виключно до під-ЦСК з заповненням адреси під-ЦСК Ethereum у поле власника нового смарт-контракту.

- Ґеш-значення смарт-контракту фіксується в білому списку батьківського ЦСК з функцією «Зареєструвати користувача».

2) Перевірка: Модуль перевірки містить смарт-контракт, який дозволяє перевірити ланцюг довіри для даного сертифікату (шлях від листа або сертифікату кінцевого об'єкта до кореня в дереві ЦСК. Важливо відзначити, що перевірка смарт-контракту не залежить від перевірки коду Golang у службі Restful, альтернативний підхід до перевірки сертифікатів також може бути реалізовано у такій структурі. Причому, обидва підходи до перевірки не передбачають жодних виплат криптовалюти, оскільки ТБЧ не змінюється.

3) Веб-інтерфейс користувача: Інтерфейс користувача дозволяє клієнтам перевіряти весь додаток – додавати сертифікати ЦСК та сертифікати кінцевого користувача на різних рівнях дерева ЦСК під різними обліковими записами ЦСК, відкликати сертифікати тощо. Очевидно, що веб-інтерфейс може розглядатися як оболонка для згаданої вище служби Restful. Варто зазначити, що тестовий веб-інтерфейс має свій власний смарт-контракт, який зберігає деякі дані, включаючи посилання від батьківського ЦСК до смарт-контрактів його під-ЦСК. Це дозволяє переходити від кореня до листів (сертифікатів кінцевого об'єкта) дерева сертифікатів, але за умов, що даний ланцюг довіри був завантажений з веб-інтерфейсом.

Переваги ІВК на основі БЧ. Аналіз показав, що ІВК на основі БЧ має перед традиційною ІВК наступні переваги:

- перевірка сертифікату і його ланцюга сертифікатів ЦСК проста і швидка;

- ІВК на основі ТБЧ вирішує давню проблему традиційних ІВК, не вимагаючи використання сервісу, який видає списки відкликання сертифікатів (CRL). Це здійснюється завдяки синхронізації ТБЧ між вузлами мережі, де будь-яка модифікація стану сертифіката буде миттєво повідомлена на всі вузли [34].

Іншим важливим аспектом в контексті безпеки Інтернету є те, що ІВК, що базується на БЧ, надає гнучкий захист від атак "людина посередині" (MITM). Традиційно MITM розглядається як серйозний ризик для безпеки, що передбачає, що зловмисник може захопити з'єднання веб-браузера для певних веб-сайтів, представивши дійсний сертифікат (тобто підроблений відкритий ключ) для цього домену. Для користувачів і веб-браузерів важко визначити заміну сертифіката у випадку, якщо зв'язаний ЦСК був зламаний зловмисником [35]. Підхід ІВК, що заснований на БЧ, робить атаки MITM практично неможливими. Це пояснюється тим, що коли ЦСК публікує або відкликає відкритий ключ веб-сайту/домену на БЧ, інформація буде розподілена по тисячам вузлів. Таким чином, порушення відкритого ключа буде (теоретично) поза питанням [36]. Традиційна ІВК усуває ризики MITM шляхом вбудовування сертифікатів кореневого ЦСК в інсталяцію браузера, таким чином штучно розширюючи бар'єри входу ЦСК і збільшуючи час, необхідний для відкликання сертифіката кореневого ЦСК.

Оцінка та експериментальні результати

Нижче подаються результати, що стосуються оцінки продуктивності, складності (вартості) функцій, заснованих на БЧ та обмеження щодо платформи ІВК на основі ТБЧ [26].

Продуктивність. Щоб визначити ефективність ІВК на основі смарт-контрактів Ethereum, в [26] проведено ряд експериментів на відкритому Ethereum Testnet (Rinkeby) (https://www.rinkeby.io). Сутність його в перевірці сертифікатів ЦСК по повному шляху від листа (даний сертифікат ЦСК) до кореня (ланцюг довіри). У результаті експерименту зроблено порівняння продуктивності між перевіркою на основі смарт-контракту та перевіркою коду Golang служби Restful.

Перевірка служби Restfu. Спочатку була зроблена перевірка сертифікату, хоча повний ланцюг довіри був заснований на службі Restful. Вона отримує сертифікат для кожного ЦСК з БЧ, аналізує сертифікат з бібліотеками Golang для вилучення адреси смарт-контракту батьківського ЦСК і потім перевіряє дійсність сертифікату на основі відповідного ґеш-значення, що зберігається у смарт-контракті батьківського CA.

Перевірка на основі смарт-контракту. Альтернативний підхід, що застосований, виявляється більш значно ефективнішим. Ідея його полягає в тому, що спеціальний смарт-контракт читає і аналізує сертифікати ЦСК, що зберігаються в БЧ виключно за допомогою коду Solidity та компілятора для смарт-контракту Ethereum. Зокрема, оскільки смарт-контракт не змінює ТБЧ, перевірка здійснюється «безкоштовно».

Із рис. 3 видно, що хоча продуктивність смарт-контрактів може бути менш вражаючою порівняно з криптографічними бібліотеками Golang, заснованими на відносно коротких ланцюгах довіри (менше 400 під-ЦСК), починаючи від ланцюга довіри довжиною, що перевищує 500 під-ЦСК, продуктивність перевірки на основі смарт-контракту вище, ніж у коду Golang.

Наприклад, для ланцюга довіри з довжиною близько 1100 під-ЦСК перевірка на основі смарт-контракту тривала приблизно 7 с, тоді як для коду Golang з використанням криптографічних бібліотек Golang було потрібно майже 15 с. У [26] для експериментів була використана стандартна робоча станція DELL з процесором Intel Core i7, але з оперативною пам'яттю 32 Гб.

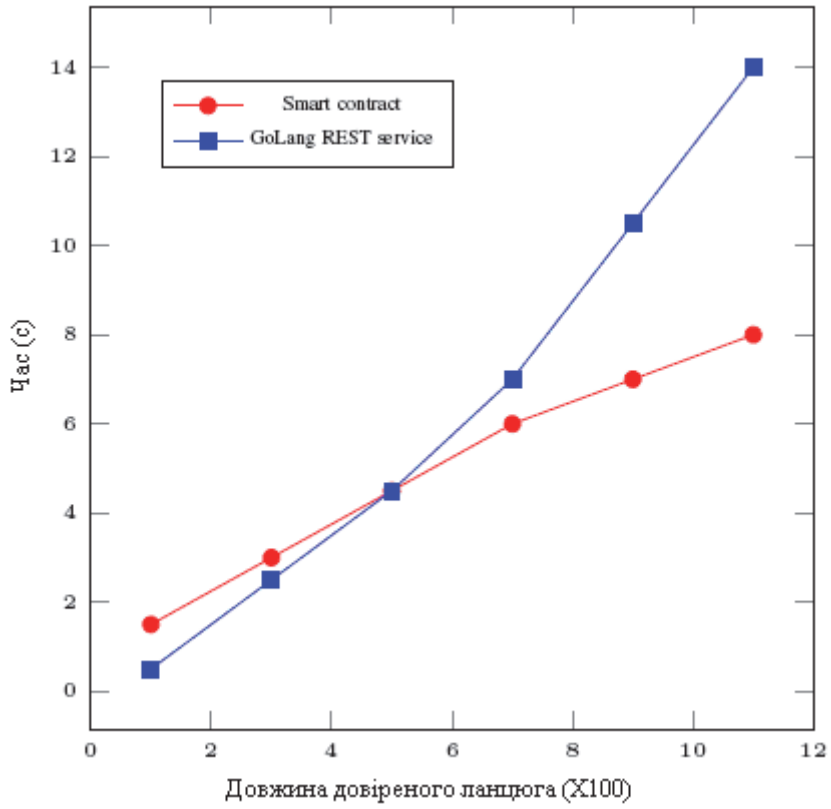


Рис. 3. Перевірка ланцюга довіри сертифіката CA

Витрати на запуск нового ЦСК в Ethereum

Витрати на запуск. Як показано в [26], витрати на підтримку ІВК є, можливо, важливою перевагою рішень ІВК на основі ТБЧ. Якщо припустити, що кінцеві користувачі все одно будуть використовувати локальну копію загальнодоступного БЧ, то усі витрати в основному складаються з плати майнерам, які підтверджують запис даних в БЧ.

Виходячи з експериментів з тестовим загальнодоступним ТБЧ Rinkeby [26], запуск ЦСК, (включаючи створення порожнього смарт-контракту, завантаження сертифікату в цей смарт-контракт, запис ґеш-значення цього сертифіката в смарт-контракт батьківського CA тощо) коштує 0,07 Ethers, що при нинішній вартості Ethers близько 700 USD за Ether переводиться у 50 USD за сертифікат ЦСК.

Можливо ініціювання звичайного сертифіката кінцевого суб'єкта, що передбачає лише записування його ґеш-значення у смарт-контракт батьківського ЦСК, призводить до набагато менших витрат у розмірі 7-10 USD за сертифікат. З огляду на поточну ціну на вихідний річний сертифікат кінцевого суб'єкта в розмірі декількох сотень доларів, витрати на сертифікат ТБЧ, схоже, не перевищують витрати, витрачені на існуючу інфраструктуру ЦСК.

Обмеження та проблеми створення та застосування. По-перше, загальнодоступні БЧ характеризуються значним збільшенням розміру БЧ, що повторюється на всі вузли, які беруть участь в системі. Особливо це стосується Ethereum та інших подібних платформ з підтримкою смарт-контрактів, які важливі для організації ефективної ІВК. Наприклад, у грудні 2017 р. розмір Ethereum ChainData з FAST Sync досяг 38,89 Гб порівняно з 20,46 Гб у вересні 2017 р. (https://etherscan.io/chart2/chaindatasizefast) [62].

По-друге, велика волатильність криптовалют призводить до певної невизначеності витрат на завантаження/оновлення сертифікатів як в довгостроковій, так і в короткостроковій перспективі. Інакше кажучи, вартість роботи БЧ безпосередньо пов'язана з ціною відповідних криптовалют, таких як Ether. Наприклад, у травні 2017 р. ціна Ефіру склала 85,43 доларів (https://bitcoinmagazine.com/price/), тоді як у грудні 2017 р. ціна досягає 729,01 доларів, що має на увазі зростання вартості операцій з ТБЧ в вісім разів за сім місяців.

По-третє, якщо для перевірки сертифіката використовується синтаксичний аналіз смарт-контракту з кодом Solidity, а не зовнішнім модулем Golang (тобто служби Restful), то виникають обмеження у використанні ґеш-функцій і асиметричних криптографічних функцій, доступних для смарт-контрактів. Наприклад, для Ethereum без попередньої обробки даних можна використовувати тільки SHA256 як ґеш-функцію і підписи ECDSA на основі криптографії еліптичних кривих.

Нарешті, оскільки права доступу для модифікації даних сертифікату засновані на системі облікових записів платформи БЧ, то втрачений пароль призводить до безповоротно втраченого доступу до облікового запису. Деяким рішенням проблем ЦСК, які втрачають свій пароль доступу до ТБЧ, може бути організація порожнього смарт-контракту і копіювання всіх даних зі старого смарт-контракту в новий, так як теоретично будь-який смарт-контракт завжди доступний для читання будь-кому. Очевидно, що створення нового смарт-контракту ЦСК може призвести до перевидання сертифіката ЦСК (принаймні в поточній реалізації), оскільки сертифікати ЦСК можуть містити посилання на відповідний смарт-контракт.

Нова модель ІВК з прозорістю сертифікатів на основі БЧ

Загальні положення та стан. У традиційних ІВК ЦСК вважаються повністю довіреними. Однак на практиці абсолютна відповідальність ЦС за забезпечення надійності викликала серйозні проблеми безпеки інформації та кібербезпеки. Щоб запобігти подібним проблемам, компанія Google у 2013 р. впровадила концепцію прозорості сертифікатів (ПС). Пізніше для зниження рівня довіри до ЦС запропоновано кілька нових моделей ІВК. Наприклад, підзвітна інфраструктура ключа (ПІК), стійка до атак інфраструктура відкритого ключа (CAIBK) та розподілена прозора інфраструктура ключа (РПІК). Проте, всі ці пропозиції все ще є вразливими до атак розділеного цифрового світу, якщо порушник здатний реалізувати різні плани загроз на журнал. У роботі [25] щоб усунути атаки розділеного світу та забезпечити ідеальну прозорість сертифікації та відкликання сертифікатів пропонується нова архітектура ІВК з прозорістю сертифікатів, заснована на БЧ, яка була названа CertLedger. Всі сертифікати TLS (Transport Layer Security), їх статус відкликання, весь процес відкликання та довірене управління ЦСК здійснюються в CertLedger. CertLedger надає унікальний, ефективний і надійний процес валідації сертифікату, що виключає звичайні непридатні та несумісні процеси сертифікації, що реалізуються різними постачальниками програмного забезпечення. Клієнти TLS в CertLedger також більше не вимагають перевірку достовірності та зберігання довірених сертифікатів ЦСК.

Як стандарт де-факто, сертифікати SSL/TLS використовуються для забезпечення послуг автентичності, цілісності та конфіденційності для цих існуючих додатків. Ці сертифікати видаються ЦСК, які вважаються довіреними організаціями у звичайних системах ІВК. Зокрема, очікується, що ЦСК діють згідно з деякими правилами, які позначені як документи про сертифікаційну політику (Certificate Policy – CP) та заява про практики сертифікатів (Cate Practice Statement – CPS). У поточній моделі довіри ЦСК мають абсолютну відповідальність за видачу правильних сертифікатів для визначеного суб'єкта. Проте ЦСК все ще можуть бути скомпрометовані та підроблені, або чинні сертифікати можуть бути видані через неналежну практику безпеки інформації або невідповідність CP та CPS. Необхідно відмітити, що протягом останнього десятиліття відбулися серйозні інциденти, які наведені вище в розділі 2 цієї статті. Вказані фатальні випадки призводять до того, що багато досліджень здійснюють абсолютну довіру до ЦСК на декілька органів. Для виявлення [25] підроблених, але дійсних сертифікатів TLS, закріплення ключа, краудсорсингу, донесення до браузерів інформації про відкликання є початковими рішеннями, які частково реалізовані, але зазнали невдачі через проблеми масштабування.

Розподілена прозора інфраструктура ключа (РПІК) [25]. Інфраструктура РПІК визначає загальнодоступну архітектуру управління сертифікатами, яка зменшує вразливості та, як наслідок, запобігає використанню підроблених сертифікатів та виявляється стійкою, навіть,

якщо всі постачальники послуг діють у домовленості [37]. Здійснюється обслуговування журналу сертифікатів (ОЖС) та обслуговування журналу відображення (ОЖВ) – двох нових об'єктів, що введені у РПІК. Причому ОЖС зберігають всі дійсні, відкликані та не чинні сертифікати для набору доменів і надають докази щодо їх існування або відсутності. Також ОЖВ підтримує зв'язок між набором доменних імен та ОЖС, які підтримують журнали для них. «Дзеркала» підтримують повну копію даних, що зберігаються як в ОЖС, так і ОЖВ. Причому ЦСК здійснюють перевірку ідентифікаторів та видають сертифікати, але вони не є єдиними суб'єктами, що забезпечує довіру при підключенні до домену. Якщо взяти концепцію «незалежного ключа», то домен володіє двома типами сертифікатів – сертифікатом TLS і майстер-сертифікатом, який використовується для запиту нового сертифіката TLS від ЦСК, і реєстрації його в ОЖС. Користувачі або, зокрема, браузери спершу роблять запит до ОЖВ для того, щоб знайти правильний ОЖС для конкретного домену. Для прийняття рішення про підключення перші доведення, отримані від ОЖВ, перевіряються, далі робиться запит до ОЖС для того, щоб отримати докази для сертифікату TLS домену.

У РПІК передбачається, що всі майстер-сертифікати є справжніми, і випуск підроблених сертифікатів не є вірогідним, оскільки ЦСК працюють на підприємствах, які не можуть втратити репутацію. Однак це не є дійсним аргументом, оскільки більшість підроблених сертифікатів генеруються через відсутність належного контролю або процесів безпеки. А саме, якщо ЦСК і ОЖС піддаються компрометації, РПІК не зможе запобігти випуску підроблених майстер та TLS сертифікатів. З цієї точки зору, порушник, який контролює ОЖС та здатний здійснювати підробки, використовуючи дійсні майстер або TLS сертифікати, може зробити атаку розділеного цифрового світу. На жаль, цю атаку не можна виявити, оскільки в РПІК не має процесу моніторингу через припущення про справжні сертифікати.

Прозорість сертифікату та відкликання на основі ТБЧ. Ванг та інші запропонували прозорість сертифікації та відкликання на основі БЧ для зберігання сертифікатів TLS та їх статусу відкликання [38]. Коротше кажучи, у цій схемі веб-сервери публікують свої сертифікати TLS у БЧ, використовуючи їхні пари ключів публікації, які використовуються для підписання операцій. Ці пари ключів видачі відрізняються від пари ключів у сертифікаті та спочатку засвідчуються певним набором веб-серверів, які вже існують в БЧ. У цій схемі транзакції мають термін дії, тому сертифікати TLS та їх статус відкликання додаються до БЧ періодично протягом свого терміну дії. Під час рукостискання TLS веб-сервер посилає транзакцію сертифікату та свій шлях аудиту Мерклі до клієнта TLS, який перевіряє його дійсність через свої локально збережені заголовки синхронізованих блоків.

Однак ця пропозиція має такі недоліки. Вона має ненадійну основу для забезпечення надійності ключів публікації. А саме, «сильний порушник», який може отримати підроблені, але дійсні сертифікати TLS від пошкоджених ЦСК, може заздалегідь створити деякі фіктивні домени (тобто веб-сервери) і може використовувати їх для створення дійсного підпису транзакції пари ключів публікації. Ця проблема виникає через довіру до веб-серверів. Автори [25] пропонують вирішити цю проблему, створивши більше публічно-довірених ЦСК, щоб визнати недійсними підроблені транзакції. Однак тоді постає питання рівня довіри, яке явно не уточнюється. Причому, прозорість відкликання покладається на ЦСК, щоб опублікувати дані про анулювання сертифікатів TLS на ТБЧ. Однак скомпрометовані або непрацюючі ЦСК не можуть видавати СВС або давати відповідь клієнту в зазначений час.

Щодо атак людина-посередині, коли порушник здатен переконати клієнта у незавершеній транзакції відкликаного TLS сертифікату, то під час рукостискання TLS веб-сервери передають клієнту TLS транзакцію сертифіката для підтвердження сертифікату TLS. Клієнт TLS приймає цю транзакцію, якщо термін її дії не закінчився, і додає її до затвердженого блоку. Однак відкликаний або оновлений TLS сертифікат також може мати незавершену транзакцію у ТБЧ. Тому, як тільки порушник надсилає цю незавершену транзакцію сертифіката зі своїми доказами Мерклі клієнту TLS, він приймається під час рукостискання TLS.

Клієнти TLS не можуть виявити остаточний стан сертифікату, оскільки клієнти лише перевіряють наявність транзакції у відповідному блоці.

Така пропозиція, з точки зору витрат на зберігання, також є неефективною. По-перше сертифікат TLS додається до ТБЧ періодично протягом його терміну дії, по-друге CBC може бути доданий до ТБЧ для кожного відкликаного сертифікату (тобто кількість вставок CBC до ТБЧ дорівнює кількості відкликаних сертифікатів), а по-третє пара ключів публікації додається до ТБЧ періодично. Крім того, він має заголовки великого розміру, які містять імена DNS, існуючі в транзакціях блоку.

Як публічний ТБЧ вирішує проблеми ІВК. По суті БЧ – це спільний, незмінний, децентралізований відкритий журнал, що містить постійно зростаючий список блоків. Блок – це структура даних, яка складається з заголовку і списку транзакцій. Кожен блок пов'язаний з попереднім блоком за рахунок криптографічного ґеш-значення, тому блоки по своїй суті захищені від підробки та перегляду. Мережа ТБЧ – це децентралізована однорангова (P2P) мережа, що складається з повних вузлів і легких вузлів. Повні вузли зберігають копію ТБЧ, перевіряють і розповсюджують нові транзакції і блоки по всій мережі, тоді як легкі вузли зберігають тільки заголовки блоків. Всі вузли можуть створювати транзакції для зміни стану ТБЧ. Нові блоки транзакцій колективно перевіряються і додаються до існуючого ланцюга відповідно до розділеного механізму консенсусу.

У [25, 39] показано, що БЧ вирішує такі проблеми ІВК:

1) атаку розділеного світу;

2) проблеми з відкликанням та перевіркою сертифікатів;

3) проблеми управління довіреними сертифікатами/сховищами ключів.

Характеристики ТБЧ для ІВК. Використовуючи послідовність прийняття рішень Wust і Gervais [25], можна визначити тип БЧ для керування журналом сертифікації. Будемо вважати, що записувач – це сутність, яка здатна накопичувати нові транзакції в новий блок і додавати його в ТБЧ.

Для повноти викладення необхідно дати відповіді га такі питання:

1. Чи потрібно зберігати стан сертифікатів?

Сертифікати TLS постійно оновлюються через закінчення терміну дії або скасування. Стан сертифікатів повинен зберігатися та оновлюватися тоді, коли це необхідно.

2. Чи існують кілька авторів?

Сертифікати TLS, створені довіреними ЦСК, додаються до журналу. Окремий підроблений записувач може додавати підроблений, але дійсний сертифікат до журналу, затримувати або ігнорувати додавання справжніх. Тому збільшення децентралізації під час запису до журналу зменшить ризик підробок через те, що широка участь записувачів призведе до більш надійного і стійкого журналу.

3. Чи можете ви використовувати третю довірену сторону (ТДС), яка завжди онлайн?

Сильний супротивник може керувати будь-якою ТДС, що може призвести до єдиної точки відмови. Прийняття онлайн-ТДС є основним джерелом вразливості.

4. Чи відомі всі записувачі?

Записувачі можуть бути відомі або невідомі. Проте, якщо вони відомі, їх слід вибирати та розганяти по всьому світу таким чином, щоб їх зловмисна співпраця та маніпуляції не могли бути можливими.

5. Чи усі записувачі є довіреними?

Незважаючи на те, що всі записувачі, здається, є довіреними, деякі з них можуть контролюватися сильним зловмисником.

6. Чи потрібна громадська перевірка?

Статус існування та дійсності всіх сертифікатів TLS повинен бути перевірений громадськістю для досягнення повної прозорості. Таким чином, блок-схема рішень призводить до ТБЧ без дозволу або публічного ТБЧ з дозволом для керування журналами сертифікації. Однак ТБЧ вимагає наступні додаткові можливості. Перш за все, він повинен містити інфра-

структуру смарт-контрактів для реалізації необхідних правил перевірки переходу станів. По-друге, базовий механізм консенсусу не повинен призводити до тимчасових розгалужень, оскільки деякі клієнти TLS можуть перевірити неправильний стан сертифіката TLS до того, як блоки будуть повністю підтверджені. По-третє, час, необхідний для підтвердження нового блоку в механізмі консенсусу, не повинен бути високим, щоб транзакції могли змінювати стан сертифікатів TLS за прийнятний період часу. Нарешті, архітектура БЧ повинна дозволяти клієнтам TLS перевірити остаточний стан сертифікатів TLS і ефективно генерувати докази Мерклі. Відмітимо, що стан дерев Мерклі в основному підтримується для ефективного вироблення підтвердження та слідкування за кінцевими станами активів. Корінь Мерклі цього дерева зберігається в заголовках блоків, тому цілісність дерева і створені з нього докази стану можна перевірити [16 – 18].

Необхідно відмітити, що CertLedger можна розгорнути на існуючій архітектурі БЧ, що задовольняє вимогам [25], як у Ethereum, Neo та Ontology. У цих архітектурах можна обрати будь-який механізм консенсусу, який не призведе до тимчасових розгалужень, таких як PBFT та DBFT [25].

Призначення та сутність CertLedger. CertLedger – це архітектура ІВК для перевірки, зберігання та анулювання сертифікатів TLS та керування довіреними сертифікатами ЦСК у публічному БЧ. Вона спрямована на те, щоб забезпечити більш прозорий життєвий цикл видачі та відкликання сертифікатів та усунути будь-які види атак «суб'єкта посередині». Більш того, він також має на меті уніфікувати процес валідації сертифікатів для всіх клієнтів TLS, оскільки реалізації різних клієнтів TLS не є узгодженими та відповідними.

CertLedger управляє функціями ІВК через об'єкти стану. Об'єкт стану – це цифровий документ, який складається з даних і незмінного коду смарт-контракту для управління ним. Кожен об'єкт стану має унікальну адресу в БЧ. Державні зміни активів ініціюються операціями та відстежуються через державні об'єкти. CertLedger, як правило, містить наступні об'єкти стану.

Об'єкт доменного стану зберігає та управляє станами всіх сертифікатів TLS та їх статусом відкликання. Цей об'єкт стану містить необхідний код для перевірки сертифіката TLS відповідно до міжнародних стандартів, таких як RFC 5280 [40]. Він використовує Об'єкт Стану Довіреного ЦСК, при побудові надійного шляху для сертифікату TLS. Крім того, він також містить необхідний код для зміни статусу сертифіката TLS на «відкликаний». Його смарт-контракт перевіряє сертифікат TLS, додаючи до CertLedger у наступній послідовності перевірки:

- чи сертифікат вже додано;
- чи є діючим сертифікат;
- чи сертифікат відповідає профілю сертифіката TLS;
- чи підписаний сертифікат одним із сертифікатів ЦСК в Довіреному ЦСК;
- чи зберігати новий сертифікат TLS у об'єкті стану домену;
- чи встановлено його статус скасування як «не відкликано».

Проблемні питання сучасних та перспективних ІВК на базі БЧ

Проблеми сучасних ІВК. Проведений аналіз показав, що стосовно сучасних ІВК з ЦСК існують наступні проблеми:

1) Єдина точка збою. ЦСК несуть повну відповідальність за відкликання сертифікатів та надання послуг по їх скасуванню. Якщо ЦСК несправний або скомпрометований, то вся система стає під загрозу, а по суті компрометується.

2) Необхідність застосування третьої довіреної сторони. Причому користувачі систем повинні довіряти ЦСК, адже він відповідає за генерацію та управління відкритими ключами користувачів. В разі компрометації ЦСК існує високий ризик безпеки системи, що використовує сертифікати відкритих ключів.

3) Висока вартість та неефективність управління ключами при великій кількості значно розподілених додатків з багатьма користувачами.

4) Проблемою для ІВК з Мережею довіри є те, що існує бар'єр для додавання нових користувачів, так як: нові користувачі повинні мати довіру у вже зареєстрованих користувачів. Обидва типи ІВК мають недолік – не можна сховати ідентичність або відкритий ключ зареєстрованої особи [25].

Переваги ІВК на основі ТБЧ. ІВК на основі ТБЧ у порівнянні з традиційними PKI мають такі переваги:

1) Швидкість та простота перевірки сертифікату та ланцюга сертифікатів ЦСК. Сертифікати не потрібно підписувати, це означає, що вони коротші та потрібно менше часу на передачу сертифікату повернутого ланцюгом сертифікатів ЦСК.

2) Немає потреби у сервісі, який видає списки відкликаних сертифікатів (Certificate Revocation Lists, CRL), що є відомою з перших років застосування проблемою традиційних ІВК. CRL можуть бути дуже великими, та повинні зберігатися верифікатором та постійно оновлюватися по всій мережі. Таке спрощення здійснюється завдяки ТБЧ-синхронізації між вузлами мережі, де про будь-яку зміну стану сертифіката негайно повідомляється всім вузлам [62].

3) Немає потреби відповідати на запити протоколу онлайн статусу сертифікатів (online certificate status protocol, OCSP). Перевірки OCSP додають затримки в мережі для підтвердження сертифікатів та викривають інформацію про те, що суб'єкт подає верифікатору сертифікат. При цьому відбувається спостереження з криптографічними можливостями (правами) [64].

4) ІВК на БЧ можна використовувати для резервування простих сертифікатів БЧ так само, як і дорогих сертифікатів БЧ, і обидва випадки виграють від вищезазначених переваг.

5) Ще одним важливим аспектом в контексті безпеки в Інтернеті є те, що ІВК на основі БЧ забезпечує гнучкий захист від атак «людина по середині» (MITM).

Проблемні питання ІВК на базі БЧ. Технологія ТБЧ має ряд недоліків, які маються також внесені, у тому числі у ІВК на базі БЧ. Серед проблемних питань можна відзначити наступні:

1) Розмір відкритих БЧ постійно зростає, що розповсюджується на всі вузли, які беруть участь в системі. Особливо це актуально для платформ БЧ з підтримкою смарт-контрактів, які є важливими для ефективної організації ІВК.

2) Вартість ТБЧ-операцій напряму залежить від ціни відповідної криптовалюти.

3) Існує таке обмеження. Якщо для перевірки сертифікату використовується аналіз смарт-контракту з кодом Solidity, а не зовнішній модуль Golang, то може з'явитись обмеження на використання ґеш-функцій та асиметричних криптографічних функцій, доступних для смарт-контрактів. Наприклад, Ethereum без попередньої обробки даних може використовувати в якості ґеш-функції лише SHA256 та ЕП з використанням ECDSA на основі криптографії на еліптичних кривих.

4) Втрата паролю призводить до безповоротної втрати доступу до облікового запису. Рішенням проблем втрати паролю доступу до ЦС може стати створення порожніх смарт-контрактів та копіювання до них усіх даних до нового смарт-контракту.

Стійкість ІВК на основі БЧ до атак

1) Атака розділеного цифрового світу

У ІВК на основі публічних журналів «сильний» порушник, який має можливість контролювати довірені об'єкти типу ЦСК, та оператор журналу, може застосовувати атаки розділеного цифрового світу, надаючи різний вигляд журналів цільовим жертвам [24]. Хоча деякі з цих пропозицій не можуть виявити цю атаку, інші пропонують використовувати швидкий моніторинг для виявлення атак. А саме, протоколи домовленості для її виявлення, шляхом почергового перегляду журналу для клієнтів TLS та серверів [25, 26].

Тим не менш, ця атака може бути виявлена лише в тому випадку, якщо:

- є достатня кількість клієнтів і серверів TLS у домовленості взаємодії;
- принаймні деякі з них можуть переглядати справжній журнал і вимагати від журналу підтвердження узгодженості.

2) Суб'єкт (людина) посередині (MITM). Зазвичай MITM вважається головним ризиком для безпеки, коли зловмисник перешкоджає підключенню веб-браузера до певних веб-сайтів, надаючи дійсний сертифікат (тобто підроблений відкритий ключ) для цього домену. Для користувачів та веб-браузерів важко визначити заміну сертифіката у випадку, якщо зловмисник зламав відповідний ЦСК [25, 26].

Аналіз показав, що атака MITM є можливою коли порушник здатен переконати клієнта у незавершеній транзакції відкликаного TLS сертифікату. Більш конкретно, під час рукостискання TLS веб-сервери передають клієнту TLS транзакцію сертифіката для підтвердження сертифікату TLS. Клієнт TLS приймає цю транзакцію, якщо термін її дії не закінчився, і додає її до затвердженого блоку. Однак відкликаний або оновлений TLS сертифікат також може мати незавершену транзакцію у БЧ. Як тільки порушник надсилає цю незавершену транзакцію сертифіката зі своїми доказами Мерклі клієнту TLS, він приймається під час рукостискання TLS. Клієнти TLS не можуть виявити остаточний стан сертифікату, оскільки клієнти лише перевіряють наявність транзакції у відповідному блоці.

Підхід до побудови ІВК на основі БЧ робить атаки MITM практично неможливими, оскільки коли ЦСК публікує або робить відкликання відкритого ключа веб-сайту/домену в БЧ, інформація буде поширюватися на тисячі вузлів, тому підробка відкритого ключа (теоретично) не можлива [16, 17]. Традиційна ІВК вирішує ризики MITM, додаючи кореневі сертифікати ЦС до налаштування браузера, тим самим штучно розширюючи вхідні бар'єри ЦС та збільшуючи час, необхідний для відкликання кореневого сертифіката ЦСК [16, 17].

Висновки

1. При застосуванні принципу децентралізації стосовно існуючих ІВК необхідно поліпшити хоча б одну із важливих для цільового застосування таких характеристик як вартість, складність (часова та просторова), швидкість, прибутковість, безпека (загальна та інформаційна), анонімність, прозорість, гнучкість тощо.

2. Цільовий параметр, за яким здійснюється покращення ІВК має бути суттєвим, причому децентралізація повинна покращувати хоча б один важливий для користувачів параметр, як мінімум в 2-3 рази.

3. Нова технологія, в даному випадку технологія ІВК з використанням ТБЧ, не повинна істотно програвати існуючим ІТ за іншими важливими параметрами. Вважається, що у цілому нова ІВК повинна бути в чомусь краще в 2-3 рази, а за всіма іншими параметрами програвати не більше, ніж в 1,5 рази.

4. «Кращість» і перевага нової ІВК на основі ТБЧ мають суб'єктивний характер, вимоги до них повинні формуватись скоріше всього користувачами, і в меншій мірі розробниками.

5. У більшості випадків кращість може визначатись рівнем продаж таких ІТ. Відсутність певних продаж показує, що явної переваги цільової аудиторії відносно нової чи удосконаленої ІВК не визнано.

6. Одним із важливих та необхідних додатків ТБЧ є, по суті, удосконалення інфраструктури відкритого ключа (ІВК) на основі використання при її побудові принципів децентралізації та прозорості тощо.

7. Відомий ряд фатальних випадків, які приводили до того, що багато досліджень розподіляють абсолютну довіру до ЦС на декілька органів. Для виявлення підроблених, але дійсних сертифікатів TLS можна, застосовувати закріплення ключа, краудсорсингу та доведення до браузерів інформації про відкликання тощо. Вказані початкові рішення, які частково реалізовані, на жаль зазнали невдачі через проблеми масштабування.

8. Основним аргументом обґрунтування застосування БЧ при побудуванні ІВК є те, що рішення, засновані на ТБЧ, можуть об'єднати переваги ІВК, засновані на журналах, та підходи WoT, а також вирішити деякі проблеми зі звичайної системи ІВК. Так, БЧ усуває потенційні точки відмови підходу, заснованого на ІВК на основі журналу, і проблеми розгортання. З іншого боку, підхід, що заснований на БЧ, пом'якшує потреби WoT у нових власниках сертифікатів, щоб довести достовірність існуючих членів мережі, а також пом'якшує вимоги WoT для нових власників сертифікатів, щоб довести достовірність існуючих членів мережі.

9. Структура ІВК на основі БЧ підтримує відкликання сертифікату, що є суттєвою проблемою в традиційних системах ІВК. Також, оскільки неможливо видалити інформацію з БЧ, то тільки «батьківський» ЦСК може позначати виданий ним сертифікат як відкликаний. Тобто, будь-яка неправильна поведінка ЦСК стосовно відкликання сертифікату буде також простежена і помічена всіма іншими суб'єктами.

10. ІВК на основі БЧ перед традиційною ІВК має наступні переваги:

- перевірка сертифікату і його ланцюга сертифікатів ЦСК проста і швидка;
- ІВК на основі ТБЧ вирішує давню проблему традиційних ІВК, не вимагаючи використання сервісу, який видає списки відкликання сертифікатів (CRL).

11. Перевірка ланцюга довіри сертифіката на основі смарт-контракту виявляється більш значно ефективнішою. Ідея його полягає в тому, що спеціальний смарт-контракт читає і аналізує сертифікати ЦСК, що зберігаються в БЧ виключно за допомогою коду Solidity та компілятора для смарт-контракту Ethereum.

12. Виходячи з експериментів з тестовим загальнодоступним ТБЧ Rinkeby [26], запуск ЦСК, включаючи створення порожнього смарт-контракту, завантаження сертифікату в цей смарт-контракт, запис ґеш-значення цього сертифіката в смарт-контракт батьківського СА тощо, коштує 0,07 Ethers, що при нинішній вартості Ethers близько 700 USD за Ether переводиться у 50 USD за сертифікат ЦСК.

13. Система CertLedger управляє функціями ІВК через об'єкти стану. Об'єкт стану – це цифровий документ, який складається з даних і незмінного коду смарт-контракту для управління ним. Кожен об'єкт стану має унікальну адресу в БЧ. Державні зміни активів ініціюються операціями та відстежуються через державні об'єкти.

14. Стосовно сучасних ІВК з ЦСК існують наступні проблеми: єдина точка збою; необхідність застосування третьої довіреної сторони; висока вартість та неефективність управління ключами при великій кількості значно розподілених додатків з багатьма користувачами; бар'єр для додавання нових користувачів, так як нові користувачі повинні мати довіру у вже зареєстрованих користувачів.

15. ІВК на основі ТБЧ у порівнянні з традиційними PKI мають такі переваги: швидкість та простота перевірки сертифікату та ланцюга сертифікатів ЦСК; немає потреби у сервісі, який видає списки відкликаних сертифікатів; немає потреби відповідати на запити протоколу онлайн статусу сертифікатів (online certificate status protocol, OCSP); ІВК на БЧ можна використовувати для резервування простих сертифікатів БЧ так само, як і специфічних сертифікатів БЧ.

16. ІВК на основі БЧ забезпечує гнучкий захист від атак «людина посередині».

17. Серед проблемних питань можна відзначити такі: розмір відкритих БЧ постійно зростає; якщо для перевірки сертифікату використовується аналіз смарт-контракту з кодом Solidity, а не зовнішній модуль Golang, то може з'явитись обмеження на використання ґеш-функцій та асиметричних криптографічних функцій, доступних для смарт-контрактів; втрата паролю призводить до безповоротної втрати доступу до облікового запису.

18. Ця стаття носить характер первинного загальносистемного аналітичного огляду та відображає погляди авторів на можливості та необхідність створення ІВК з використанням ТБЧ. Ця наша впевненість ґрунтується на тому, що діюча ІВК України практично реалізована та підтримується при експлуатації [16 – 18].

Список літератури:

1. Andreas M. Antonopoulos Mastering Bitcoin: Unlocking Digital Cryptocurrencies /Andreas M. Antonopoulos Kyiv : NGITS, 2014. С. 10 – 150.

2. Що таке децентралізований додаток? [Електронний ресурс]. Режим доступу: https://www.coindesk.com/information/what-is-a-decentralized-application-dapp.

3. Don Tapscott, Alex Tapscott Blockchain Revolution: How the Technology Behind Bitcoin is Changing Money, Business, and the World / Don Tapscott, Alex Tapscott Blockchain. Kyiv : Information Systems, 2016. С. 65 – 102.

4. 20 основних застосувань БЧ [Електронний ресурс]. Режим доступу: https://biznesmodeli.ru/blockhain-chto-eto-cases-crypto-top10/.

5. БЧ: атаки, безопасность и криптография [Електронний ресурс]. Режим доступу: https://www.securitylab.ru/blog/personal/ Informacionnaya_bezopasnost_v_detalyah/343072.php.

6. Распределённые реестры и информационная безопасность: от чего защищает БЧ [Електронний ресурс]. Режим доступу: https://habr.com/company/bitfury/blog/341902/.

7. Pavan Duggal Blockchain Contracts and Cyberlaw / Pavan Duggal. Kyiv : Information Systems, 2015. С. 15–39.

8. Tim Harris. Bitcoin: Mastering Bitcoin & Cyptocurrency for Beginners – Bitcoin Basics, Bitcoin Stories, Dogecoin, Reinventing Money & Other Digital Currencies. Kyiv : Economic, 2016. С. 30–47.

9. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008. https://bitcoin.org/bitcoin.pdf.

10. NISTIR 8202 – Blockchain Technology Overview. 2018, 68 p. Access mode: https://doi.org/10.6028/NIST.IR.820210.

11. Возможные атаки на функции хэширования [Електронний ресурс]. Режим доступу: https://studfiles.net/preview/2157418/page:2/.

12. Прикладна криптологія. Теорія. Практика. Застосування : монографія / І.Д. Горбенко, Ю.І. Горбенко. Харків, 2012. С. 340-347.

13. Алгоритмы шифрования – основа работы криптовалют [Електронний ресурс]. Режим доступу: https://tgraph.io/Algoritmy-shifrovaniya – osnova-raboty-kriptovalyut-09-27.

14. Blockchain 3.0 – 5 лучших проектов нового поколения: https://privatfinance.com/blockchain-3-0-5-luchshih-proektov-novogo-pokoleniya.

15. Comparison of cryptographic hash functions [Електронний ресурс]. Режим доступу: https://en.wikipedia.org/wiki/Compaison_of_cryptographic_hash.

16. Горбенко Ю.І., Горбенко І.Д. Інфраструктури відкритих ключів. Системи ЕЦП. Теорія та практика. Харків : Форт. 2010. 593 с.

17. Горбенко І.Д., Горбенко Ю.І. Прикладна криптологія. Харків : ХНУРЕ ; Форт, 2012. 1 та 2-е вид. 868 с.

18. Горбенко Ю.І. Побудування та аналіз систем, протоколів та засобів криптографічного захисту інформації ; за ред. І.Д. Горбенко. Харків : Форт. 2015. 902с.

19. Закон України «Про електронні довірчі послуги» // Відомості Верховної Ради (ВВР). 2017. № 45. ст.400.

20. Закон України «Про захист інформації в інформаційно-телекомунікаційних системах» від 31.05.2005 № 2594-IV.

21. Закон України «Про основні засади забезпечення кібербезпеки України» // Відомості Верховної Ради (ВВР). 2017. № 45, ст.403.

22. Регламент (ЄС) № 910/2014 Європейського Парламенту та Ради від 23 липня 2014 р. «Про електронну ідентифікацію та довірчі послуги для електронних транзакцій у межах внутрішнього ринку та про скасування Директиви 1999/93/ЄС» (1) (COM (2012) 0238-C7-0133/2012 – 2012/0146 (COD)).

23. ДСТУ 4145-2002 «Інформаційні технології. Криптографічний захист інформації. Цифровий підпис, що ґрунтується на еліптичних кривих. Формування та перевіряння».

24. ДСТУ ITU-TRec.X.509|ISO/IEC 9594-8:2015 «Інформаційні технології. Взаємозв'язок відкритих систем. Каталог: Основні положення щодо сертифікації відкритих ключів та сертифікації атрибутів».

25. CertLedger: A New IBK Model with Certificate Transparency Based on Блокчейн. Murat Yasin Kubilay, Mehmet Sabir Kiraz and Haci Ali Mantar. Access mode: https://eprint.iacr.org/2018/1071.pdf.

26. Yakubov Alexander A Blockchain-Based PKI Management Framework / Alexander Yakubov, Wazen M. Shbair, Anders Wallbom, David Sanda, Radu State // Access mode: https://orbilu.uni.lu/bitstream/10993/35468/1/blockchain-based-pki.pdf.

27. L. Axon and M. Goldsmith. PB-PKI: A privacy-aware blockchain-based PKI // Proceedings of the 14th International Joint Conference on e-Business and Telecommunications (ICETE 2017). Volume 4: SECRYPT, Madrid, Spain, July 24-26, 2017, 2017, pp. 311–318. [Electronic resource]. Access mode: https://doi.org/10.5220/0006419203110318.

28. C. Fromknecht, D. Velicanu, and S. Yakoubov. Certcoin: A namecoin based decentralized authentication system // Massachusetts Inst. Technol., Cambridge, MA, USA, Tech. Rep, vol. 6, 2014.

29. Mozilla included CA certificate list, 2017. [Electronic resource]. Access mode: https://www.mozilla.org/en-US/about/governance/policies/security-group/certs/.

30. D. Cooper. Internet x. 509 public key infrastructure certificate and certificate revocation list (crl) profile. 2008.
31. E. Androulaki, C. Cachin, A. D. Caro, A. Sorniotti, and M. Vukolic. Permissioned blockchains and hyperledger fabric // ERCIM News, vol. 2017, no. 110, 2017. [Electronic resource]. Access mode: https://ercim-news.ercim.eu/en110/special/permissioned-blockchains-and-hyperledger-fabric.
32. A. J. Nicholas Stifter and E. Weippl. A holistic approach to smart contract security // ERCIM News, vol. 2017, no. 110, 2017. [Electronic resource]. Access mode: https://ercim-news.ercim.eu/en110/special/a-holistic-approach-to-smart-contract-security/.
33. M. Ali, J. C. Nelson, R. Shea and M. J. Freedman. Blockstack: A global naming and storage system secured by blockchains // USENIX Annual Technical Conference, 2016. pp. 181–194.
34. K. Lewison and F. Corella. Backing rich credentials with a blockchain pki, 2016.
35. M. Alicherry and A. D. Keromytis. Doublecheck: Multi-path verification against man-in-the-middle attacks // Computers and communications, 2009. iscc 2009. ieee symposium on. IEEE, 2009, pp. 557–563.
36. Blockchain & cyber security. let's discuss, 2017. [Electronic resource]. Access mode: https://www2.deloitte.com/content/dam/Deloitte/ie/Documents/Technology/IECBlockchainandCyberPOV 0417.pdf.
37. Jiangshan Yu, Vincent Cheval, and Mark Ryan. DTKI: A new formalized PKI with verifiable trusted parties // The Computer Journal, 59(11):1695-1713, 2016.
38. B. Laurie, A. Langley, and E. Kasper. Certificate transparency. RFC 6962 (experimental), 2013.
39. Scott A Crosby and Dan S Wallach. Efficient data structures for tamper-evident logging // USENIX Security Symposium, pages 317-334, 2009.
40. RFC 5280: Internet X.509 public key infrastructure: certificate and CRL profile.

Харківський національний
університет імені В.Н. Каразіна;
АТ «Інститут інформаційних технологій»;
Департамент Державної служби спеціального зв'язку
та захисту інформації України

Надійшла до редколегії 15.09.2019

УДК 004.728:004.728.3, 004.056.055 **DOI:10.30837/rt.2019.3.198.13**

О.А. ЗАМУЛА, д-р техн. наук

ОПТИМІЗАЦІЯ МЕТОДІВ СИНТЕЗУ ДИСКРЕТНИХ СКЛАДНИХ СИГНАЛІВ У СУЧАСНИХ БАГАТОКОРИСТУВАЧЕВИХ СИСТЕМАХ ЗВ'ЯЗКУ ШИРОКОСМУГОВОГО ДОСТУПУ

Вступ

У багатокористувачевих системах зв'язку з кодовим поділом необхідні сімейства широкосмугових дискретних складних сигналів з особливими ансамблевими, структурними, технологічними, кореляційними властивостями. Застосування складних широкосмугових сигналів дозволяє підвищити захищеність систем зв'язку при впливі навмисних перешкод у вигляді: загороджувальних перешкод (перешкода у вигляді стаціонарного гаусова шуму з нульовим середнім і рівномірним розподілом спектральної щільності потужності, принаймні, в області частот, зайнятої сигналом), і яка створюється навмисно як засіб радіоелектронної протидії; вузькополосної перешкоди; потужних структурних перешкод з нерівномірним спектром і деяких інших типів перешкод. При радіоелектронній протидії ефективна перешкода може бути організована тільки після виявлення присутності системи, що протистоїть в ефірі і оцінки таких її параметрів як частотний діапазон, яку смугу займає сигнал, форми використовуваних сигналів. Для запобігання можливості виявлення сигналу станцією протидії інфокомунікаційна система (ІКС) повинна використовувати сигнали з розподіленим або широким спектром, які мають максимально можливе значення виграшу від обробки (твір смуги частот, займаної сигналом на його тривалість), і структуру, що практично не розкривається.

Основні результати досліджень

Проектування і створення сучасних систем зв'язку передбачає використання ансамблів сигналів, які володіють однією з властивостей [1]:

- кожен з сигналів даного ансамблю легко може бути розрізнено від своєї копії, що зсунена у часі;
- кожен з сигналів даного ансамблю легко може бути розрізнено від іншого сигналу цього ансамблю.

Перша властивість є важливою для радіолокаційних, сонарних систем, систем синхронізації, а також для широкосмугових систем зв'язку, друга – для широкосмугових систем зв'язку з багатостанційним доступом і кодовим ущільненням каналів.

Типовим для теорії зв'язку є підхід, що полягає в розробці оптимального приймального пристрою, який з найкращою якістю відновить інформацію, що міститься в коливанні, що спостерігається. Визначення оптимального алгоритму обробки, що базується на обліку специфічних властивостей переданого сигналу, дозволяє синтезувати оптимальним чином і сам сигнал, тобто вибрати найкращий метод його кодування і модуляції.

У теорії зв'язку найбільш поширеною моделлю служить канал з адитивним білим гаусовським шумом, в якому ймовірність трансформації каналом заданого вхідного сигналу в те чи інше вихідне спостереження $y(t)$ (перехідна ймовірність – $P\left[y(t)|S(t)\right]$) експоненціально зменшується зі зростанням квадрата Евклідової відстані між переданим сигналом і вихідним коливанням [2]:

$$P\left[y(t)|S(t)\right]=\kappa\exp\left(-\frac{1}{\mathrm{N}_0}\,d(s,y)\right), \qquad (1)$$

де κ – константа, що не залежить від $S(t)$ і $y(t)$, N_0 – спектральна щільність потужності одностороннього білого шуму; а Евклідова відстань між $S(t)$ і $y(t)$ визначається як

$$d(S,y)=\sqrt{\int_0^T[y(t)-S(t)]^2\,dt}\,. \qquad (2)$$

Згідно з співвідношеннями (1) і (2) схожість сигналу (ймовірність того, що він перетворений каналом в спостереження $y(t)$ зменшується зі збільшенням Евклідової відстані між $S(t)$ і $y(t)$. У разі рівної ймовірності всіх повідомлень джерела (що досягається при правильному проектуванні системи) оптимальною стратегією спостерігача, що забезпечує мінімальну помилку визначення дійсно переданого з деяким іншим сигналом, є правило (критерій) максимальної правдоподібності (МП). Згідно з цим алгоритмом, після того, як коливання $y(t)$ стало прийнято, рішення приймається на користь того сигналу, для якого ймовірність трансформації його каналом в прийняте спостереження є найбільшим (в порівнянні з можливостями для інших сигналів). З урахуванням викладеного, МП рішення для гаусова каналу може бути перетворено в правило мінімуму відстані:

$$d(S_j,y)=\min d(S_i,y)\Rightarrow H\ j\,, \qquad (3)$$

тобто рішення приймається на користь сигналу $S_j(t)$, оскільки він найбільш близький (в сенсі Евклідової відстані) до спостереження $y(t)$ серед всіх конкуруючих сигналів.

При виборі класу дискретних сигналів орієнтуються, як правило, на критерій мінімуму взаємних перешкод (мінімаксний критерій). Такий критерій має на меті побудову ансамблів сигналів обсягу M, маніпульованих дискретними послідовностями (ДП), які як можна помітніше відрізняються один від одного при можливих циклічних зрушеннях. Кількісною мірою відмінності маніпулюючих ДП служать максимальні за ансамблями рівні бічних пелюсток періодичної функції автокореляції (ПФАК) і рівні бічних пелюсток періодичної функції взаємної кореляції (ПФВК), що визначаються відповідно як [2]:

$$p_p(m)=\frac{1}{\left\|a^2\right\|}\sum_{i=0}^{n-1}a_i\cdot a^*_{i-m}\,,\ \rho_p,k_l(m)=\frac{1}{\left\|a_k\right\|\left\|a_l\right\|}\cdot\sum_{i=0}^{N-1}a_{k,i}\cdot a^*_{l,i-m}\,, \qquad (4)$$

де a_k (a_l) – комплексна амплітуда k(l)-ї дискретної послідовності.

Виходячи з цього, широкосмугові сигнали (ШПС), що застосовуються в системах зв'язку, мають володіти такими кореляційними властивостями, коли бічні піки кореляційних функцій ШПС є якомога меншими, тобто в ідеальному випадку повинні прагнути до нуля. При цьому, процес вибору раціональних по тих чи інших критеріях структур складних сигналів тотожний синтезу відповідних маніпулюючих ДП. Однак вимога ідеальності (нульові значення бічних піків) авто- і взаємно-кореляційних функцій між всіма циклічними зрушеннями K послідовностей і різними ізоморфізмами системи сигналів з періодом N не здійснена, оскільки значення бічних піків не можуть опуститися нижче $1/2\sqrt{B}$ (де B – база сигналу) [3]. Зазначене пояснюється наступним. Оптимальний прийом сигналів здійснюється за допомогою узгодженого фільтра (УФ) або корелятора. Нормований відгук УФ визначається за допомогою інтеграла згортки [3]

$$R_{ij}=\ 1/E\int U_j(t)\ U_k(t-\tau)dt\,, \qquad (5)$$

де $U_j(t)$ – сигнал на вході фільтра, узгодженого з сигналом.

Залежно від того, узгоджений або не узгоджений сигнал з фільтром, чи є додаткове допплерівське зміщення частоти сигналу, кореляційна функція має різні представлення. Одним з таких представлень є взаємна функція невизначеності (ВФН) сигналів з номерами j і k. ВФН може бути представлена через комплексні обхідні сигналів і через їх спектри наступним чином:

$$R_{jk}(\tau,\Omega)\ =\ 1/2E\int_{-\infty}^{\infty}U_j(t)\ U^*_k(t-\tau)\ e^{i\Omega t}dt\ =\ 1/4E\int_{-\infty}^{\infty}G_j(\omega-\Omega)\ G^*_k(\omega)\ e^{i\omega t}d\omega\,, \qquad (6)$$

де τ – зсув за часом між сигналами; Ω – допплерівській зсув частоти.

Відгук узгодженого фільтра пов'язаний з ВФН співвідношенням

$$r_{jk}(\tau,\Omega) = \mathrm{Re}\, R_{jk}(\tau,\Omega)\exp(i\omega_0\tau). \quad (7)$$

Об'єм ВФН R_{jk} (τ, Ω) сигналів j і k (обсяг, укладений між поверхнею, яка описується квадратом модуля ВФН і площиною невизначеності), дорівнює одиниці, тобто

$$1/2\pi\int\int_{-\infty}^{\infty} \left| R_{jk}(\tau,\Omega)\right|^2 d\tau\, d\Omega = 1 \quad . \quad (8)$$

і не залежить від номерів і форми сигналів. Іншими словами, отримати тіло невизначеності з нульовими бічними пелюстками неможливо. Тіло невизначеності за умови, що всі бічні піки рівні і рівномірно розподілені в квадраті (2T, 2F) (в цьому випадку бічні піки мінімальні за амплітудою), зображено на рис. 1. Вузький основний пік розташовується на підставі, висота якої $R_0 = ½\sqrt{B}$. Розгляд властивостей ФН дозволяє визначити основні правила побудови сигналів, які можуть бути використані для передачі в сучасних високошвидкісних системах стільникового зв'язку, бездротових дискретних комунікаційних системах, при передачі інформації цифрового телебачення і радіо, в системах радіолокації тощо. Інтегральні відношення, що витікають з властивостей інваріантності об'єму, свідчать, що сигнал, який має єдиний пік у началі координат плоскості невизначеності знайти неможливо. Так звана «кнопочна» функція невизначеності (рис. 1) є найбільш близькою апроксимацією поверхні невизначеності з єдиним піком. Наявність в N – вимірному лінійному просторі не більше N ортогональних векторів (сигналів) робить гіпотетичним ідеальний, з точки зору мінімаксного критерію, ансамбль дискретних послідовностей з нульовими бічними пелюстками функції авто – і взаємної кореляції, і обмежує потенціал зниження кореляційного викиду R при фіксованих N і числі абонентів K.

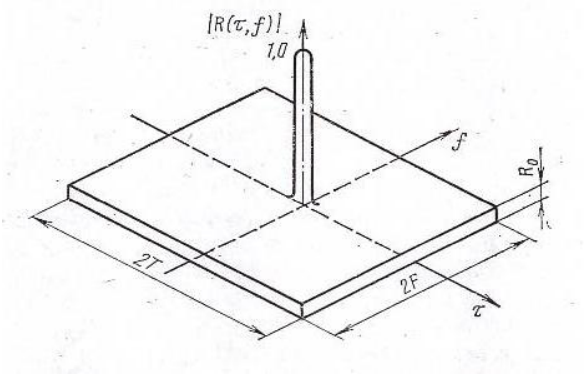


Рис. 1. Тіло невизначеності

У зв'язку з наведеним, при проектуванні і створенні ІКС важливим завданням є вибір сигналів, що забезпечують мінімально можливий рівень взаємних перешкод, який в основному визначається допустимим рівнем максимальних піків взаємно кореляційних функцій (ВКФ). Для режиму виявлення важливо мати систему, складену з сигналів, що володіють малими піками періодичних і аперіодичних автокореляційних функцій (ПФАК і АФАК).

В теорії складних сигналів відомий ряд інтегральних рівності [1]. Нехай C – множина комплексних чисел, а C^N множина векторів з комплексними компонентами. Елементи множини $w, x, y, z \in C^N$ – довільні вектори, а w, x, y, z – відповідні їм дискретні послідовності. Чотири взаємно-кореляційні функції $R_{w,x}$, $R_{y,z}$, $R_{w,y}$, $R_{x,z}$ пов'язані співвідношенням

$$\sum_{l=0}^{N-1} R_{w,y}(l)[R_{x,z}(l+n)]^* = \sum_{l=0}^{N-1} R_{w,x}(l)[R_{y,z}(l+n)]^* . \quad (9)$$

Поклавши в (1) $z = y$, отримаємо

$$\sum_{l=0}^{N-1} R_{w,y}(l)[R_{x,y}(l+n)]^* = \sum_{l=0}^{N-1} R_{w,x}(l)[R_y(l+n)]^* . \quad (10)$$

Поклавши в (2) $w = x$, отримаємо

$$\sum_{l=0}^{N-1} R_{x,y}(l)[R_{x,y}(l+n)]^* = \sum_{l=0}^{N-1} R_x(l)[R_y(l+n)]^* . \quad (11)$$

Нарешті, поклавши в (5) $n = 0$, отримаємо

$$\sum_{l=0}^{N-1} \left|R_{x,y}(l)\right|^2 = \sum_{l=0}^{N-1} R_x(l)[R_y(l)]^* . \quad (12)$$

За допомогою (9) – (12) отримано ряд важливих границь оцінки кореляційних функцій. Рівність (11) означає, що автокореляційна функція (АКФ) послідовності $R_{x,y}$ збігається з взаємно-кореляційною функцією(ВКФ) послідовностей R_x і R_y. Крім того, з (12) видно, що середнє значення квадрата модуля функції взаємної кореляції сигналів x і y дорівнює середньому значенню добутку їх АКФ. Фактично це означає, що сигнали, що володіють хорошими автокореляційними властивостями, будуть володіти і хорошими властивостями ВКФ. Це фундаментальне положення теорії систем сигналів було покладено в основу синтезу ансамблів сигналів з відповідними покращеними кореляційними властивостями.

На сьогодні немає єдиної теорії синтезу (з «щільно упакованою» по ПФАК) ДП для довільних довжин послідовностей. У той же час, для вирішення завдань як циклової синхронізації, так і забезпечення необхідної завадостійкості, скритності функціонування системи зв'язку, необхідно використовувати дискретні сигнали з довільними значеннями тривалості послідовностей і мінімальними значеннями бічних пелюсток ПФАК. Процес вибору раціональних по тих чи інших критеріях структур складних сигналів тотожній синтезу відповідних маніпулюючих ДП.

В [4] вказані принципово досяжні значення максимальних бічних піків періодичної функції автокореляції (межі щільної упаковки) для заданого періоду послідовності N. Наведені границі встановлюють критерії синтезу множини ДП (сигнатур). Ансамблі зі значеннями, що передбачають відповідність цим границям, є оптимальними і називаються мінімаксними:

$$\mathrm{R}^{\mathrm{a}}_{\max} \geq \begin{cases} 0, \text{ якщо } \mathrm{N} \equiv 0 (\mathrm{mod}\ 4); \\ 1, \text{ якщо } \mathrm{N} \equiv 1 (\mathrm{mod}\ 4); \\ 2, \text{ якщо } \mathrm{N} \equiv 2 (\mathrm{mod}\ 4); \\ -1, \text{ якщо } \mathrm{N} \equiv 3 (\mathrm{mod}\ 4), \end{cases} \quad (13)$$

Для ідеального гіпотетичного ансамблю $\mathrm{R}_{\max}$ дорівнює нулю, а для будь-якого реального ансамблю мінімальне значення кореляційної функції може служити адекватною мірою його близькості до ідеального.

Ансамблі дискретних послідовностей, що застосовуються у широкосмугових системах зв’язку з прямим розширенням спектру можуть виправдовувати своє призначення тільки в ситуаціях, коли взаємні тимчасові зрушення користувачевих сигналів повністю контролюються системою і можуть бути утримані в рамках передбаченого діапазону. Якщо ж це не виконується, то асинхронний тип широкосмугової системи з множинним доступом і кодовим поділом каналів, заснований на використанні зрушених копій однієї і тієї ж послідовності, схильний до виникнення колізій: сигнал одного з користувачів може придбати затримку, що не дозволяє відрізнити його від сигналу деякого іншого користувача. Останнє може бути підставою для використання ансамблів мінімаксних сигналів. Оскільки кореляційний пік мінімаксного ансамблю отримано в результаті максимізації на всьому періоді, то його мале значення (досягнуте за рахунок досить великої довжини N) забезпечує близькість кореляційних властивостей ансамблю до ідеальних характеристик. Інтерес до послідовностей з хоро-

шою періодичною АКФ не обмежується тільки їх роллю вихідного матеріалу для побудови хороших аперіодичних послідовностей. Існує безліч програм, заснованих на використанні періодичних дискретних сигналів (CW – локація, навігація, пілотний канал і канал синхронізації в мобільних системах радіозв'язку, радарні і сонарні системи з безперервним випромінюванням і ін.), що зумовлює важливість періодичної АКФ щодо системних характеристик. Прийнято вважати «ідеальною» таку ПФАК, яка володіє нульовими бічними пелюстками, тобто нульовими значеннями між періодичними основними пелюстками, що повторюються з періодом N. При цьому можлива ситуація, коли прийнятне значення вимагає досить великої довжини N. Наприклад, для локаційних дальномірних і сонарних систем вимоги часового розрішення сигналів в динамічному діапазоні, що перевищує 80дБ, є досить звичайним. Для виконання цієї умови потрібні оптимальні бінарні послідовності довжини, що перевищують 10^4, що може уповільнити початкову процедуру пошуку сигналу [2]. Очевидно, що для подібних випадків може служити ідеальна ПФАК (13), яка є недосяжною на безлічі бінарних кодів.

Розглянемо можливі шляхи досягнення ідеальної ПАКФ для випадків, коли алфавіт не обмежений вимогою бінарності сигналів {± 1}.

Бінарні послідовності з непротилежною модуляцією

Сутність побудови таких послідовностей полягає у заміні алфавіту {+1,-1} на деякий інший алфавіт є додаванням константи *c* до вихідного алфавіту {+1,-1} послідовності $a_o, a_1, ...a_{n-1}$, а саме: символи +1 і - 1 змінюються на + 1 + *c* і - 1 + *c* відповідно. В [1] наведено правило перетворення бінарної мінімаксної послідовності з ПФАК виду (14) в нову з ідеальною АКФ: елементи, що відповідають - 1, замінюють на $-1 \pm \frac{2}{\sqrt{N+1}}$, а елементи +1 залишаються без зміни. Даний метод визначає використання значень комплексних амплітуд тоді, коли установка і підтримування їх може виявитися утрудненою на практиці.

Багатофазні коди

Одним з правил конструювання недвійкової фазової модуляції з основою M> 2 з ідеальною ПФАК є алгоритм, відповідний кодам квадратичних лишків [2]. Зазначені коди існують при довільному значенні довжини N і формуються як [3]:

$$a_i = \begin{Bmatrix} \exp\left(\frac{j\pi i^2}{N}\right), N - четное \\ \exp\left(\frac{j2\pi i^2}{N}\right), N - нечетное \end{Bmatrix}, \quad (14)$$

де *i* = ……. -1,0,1 ……..

Практична реалізація кодів квадратичних лишків, незважаючи на те, що ці коди є переконливим прикладом ФМ послідовності з ідеальною АКФ, проблематична. Зазначене обумовлено наступним: розмір фазового алфавіту лінійно зростає зі збільшенням довжини і відстань між сусідніми фазами стає надзвичайно малою. Цим, в свою чергу, обумовлена зростаюча вимогливість до точності формування символів коду, якості відтворення фаз, умовам експлуатації та ін. До множини багатофазних кодів відносять коди Франка. Вони здійснюють так само як і коди квадратичних лишків покрокову апроксимацію лінійної частотної модуляції і існують при значеннях довжин, що представляють квадрат цілого числа $N = h^2 = 4,9,16,25,36,49$ .... Правило їх формування описується співвідношенням

$$a_i = \exp\left(\frac{j\pi i}{h}\left[\frac{i}{h}\right]\right), i = ...-1,0,1...., \quad (15)$$

де [x] позначає округлення x в меншу сторону.

З (14) і (15) випливає, що збільшення обсягу алфавіту з ростом N відбувається значно повільніше. Аналіз багатофазних кодів показує, що технологічно дані коди не настільки привабливі в порівнянні з бінарними протилежними кодами.

Троїчні послідовності

На відміну від бінарних послідовностей елементи троїчних послідовностей a^2 на додаток до значень ± 1 приймають ще й нульове значення, тобто використовується трійчастий алфавіт {-1,0,1}. Такий алфавіт означає комбінування бінарної ФМ з паузами, тобто інтервалами часу, протягом яких відсутня передача символів. В [2] розглянуто способи конструювання троїчних послідовностей. Суть одного із способів полягає в наступному. Нехай $d_i, i = ...-1,0,1,....p$ - на *m*-послідовність, де *p* – просте непарне число. Кожен символ послідовності є елементом простого поля GF(P). Послідовність перетворюється в троїчну шляхом відображення нульового елемента в уявний нуль, а ненульові елементи – в їх двозначні характери. Після подібного перетворення змінюють знаки всіх елементів, що стоять на непарних позиціях. Формально, алгоритм може бути представлений таким співвідношенням

$$a_i = \begin{Bmatrix} (-1)^i \Psi(d_i) d_i \neq 0 \\ 0, d_i = 0 \end{Bmatrix}, \quad (16)$$

де *i* = ….., -1,0,1……

Однією з основних причин появи інтересу до розширення спектра в задачах часового виміру служить прагнення досягнути високих показників при низькій пікової потужності, тобто при розподілі енергії сигналу на великому часовому інтервалі. Як показник ефективності розподілу енергії в часі використовують величину пік-фактору V, тобто відношення пікової та середньої потужності. Для будь-якої ФМ, і зокрема бінарної, енергія послідовності сигналу рівномірно розподілена на періоді так, що пікова і середня потужності однакові і, отже, V = 1. Зведення N_p пауз на періоді троїчної послідовності порушує рівномірність розподілу енергії і збільшує пік-фактор в $\frac{N}{N-Np}$ раз. Таким чином, цільовою функцією синтезу є побудова троїчних послідовностей не тільки з ідеальною ПФАК, але і малим числом нулів N_p на періоді, тобто пік-фактором, який незначно перевищує одиницю. Троїчна послідовність може бути утворена за допомогою посимвольного множення отриманої відповідно до правила (17) послідовності на єдину бінарну послідовність 1,1,1, – 1, яка має ідеальну ПФАК. Результуюча троїчна послідовність буде характеризуватися учетверо більшою довжиною без зміни значення пік-фактору і ідеальності АКФ. Крім того, послідовності, що утворюються шляхом посимвольного добутку двох троїчних послідовностей з ідеальною ПФАК і взаємно простими довжинами N_1, N_2, так само будуть мати ідеальну АКФ, довжину $N_1 \cdot N_2$ і пік-фактор $V = V_1 \cdot V_2$. В [1] показано, що такі послідовності володіють ідеальною періодичною АКФ:

$$P_p(m) = \begin{Bmatrix} P^n - 1, m \neq 0 \bmod N \\ 0, m = 0 \bmod N \end{Bmatrix}, \quad (17)$$

де $N = \frac{P^n - 1}{P - 1}$ – істинний період троїчної послідовності.

Характеристичні коди

До числа привабливих, з точки зору кореляційних властивостей, відносяться характеристичні дискретні сигнали (характеристичні коди, далі – ХДС) з числом позицій $N = 4x + 2$ і, $N = 4x, x = 1,2$ [4]. Максимальні бічні викиди ПФАК таких сигналів складають {- 4, 0}, тобто даний клас сигналів відноситься (у відповідності до (14)) до оптимальних або мінімаксних

сигналів. Обсяг системи ХДС визначається зі співвідношення [4]: $M = \phi(N)/n$, (φ (N) – функція Ейлера, *n* – ступінь розширення поля $GF(P^n), n \geq 1$.

Криптографічні сигнали

Під час досліджень [5, 6] вперше отримано метод синтезу складних нелінійних криптографічних сигналів (КС), що дозволяє створювати: великі ансамблі дискретних послідовностей практично будь-якого періоду з необхідними (для відповідних задач, що стоять перед ІКС) значеннями бічних пелюсток авто, взаємної і стикової функцій кореляції в періодичному і аперіодичному режимах роботи; послідовності з статистичними характеристиками кореляційних функцій (КФ), аналогічними характеристиками кращих, з погляду кореляційних функцій, лінійних класів сигналів; послідовності, які відповідають вимогам незворотності, нерозрізненості, непередбачуваності і володіють необхідними структурними та ансамблевими властивостями. КС засновані на нелінійних правилах побудови, оскільки при їх створенні застосовують випадкові (псевдовипадкові) процеси, зокрема, алгоритми криптографічного перетворення даних. Зазначене дозволяє покращити показники завадозахищеності, імітостійкості, структурної скритності ІКС, а також завадостійкості прийому сигналів в умовах впливу структурних, загороджувальних, ретрансльованих та інших видів завад і мають покращені кореляційні, ансамблеві і структурні властивості. Зокрема, при використанні таких сигналів як фізичного переносника інформації або сигналів синхронізації часові витрати на розкриття структури використовуваних сигналів зростають і постановка «оптимальних», з точки зору станції протидії, перешкод стає проблематичною.

Будь-який циклічний зсув послідовності довжини N володіє такою ж періодичною ПФАК, що і вихідна послідовність, оскільки періодична ПФАК – інваріантна до циклічного зсуву. Аперіодична ФАК (АФАК) циклічно зрушеної копії може відрізнятися від АКФ первісної. Даний факт разом з границею (13) становить основу методу (алгоритму) пошуку послідовностей з прийнятною АКФ. Очевидно, що знаходження оптимальних бінарних послідовностей великої довжини практично не піддається реалізації. Така задача може бути сформульована у вигляді: найти бінарній код з задовільно малим рівнем бічної пелюстки. Загальна ідея алгоритмів, спрямованих на вирішення цієї задачі, полягає в попередньому відборі деякої обмеженої множини послідовностей, яка здається багатообіцяючою в плані кореляційних властивостей, и подалі – в пошуку коду з мінімальним значенням тільки серед послідовностей, які увійшли у зазначену множину. На першому етапі для заданої довжини N деяким чином формується безліч послідовностей з хорошою ПФАК. Вона може включати всі відомі послідовності заданої довжини N, рівень бічних пелюсток ПФАК яких згідно (14) дозволяє сподіватися на отримання низького значення R_{max}. Наприклад, якщо необхідні бінарні коди довжиною N = 63, то початкова множина може бути обмежена *m*-послідовностями, криптографічними сигналами, або включати інші послідовності з задовільною ПФАК.

На другому етапі здійснюється пошук за критерієм найменшого рівня максимуму АФАК серед усіх одноперіодних сегментів послідовностей кандидатів. Зокрема, береться одноперіодний сегмент першої послідовності кандидата, обчислюється його АФАК і запам'ятовується в пам'яті рівень максимального бічного пелюстка поряд з номерами послідовності кандидата і його зсуву. Потім здійснюється циклічний зсув сегмента на одну позицію і проводяться необхідні обчислення. Якщо нове значення максимуму аперіодичної бокової пелюстки виявиться нижче попереднього, то його значення і номер нового зсуву замінюють раніше записані в пам'яті дані, в іншому випадку зареєстровані значення зберігаються без зміни. Дана процедура повторюється N раз, тобто для всіх циклічних зрушень першої послідовності кандидата. Результатом пошуку є послідовність з мінімальним значенням R_{max} серед послідовностей, відібраних на першому етапі. Ансамбль можливих сигналів може бути складений, поряд з іншими кодами, і з характеристичних кодів. Для кожного з кодів слід шляхом циклічного зсуву його символів знайти оптимальні за мінімаксним критерієм аперіодичні коди і відібрати з них найкращі.

Дослідження автокореляційних властивостей ХДС в аперіодичному режимі передач показали [7], що для періоду ДП 256 символів існує 56 характеристичних ДП, для яких значення максимальних бічних піків АФАК не перевищує значення 18 (1,1√N). Такі значення бічних піків є меншими ніж бічні піки одних з кращих, з точки зору кореляційних властивостей, *m*-послідовностей. Форму АФАК для однієї з таких ДП показано на рис. 2. Дані про деякі з характеристичних сигналів (коефіцієнт децимації, який використаний для отримання відповідного ізоморфізму ДП, і циклічний зсув цих ізоморфізмів) наведено в табл. 1. Було синтезовано 470 ХДС, нормовані значення максимальних бічних піків АФАК яких не перевищують величини 20/256. У стандарті системи з кодовим поділом UMTS в якості коду первинної синхронізації використовується бінарна синхропослідовність (СП) з періодом 256 двійкових елементів, які володіють аперіодичними бічними пелюстками аж до ¼, тобто R_{max} = 64 (або -12 дБ) [2]. При виборі ХДС як СП, у порівнянні з сигналами, що застосовуються в стандарті UMTS, виграш, з точки зору завадостійкості прийому сигналів, становитиме більше 4 дБ [2]. Необхідно зауважити, що для періоду ХДС з періодом 256 символів існує 64 ізоморфних сигналів, що суттєво перевищує обсяг системи лінійних сигналів (М-послідовностей) для даного періоду.

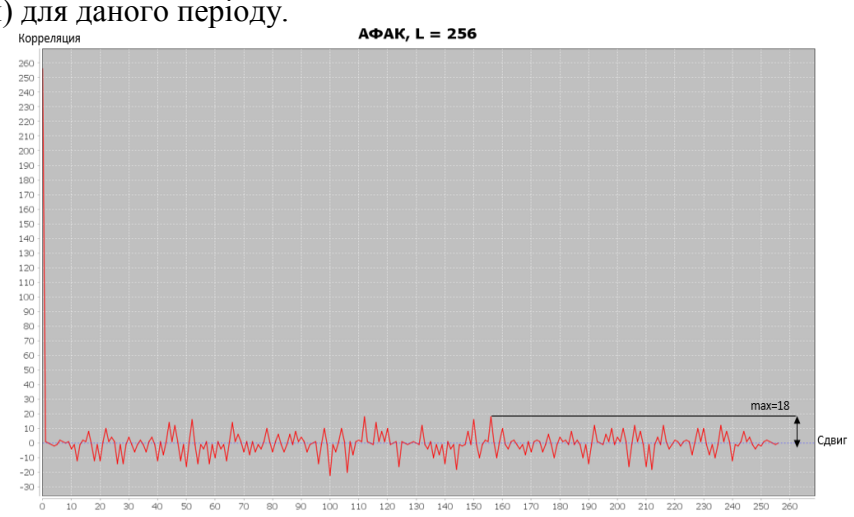


Рис. 2. АФАК ХДС (циклічний зсув {112}, коефіцієнт децимації – 7)

Зазначене має істотне значення для систем, однією з вимог до яких, є забезпечення захищеності від нав'язування (введення) неправдивих повідомлень, помилкових режимів роботи і інше. Для перевірки гіпотези щодо можливості застосування (з метою покращення показників інформаційної безпеки, завадостійкості прийому сигналів, скритності функціонування ІКС) криптографічних сигналів було синтезовано 680 сигналів даного класу, величина R_{max} АФАК для яких, не перевищує значень 33 (це найкраще граничне значення для максимальних бічних піків двійкових сигналів з періодом 256 елементів).

Таблиця 1

Період (N)	Коефіцієнт децимації	Максимальні бокові піки АФАК	Відповідні зсуви ізоморфізмів ХДС
256	7	18	{112,156 }
256	13	18	{44,66 }
256	37	18	{58}
256	47	18	{114}
256	61	18	{84,114,146,160}
256	101	18	{92,94}
256	127	18	{132,180}

В цьому випадку, виграш з точки зору ймовірності правильного прийому, в порівнянні з використанням послідовностей, що застосовуються в стандарті мобільного зв'язку UMTS, складає 3 дБ. Якщо висуваються більш жорсткі умови до завадозахищеності прийому сигналів в ІКС, можна запропонувати застосовування КС, для яких значення R_{max} АФАК менше ніж 33 [8, 9]. В табл. 2 наведено дані щодо деяких КС, R_{max} для яких не перевищують значення 26, а на рис. 3 наведено вид АФАК для одного з таких КС.

Таблиця 2

Номер сигнала	Значення максимальних бокових піків АФАК	Відповідні зсуви КС
1	25	{31}
2	25	{61}
3	26	{60}
4	26	{10,22}
5	24	{212}
6	26	{48}
7	21	{3,83}
8	26	{66}
9	26	{62}
10	24	{6}
11	23	{57}
12	26	{8,90}
13	25	{111}
14	25	{39,77,123}
15	24	{26,90}
16	23	{5,23,37,39}
17	26	{16,86}
18	24	{80}
19	24	{70}
20	25	{19}

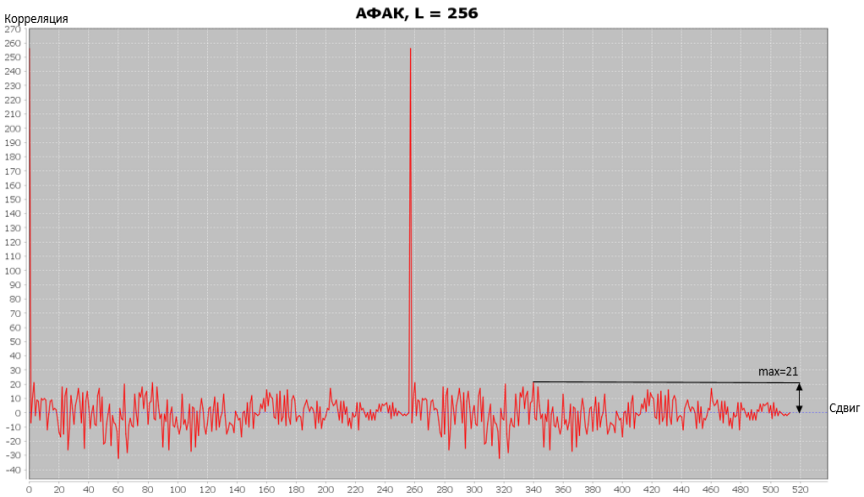


Рис. 3. АФАК КС для N = 256. Циклічний зсув{83}

Висновки

Оскільки кодовий поділ абонентів в багатокористувачевих інфокомунікаційних системах ґрунтується на відмінності сигналів, то побудова багатокористувачевих ІКС і показники

ефективності зазначених систем визначаються вибором сигналів і їх властивостями. Зазвичай число абонентів в сучасних інфокомунікаційних системах досить велике, тому вибір сигналів зводиться до визначення систем сигналів із заданими властивостями. Показано, що дискретні послідовності (ДП), властивості яких тотожні властивостям дискретних сигналів з необхідними характеристиками АФАК, насамперед з відповідними значеннями бокових піків АФАК, можуть бути відібрані з множини ДП, значення бокових піків ПФАК яких відповідають (13). Саме ці обставини було застосовано для оптимізації пошуку ДС з покращеними характеристиками АФАК. На основі застосування критерію мінімуму взаємних перешкод (мінімаксний критерій) та існуючих в теорії систем сигналів рівностей, що встановлюють залежність авто- і взаємно-кореляційних властивостей дискретних сигналів, вирішена задача оптимізації пошуку нелінійних дискретних сигналів с покращеними ансамблевими, структурними і кореляційними властивостями. Показано, що застосування синтезованих систем сигналів дозволить підвищити завадостійкість прийому сигналів (ймовірність правильного прийому сигналів) та показники інформаційної безпеки та скритності функціонування інформаційно-комунікаційних систем в умовах кібератак, дії природніх та організованих, в тому числі, структурних, ретрансльованих і інших завад. Показано, що при побудові криптографічних сигналів ІКС, до яких висуваються підвищені вимоги щодо інформаційної безпеки, завадостійкості прийому та скритності функціонування, до джерела сигналів слід включати всі відомі послідовності заданої довжини, рівень бічних пелюсток ПФАК яких відповідає граничним оптимальним значенням. У таку множину сигналів в якості кандидатів можуть увійти, зокрема, як свідчать представлені результати, і нелінійні дискретні сигнали, синтез яких заснований на використанні властивостей елементів кінцевого поля, а також сигнали, які створені із застосуванням випадкових (псевдовипадкових) процесів, криптографічних сигналів. Для кожної послідовності-кандидата шляхом циклічної перестановки його символів знаходять оптимальні за мінімаксним критерієм аперіодичні коди і відбирають з них найкращі.

Список літератури:

1. Sarvate D.V. Crosleration Properties of Pseudorandom and Related Sequences / D.V. Sarvate, M.V. Parsley // IEEE Trans. Commun. 1980. Vol. Com 68. P. 59–90.

2. Ipatov Valery P. Spread Spectrum and CDMA. Principles and Applications / Valery P. Ipatov. University of Turku, Finland and St. Petersburg Electro technical University ‘LETI’, Russia. John Wiley & Sons Ltd, the Atrium, Southern Gate, Chi Chester, West Sussex PO19 8SQ, England.

3. Варакин Л. Е. Системы связи с шумоподобными сигналами. Москва : Радио и связь, 1985. 384 с.

4. Свердлик М.Б. Оптимальные дискретные сигналы. Москва : Сов. радио, 1975. 200 с.

5. Горбенко І.Д., Замула О.А. Моделі та методи синтезу криптографічних сигналів та їх оптимізація за критерієм часової складності // Математичне та комп’ютерне моделювання. Серія: Фіз.-мат. науки: зб. наук. праць / Ін-т кібернетики імені В.М. Глушкова Національної академії наук України, 2017. Вип. 15. 272 с.

6. Gorbenko I.D., Zamula A.A. Cryptographic signals: requirements, methods of synthesis, properties, application in telecommunication systems // Telecommunications and Radio Engineering Volume 76, 2017. Issue 12, pages 1079-1100. DOI: 10.1615/TelecomRadEng.v76.i12.50.

7. Замула А.А. Перспективы применения нелинейных дискретных сигналов в современных телекоммуникационных системах и сетях / Замула А.А., Семенко Е.А // Системи обробки інформації. Харьков : ХУПС, 2015. Вип. 5 (130). С. 129–134.

8. Горбенко И.Д., Замула А.А. Аналитическая оценка значений максимальных боковых выбросов функций корреляции сложных нелинейных дискретных сигналов // Радиотехника. 2017. Вып. 191. С. 76 – 88.

9. Gorbenko I.D., Zamula A.A. Morozov Information security and noise immunity of telecommunication systems under conditions of various internal and external impacts // Telecommunications and Radio Engineering Volume 76, 2017 Issue 19, pages 1705-1717 DOI: 10.1615/TelecomRadEng.v76.i19.30.

Харківський національний університет
імені В.Н. Каразіна

Надійшла до редколегії 28.09.2019

УДК 004.491.4 **DOI:10.30837/rt.2019.3.198.14**

Р.С. ГРИНЬОВ, О.В. СЄВЄРІНОВ, канд. техн. наук

МЕТОД ПОДОЛАННЯ ЗАСОБІВ ЗАХИСТУ З ВИКОРИСТАННЯМ ВРАЗЛИВОСТЕЙ ГРАФІЧНИХ ФАЙЛІВ ФОРМАТУ BMP

Вступ

Незважаючи на використання засобів захисту, комп'ютерні віруси несуть серйозну загрозу як для простих користувачів, так і для інформаційних систем великих фірм і компаній. Сьогодні існує величезна різноманітність комп'ютерних вірусів, створюються нові методики їх приховування і поширення [1, 2]. Серед них найбільш частим є використання стеганографії для приховування вірусного програмного забезпечення в файлах [3].

На даний час для забезпечення захисту конфіденційних даних організацій використовуються антивірусне програмне забезпечення, системи виявлення вторгнень (IDS), системи запобігання вторгнень (IPS) та брандмауери [4]. Проте навіть вони не можуть гарантувати повного захисту.

Аналіз структури та особливостей файлів зображень формату BMP

Зловмисники для того, щоб приховати віруси додають до них різні нові функції та розробляють нові методики приховування шкідливого коду [5]. Для обходу антивірусних засобів, IDS/IPS і пісочниць зловмисники можуть впроваджувати вірусне програмне забезпечення в зображення.

Більшість методів аналізу файлів включають використання сигнатур вірусів і аналіз поведінки в пісочниці, а саме:

- перевірка поточного домену;
- перевірка запущених процесів;
- перевірка обсягу пам'яті;
- перевірка розміру диска;
- перевірка часу безвідмовної роботи.

Пісочниці зазвичай аналізують тільки виконувані файли, бібліотеки динамічних посилань (DLL), документи, що були створені в пакеті Microsoft Office та подібних, аплети Java [6]. Більшість із засобів захисту просто не реагують на файли зображень або інший безпечний тип файлів, оскільки вважається, що немає причин витрачати процесорний цикл на аналіз зображення [7].

Проаналізуємо можливість використання файлу зображення в якості контейнеру для непомітного транспортування комп’ютерного вірусу, ускладнення розбору інциденту інформаційної безпеки, приховування факту проникнення в систему, а також канали та методики, що при цьому використовуються.

Розглянемо формат файлів зображень BMP. Кожен файл цього формату має заголовок файлу, заголовок зображення, растрові данні та карту кольорів (крім зображень з 24-бітним кольором). Структура заголовку файлу формату BMP представлена в табл. 1. Він являє собою 14-байтну структуру, що знаходиться на початку файлу та містить в собі інформацію про тип та розмір файлу та розташування даних. Далі в файлі формату BMP знаходиться заголовок зображення, що містить інформацію про розмір, колір та стиск зображення (див. табл. 2).

В полі Compression визначається тип стиску. Якщо значення поля дорівнює 0, то стиск відсутній. Якщо значення RLE-4 або RLE-8, то використовується метод стиску груповими координатами відповідно із 4-біт/піксель та 8-біт/піксель.

Таблиця 1

Структура заголовку файлу формату BMP

Зміщення	Розмір (байт)	Ім’я	Опис
0	2	Type	Сигнатура формату. Використовується для ідентифікації формату. Має бути 4D42(hex)/424D(hex) (little-endian/big-endian). Після приведення до системи ASCII-символів має вигляд “BM”.
2	4	Size	Розмір файлу в байтах.
6	2	Reserved 1	Зарезервоване поле має містити 0.
8	2	Reserved 2	Зарезервоване поле має містити 0.
10	4	OffsetBits	Положення піксельних даних відносно початку файлу (в байтах).

Таблиця 2

Структура заголовку зображення

Зміщення	Розмір (байт)	Ім’я	Опис
14	4	Size	Довжина заголовку.
18	4	Width	Ширина зображення в пікселях.
22	4	Height	Висота зображення в пікселях.
26	2	Planes	Кількість площин.
28	2	BitCount	Глибина кольору, біт на піксель (1, 4, 8, 24).
30	4	Compression	Тип компресії (0 – відсутня, 1 – RLE-8, 2 – RLE-4).
34	4	SizeImage	Розмір зображення, байт (включно з доповненням).
38	4	XpelsPerMeter	Горизонтальна роздільна здатність, пікселів на метр.
42	4	YpelsPerMeter	Вертикальна роздільна здатність, пікселів на метр.
46	4	ColorsUsed	Число кольорів, що використовується (0 – максимально можливе для даної глибини кольору).
50	4	ColorTable	Кількість основних кольорів.

Найбільшу увагу привертають поля Size (розмір файлу BMP в байтах), XpelsPerMeter та YpelsPerMeter (горизонтальна та вертикальна роздільна здатність, пікселів на метр) та зарезервовані поля, адже вони є ненадійними [8]. Звичайний заголовок BMP починається з подібного рядка 42 4D XX XX XX XX 00 00 00 00.

Так наприклад, зображення з візерунком шахової дошки (рис. 1) має заголовок представлений на рис. 2.

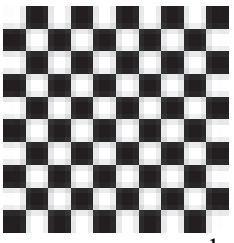


Рис. 1. Приклад малюнку в форматі BMP

```
Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Текст декодиров
00000000   42 4D 4A 13 00 00 00 00 00 00 8A 00 00 00 7C 00  BMJ.......Љ...|
00000010   00 00 28 00 00 00 28 00 00 00 01 00 18 00 00 00  ..(...(.........
00000020   00 00 C0 12 00 00 C3 0E 00 00 C3 0E 00 00 00 00  ..А...Г...Г.....
00000030   00 00 00 00 00 00 00 00 FF 00 00 FF 00 00 FF 00  ........я..я..я
00000040   00 00 00 00 00 FF 42 47 52 73 80 C2 F5 28 60 B8  .....яBGRsЂВх(`
```

Рис. 2. Заголовок файлу зображення

На рис. 3 представлені значення полів початку заголовку файлу з попереднього прикладу.

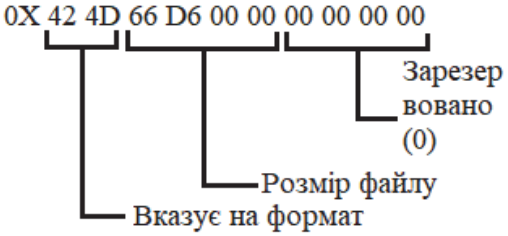


Рис. 3. Значення полів заголовку зображення

Існує можливість зміни зарезервованих полів та полів, що містять інформацію про розмір файлу в будь-якому hex-редакторі. Запишемо наприклад в ці ділянки слово “testtest” (рис. 4).

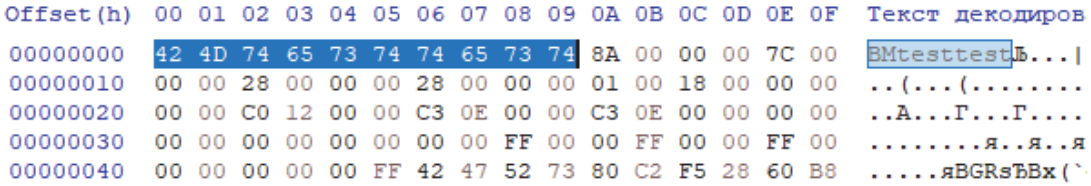


Рис. 4. Зображення зі зміненим заголовком

Тепер заголовок зображення матиме вигляд (рис. 5).

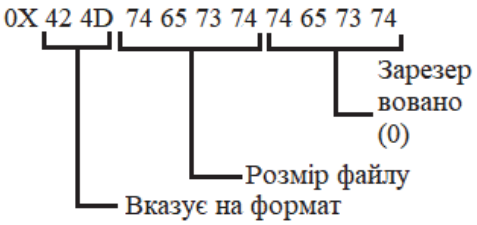


Рис. 5. Змінений заголовок зображення

Проте на самому зображенні не буде ніяких спотворень (рис. 6), оскільки поля службові і не впливають на відображення інформації. Крім того розмір файлу також не змінюється.

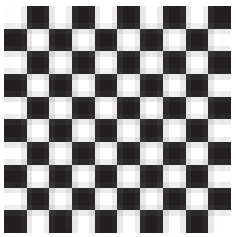


Рис. 6. Зображення зі зміненими полями

Як вже згадувалося раніше заголовок файлу BMP починається з 0x 42 4D, що вказує на тип файлу (BM). При конвертації в інструкції асемблера отримаємо, що 42 – це inc edx, а 4D – dec ebp. Це означає, що якщо ці інструкції будуть передувати коду програми, то вони не викличуть збоїв та не мають команд переходів.

В hex-редакторі змінимо значення декількох останніх рядків (рис. 7).

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Текст декодиров
00001240	13	13	E9	E9	E9	FF	FF	FF	FF	FF	FF	EC	EC	EC	16	16	..ййяяяяяяммм.
00001250	16	00	00	00	01	01	01	00	00	00	FC	FC	FC	FD	FD	FD	..........ьььээ
00001260	FF	FF	FF	EA	EA	EA	17	54	68	69	73	20	70	6C	61	63	яяяккк.This pla
00001270	65	20	69	73	20	65	6E	6F	75	67	68	20	74	6F	20	61	e is enough to
00001280	63	63	6F	6D	6D	6F	64	61	74	65	20	6D	61	6C	69	63	ccommodate mali
00001290	69	6F	75	73	20	63	6F	64	65	2C	20	77	68	69	63	68	ious code, whic
000012A0	20	77	69	6C	6C	20	62	65	20	68	69	64	64	65	6E	20	will be hidden
000012B0	69	6E	20	74	68	65	20	69	6D	61	67	65	2E	20	42	75	in the image. B
000012C0	74	20	64	75	65	20	74	6F	20	74	68	65	20	63	68	61	t due to the ch
000012D0	6E	67	65	73	20	74	68	65	72	65	20	77	69	6C	6C	20	nges there will
000012E0	62	65	20	6E	6F	74	69	63	65	61	62	6C	65	20	76	69	be noticeable v
000012F0	73	75	61	6C	20	64	69	73	74	6F	72	74	69	6F	6E	73	sual distortion
00001300	2E	20	49	66	20	61	72	74	69	66	69	63	69	61	6C	6C	. If artificial
00001310	79	20	72	65	64	75	63	65	20	74	68	65	20	68	65	69	y reduce the he
00001320	67	68	74	20	6F	66	20	74	68	65	20	69	6D	61	67	65	ght of the imag
00001330	2C	20	74	68	65	6E	20	74	68	65	79	20	63	61	6E	20	, then they can
00001340	62	65	20	68	69	64	64	65	6E	2E							be hidden.

Рис. 7. Змінене зображення

Відкривши отримане зображення ми побачимо, що у правому верхньому куті з’явилися спотворені пікселі (рис. 8).

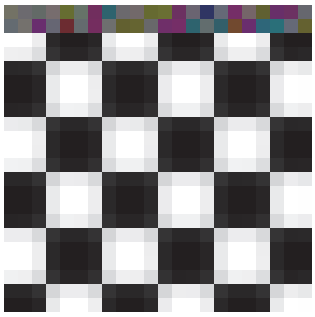


Рис. 8. Збільшений правий кут зміненого зображення

Скориставшись hex-редактором можна змінити висоту зображення і приховати спотворені пікселі (рис. 9).

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Текст декодиров
00000000	42	4D	74	65	73	74	74	65	73	74	8A	00	00	00	7C	00	BMtesttestЉ...|
00000010	00	00	28	00	00	00	26	00	00	00	01	00	18	00	00	00	..(...&.....
00000020	00	00	C0	12	00	00	C3	0E	00	00	C3	0E	00	00	00	00	..А...Г...Г....
00000030	00	00	00	00	00	00	00	00	FF	00	00	FF	00	00	FF	00	........я..я..я
00000040	00	00	00	00	00	FF	42	47	52	73	80	C2	F5	28	60	B8	.....яBGRsЂВх(`

Рис. 9. Штучне зменшення висоти зображення на 1 піксель

Після здійснення цих операцій при перегляді зображення спотворені пікселі не будуть відображатися на екрані (рис. 10).

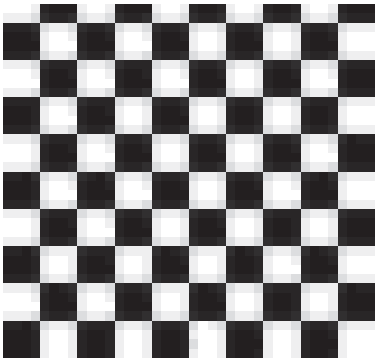


Рис. 10. Зображення зі штучно зменшеною висотою

Таким чином, використовуючи зображення формату BMP можна провадити в нього вірус, штучно зменшивши висоту зображення, який не буде помітним для користувача (на зображенні не буде спотворених пікселів). Ін'єкція можлива через те, що байти, які вказують на тип файлу, з яких і починається файл, BM в ASCII, в шістнадцятковому вигляді – 42 4D, при конвертації в інструкції асемблера не призводять до помилки виконання, а подальші 8 байт заголовка ніяк не впливають на інтерпретацію зображення. Ці 8 байт можна заповнити будь-якими інструкціями асемблера, наприклад, записати в них jmp-інструкцію, яка вкаже на вірус, що зберігається в зображенні [8]. Наприклад, в нашому випадку функція jmp могла вказувати на адресу 0000D583, де починається текст.

За допомогою нескладних дій можливо отримати зображення, що містить вірус. Така можливість обумовлена особливостями формату зображення BMP. Такий підхід приховування вірусу дозволить уникнути аналізу та виявлення шкідливого коду пісочницями. Проте IPS/IDS та антивіруси можуть виконати статичний аналіз та виявити вірус [8]. Для приховування шкідливого коду від цих засобів захисту необхідно використати обфускацію (заплутування коду) [7]. Найпростішим варіантом є використання найпростіших логічних операцій для кодування, таких як XOR. В кості ключа можна використати 32-х бітне значення в діапазоні від 0x11111111 до 0xffffffff. Обфускація буде досягатися за рахунок сили ключа, це дозволить уникнути жорстко закодованого коду.

Наприклад, припустимо, що ключ дорівнює 0x39643964. В цьому випадку маємо результат представлений на рис. 11.

$$
\begin{array}{ccccc}
0x54455445 & + & 0xCCCCCCCC & + & 0xCCCCCCCC \\
\oplus & & \oplus & & \oplus \\
0x39643964 & + & 0x39643964 & + & 0x39643964 \\
= & & = & & = \\
0x6d216d21 & + & 0xf5a8f5a8 & + & 0xf5a8f5a8
\end{array}
$$

Рис. 11. Приклад результату операції обфускації

Щоб виконати код, що зберігається в зображенні, можна використати корисне навантаження PowerShell, воно завантажить зображення та виконає код в пам’яті. Ми використаємо готове рішення від Cobalt Strike. Воно використовує System.Net.WebClient для завантаження файлу, в нашому випадку це зображення, а потім VirtualAlloc та CreateThread для зчитування та виконання вірусу.

Проведений аналіз підтверджує можливість використання зображень формату BMP з впровадженим вірусним кодом для подолання засобів захисту. Існує багато варіантів завантаження зображення з впровадженим вірусом та виконання вірусного коду. Це можна зробити за допомогою будь-якого виконуваного файлу або скрипта, впровадити код у вже існуючий виконуваний файл. Такі методи дозволять оминути засоби захисту. Адже під час сканування засоби захисту не виявлять у них вірус. Проте більш перспективним є використання апаратних закладок у поєднанні з цим методом [9, 10].

Аналіз виявлення вірусних зразків різними антивірусними засобами

В ході дослідження була розроблена програма, яка отримує в якості вхідних даних необроблене корисне навантаження, що може бути згенеровано за допомогою Metasploit та зображення формату BMP. Тестування проводилось на персональному комп’ютері під керуванням операційної системи Parrot Security OS. Metasploit – це інструмент, призначений для створення та використання експлойтів, а також експлуатації та постексплуатації вразливостей. Для створення вхідних даних також можна скористатися утилітою Msfvenom, яка є складовою Metasploit [11]. Розроблена програма впроваджує в зображення вірус, змінює ви-

соту зображення та записує в заголовок jump-функцію. Після чого запускає web-сервер на комп'ютері з інфікованим зображенням та створює команду PowerShell, яка завантажую зображення та виконує вірус.

Для перевірки файлів використовувались ресурси web-сайту www.virustotal.com. Він надає можливість аналізу файлу по базам даних всіх популярних антивірусних засобів та безпосереднього аналізу файлу антивірусами. Результат сканування шелл-коду в форматі raw, що був згенерований за допомогою msfvenom представлені на рис. 12.

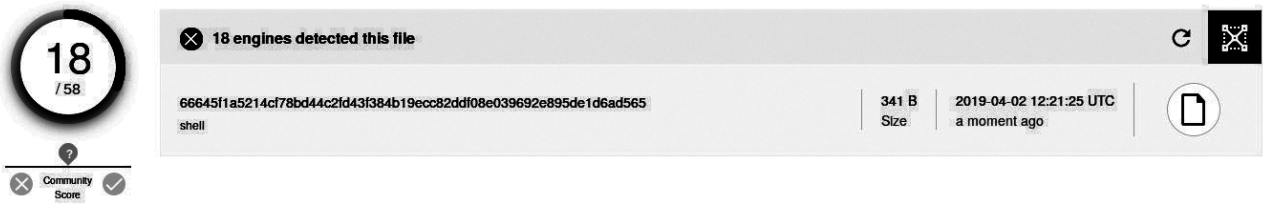


Рис. 12. Результат сканування шелл-коду в форматі raw

18 з 58 антивірусних засобів визначили тестовий файл, як вірусний. Той самий вірусний код але у вигляді виконуваного файлу визначили вірусним вже 53 з 70 антивірусів (рис. 13). Різна кількість антивірусних засобів, що перевіряла файл зумовлена типом файлу.

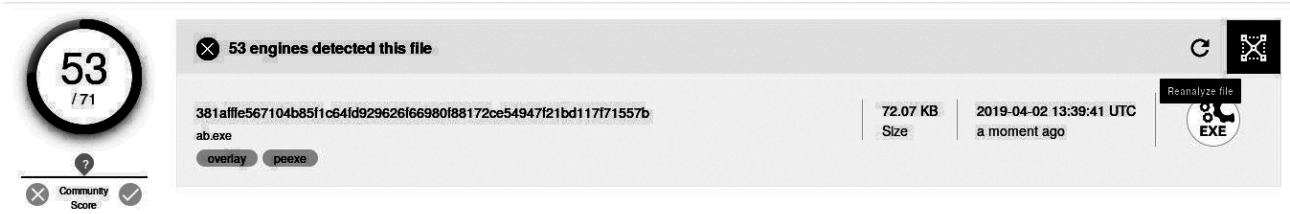


Рис. 13. Результати сканування шелл-коду в форматі exe

В дослідженнях була перевірена можливість виявлення вірусу, що був прихований в програму PuTTY – безкоштовний клієнт для протоколів Telnet, SSH. Результати перевірки оригінального файлу PuTTY представлені на рис. 14.

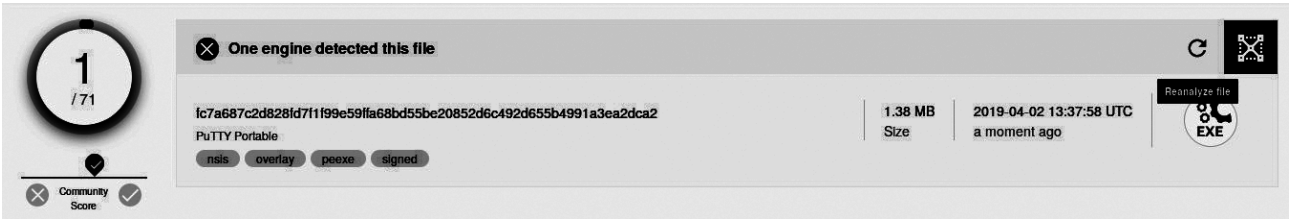


Рис. 14. Результати сканування оригінального PuTTY

Як видно з результатів, лише один антивірус зробив припущення про наявність вірусу (рис. 15).

Рис. 15. Припущення про наявність вірусу

Для впровадження вірусу у виконуваний файл були використані дві найпоширеніші методики, що дозволяють зберегти працездатність файлу: створення нової секції та впровадження у code cave. Code cave – це послідовність нульових байтів в пам'яті процесу. Для впровадження вірусу була використана утиліта Backdoor Factory – спеціальна утиліта для впровадження шелл-коду у виконувані файли та динамічні бібліотеки. Спочатку був видале-

ний цифровий підпис файлу. Результати перевірки PuTTY з видаленим цифровим підписом представлені на рис. 16.

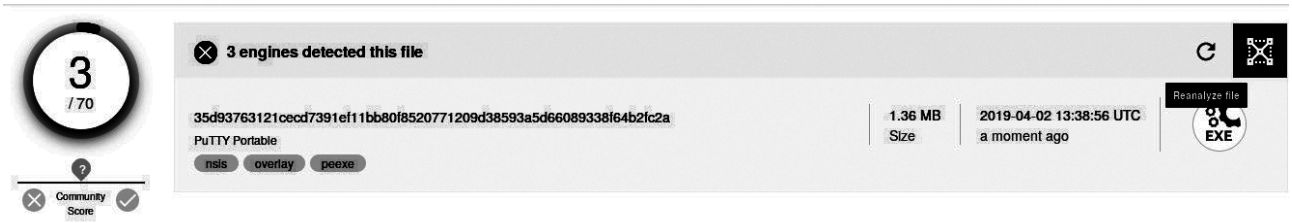


Рис. 16. Перевірка PuTTY з видаленим цифровим підписом

На рис. 17 представлені результати сканування файлу PuTTY з новою порожньою секцією (без вірусу) в оригінальному файлі.

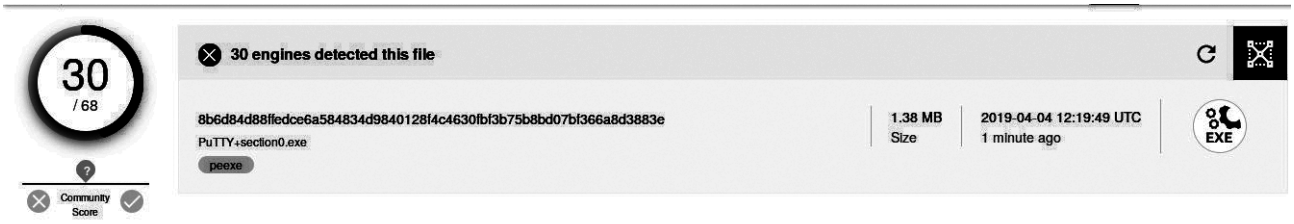


Рис. 17. Перевірка PuTTY з новою порожньою секцією

Як видно з результатів після створення нової порожньої секції без впровадження вірусного коду файл визначається 30 антивірусними засобами як вірус. Після створення нової секції та впровадження в неї вірусу 32 з 70 антивірусних засобів його виявили (рис. 18).

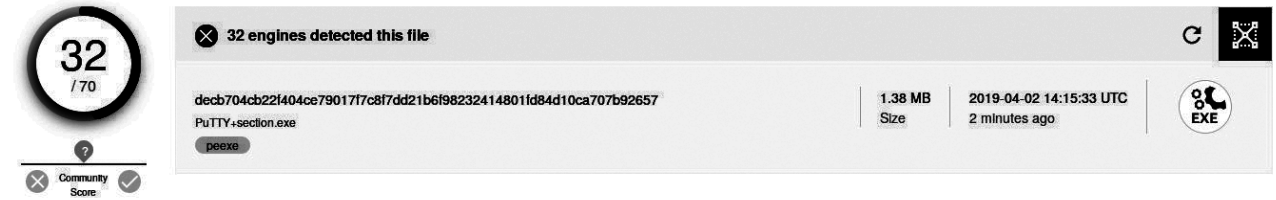


Рис. 18. Результати приховування вірусу в новій секції

В дослідженнях була перевірена можливість виявлення вірусу, що був прихований у code cave (рис. 19).

```
└──╼ $ backdoor-factory -f PuTTY.exe -s user_supplied_shellcode_threaded -U sect
ion  -o PuTTY+cave.exe -Z
```

Рис. 19. Впровадження вірусу в code cave

За результатами сканування лише 9 антивірусів вивили вірус (рис. 20).

Рис. 20. Результати приховування вірусу в code cave

В дослідженнях був перевірений оригінальний файл BMP (рис. 21). Після цього був перевірений файл, в який був впроваджений шелл-код за допомогою розробленої програми (рис. 22).

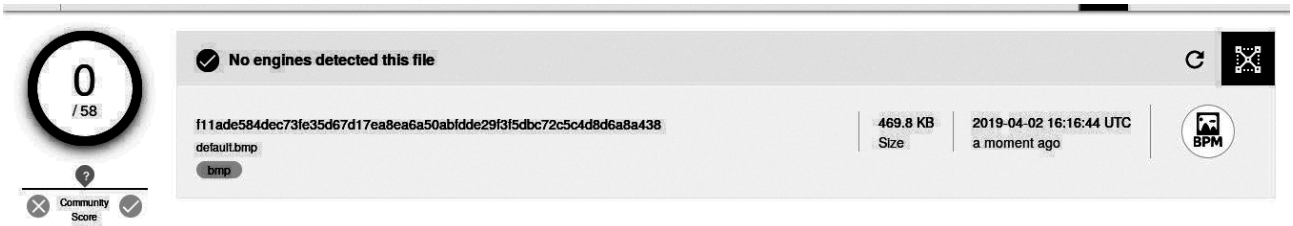


Рис. 21. Результати перевірки оригінального зображення BMP

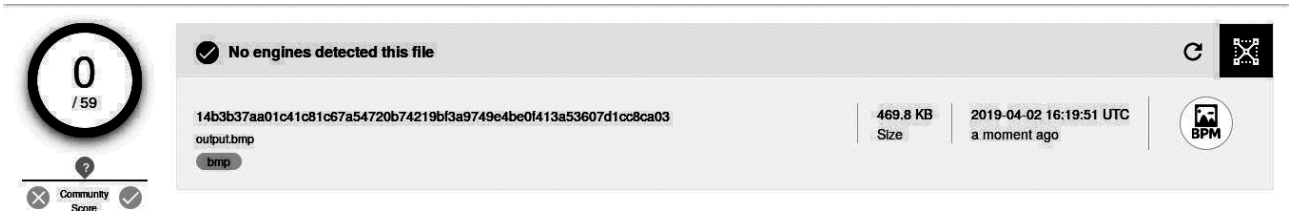


Рис. 22. Результати перевірки інфікованого зображення BMP

Як видно з результатів перевірки жоден антивірусний засіб не зміг знайти вірус у зображенні. Що означає, що метод приховування вірусів від засобів захисту, в тому числі й антивірусів, розроблений в цій роботі є найбільш ефективним.

Дослідження показали, що детальна інформація про оригінальне (рис. 23, *а*) та інфіковане зображення (рис. 23, *б*), відрізняється лише значеннями висоти в пікселях та значеннями хеш функцій, вирахуваних від файлів.

а *б*

Рис. 23. Детальна інформація про зображення

Узагальнені результати проведених перевірок наведено у табл. 3. Знаком “-” позначається, якщо антивірус не виявив вірус, знаком “+” , якщо вірус був виявлений, знаком “*”, якщо антивірус не обробляє цей тип файлу.

Таблиця 3

Результати сканування зразків

Назва антивірусу	Шелл-код у фо-рматі raw	Шелл-код у фо-рматі exe	PuTTY	PuTTY без циф-рового підпису	PuTTY з порож-ньою новою секцією	PuTTY з шелл-кодом в новій секції	PuTTY з шелл-кодом в code cave	Оригінальне BMP зображення	Інфіковане BMP зобра-ження
Acronis	*	+	-	-	-	-	-	*	*
Ad-Aware	+	+	-	-	-	-	-	-	-
AegisLab	+	+	-	-	+	+	+	-	-
AhnLab-V3	+	+	-	-	-	-	-	-	-
Alibaba	*	+	-	-	-	-	-	*	*
ALYac	+	+	-	-	-	-	-	-	-
Antiy-AVL	-	-	-	-	-	-	-	-	-
Arcabit	+	+	-	-	-	-	-	-	-
Avast	+	+	-	-	+	+	-	-	-
Avast Mobile Security	-	-	-	-	-	-	-	-	-
AVG	+	+	-	-	+	+	-	-	-
Avira	-	+	-	-	+	+	-	-	-
Babable	-	-	-	-	-	-	-	-	-
Baidu	-	-	-	-	-	-	-	-	-
BitDefender	+	+	-	-	-	-	-	-	-
Bkav	-	+	-	+	+	+	+	-	-
CAT-QuickHeal	-	+	-	-	-	-	-	-	-
ClamAV	+	+	-	-	+	+	-	-	-
CMC	-	-	-	-	-	-	-	-	-
Comodo	-	+	-	-	+	+	-	-	-
CrowdStrike Falcon	*	+	-	-	+	+	+	*	*
Cybereason	*	+	-	-	-	-	-	*	*
Cylance	*	+	-	-	+	+	+	*	*
Cyren	-	+	-	-	+	+	-	-	-
DrWeb	+	+	-	-	+	+	-	-	-
eGambit	*	+	-	-	+	+	+	*	*
Emsisoft	+	+	-	-	-	-	-	-	-
Endgame	*	+	-	-	+	+	-	*	*
eScan	+	+	-	-	-	-	-	-	-
ESET-NOD32	-	+	-	-	+	+	-	-	-
F-Prot	-	+	-	-	-	+	-	-	-
F-Secure	-	+	-	-	+	+	-	-	-
FireEye	+	+	-	-	+	+	-	-	-
Fortinet	-	+	-	-	+	+	-	-	-
GData	+	+	-	-	-	-	-	-	-
Ikarus	-	+	-	-	-	+	+	-	-
Jiangmin	-	-	-	-	-	-	-	-	-
K7AntiVirus	-	+	-	-	-	-	-	-	-
K7GW	-	+	-	-	-	-	-	-	-
Kaspersky	+	+	-	-	+	+	-	-	-
Kingsoft	-	-	-	-	-	-	-	-	-
Malwarebytes	-	-	-	-	-	-	-	-	-
MAX	+	+	-	-	-	-	-	-	-
MaxSecure	*	-	-	-	-	-	-	*	*
McAfee	-	+	-	-	+	+	-	-	-
McAfee-GW-Edition	-	+	-	+	+	+	-	-	-
Microsoft	+	+	-	-	+	+	+	-	-

Продовження табл. 3

Назва антивірусу	Шелл-код у фор-маті raw	Шелл-код у фо-рматі exe	PuTTY	PuTTY без циф-рового підпису	PuTTY з порожньою новою секцією	PuTTY з шелл-кодом в новій секції	PuTTY з шелл-кодом в code cave	Оригінальне BMP зображення	Інфіковане BMP зобра-ження
NANO-Antivirus	+	+	-	-	+	+	-	-	-
Palo Alto Networks	*	-	-	-	-	-	-	*	*
Panda	-	-	-	-	-	-	-	-	-
Symantec	+	-	-	-	-	-	-	*	*
Symantec Mobile Insight	*	-	-	-	-	-	-	*	*
Qihoo-360	-	+	-	-	+	+	-	-	-
Rising	-	+	+	+	+	+	+	-	-
SentinelOne	*	+	-	-	+	+	-	*	*
Sophos AV	-	+	-	-	+	+	-	-	-
Sophos ML	*	+	-	-	-	-	-	*	*
SUPERAntiSpyware	-	+	-	-	-	-	-	-	-
Symantec	*	+	-	-	+	+	+	-	-
TACHYON	-	-	-	-	-	-	-	-	-
Tencent	+	-	-	-	-	-	-	-	-
TheHacker	-	-	-	-	-	-	-	-	-
TotalDefense	-	-	-	-	-	-	-	*	*
Trapmine	*	+	-	-	-	-	-	*	*
TrendMicro	-	+	-	-	-	-	-	-	-
TrendMicro-HouseCall	-	+	-	-	-	-	-	-	-
Trustlook	*	-	-	-	-	-	-	*	*
VBA32	-	-	-	-	-	-	-	-	-
VIPRE	-	-	-	-	+	-	-	-	-
ViRobot	-	+	-	-	-	-	-	-	-
Webroot	*	+	-	-	-	-	-	*	*
Yandex	-	+	-	-	+	+	-	-	-
Zillya	-	-	-	-	-	+	-	-	-
ZoneAlarm	+	+	-	-	+	+	-	-	-
Zoner	-	-	-	-	-	-	-	-	-

Висновки

У ході досліджень була розроблена та продемонстрована концепція атаки, що дозволяє приховати вірус у зображеннях BMP. Така можливість зумовлена особливостями структури даного формату.

Проаналізувавши отримані результати, можна стверджувати, що даний метод приховування вірусу дозволяє подолати всі засоби захисту та приховати шкідливий код. Крім того подібний підхід може суттєво ускладнити розбір інциденту інформаційної безпеки. Це обумовлено тим, що більшість із засобів захисту просто не звертають уваги на зображення або інший безпечний тип файлу. Оскільки вважають, що немає причин витрачати процесорний цикл на аналіз зображення. Антивірусні засоби захисту під час сканування не виявили вірус у файлі формату BMP. Крім того аналіз поведінки зображення в "пісочниці" також не виявить вірус, оскільки при відкритті зображення шкідливий код не почне виконуватися. Враховуючи вище сказане IDS, IPS також не зможуть виявити вірус. Це свідчить про неефективність виявлення та протидії існуючих засобів захисту розглянутій атаці.

За результатами випробувань можна стверджувати, що розроблений метод приховування вірусного коду в зображенні формату BMP є найбільш ефективним, оскільки шелл-код не був виявлений жодним засобом захисту. Результати даної роботи можна використовувати під час розробки засобів антивірусного захисту та комплексних засобів захисту ІТС та для їх модернізації з метою попередження подібних атак.

При цьому необхідно враховувати, що навіть звичайні файли з простою структурою без підтримки скриптів можуть становити серйозну загрозу. Тому необхідно розроблювати нові більш ефективні засоби захисту.

Основна небезпека подібних зображень з вірусами полягає в тому, що для виявлення загрози необхідно використовувати нестандартні методи. Можна змінити налаштування засобів захисту, щоб вони перевіряли всі типи файлів, але це суттєво сповільнить або навіть повністю паралізує роботу всієї інформаційно-комунікаційної системи.

Список літератури:

1. Гриньов Р.С. Аналіз тенденцій вірусних загроз в Україні / Гриньов Р.С., Сєвєрінов О.В. // Сучасні напрямки розвитку інформаційно-комунікаційних технологій та засобів управління: міжнар. конф. Харків, 2019. С. 100.

2. Гриньов Р.С. Аналіз статистики та особливостей розповсюдження вірусів в Україні // Сучасні напрямки розвитку інформаційно-комунікаційних технологій та засобів управління: міжнар. конф. Харків, 2019. С. 100.

3. Pare. Virus spread over networks: Modeling, analysis, and control : Ph.D. Electrical & Computer Eng / Pare. University of Illinois at Urbana-Champaign, 2018.

4. Jingwei LEI. Virus program detection method, terminal, and computer readable storage medium. United States, 2018. 19 c.

5. Wen-Kwang Tsao. Detecting malicious code in sections of computer files / Wen-Kwang Tsao, Pinghuan Wu, Zipan Bai. United States, 2018. 15 c.

6. Lubomir Sikora. Swarm Virus, Evolution, Behavior and Networking / Lubomir Sikora, Ivan Zelinka. Berlin, 2017.

7. Carey Parker. Computer Security. North CarolinaUSA, 2018.

8. Гриньов Р.С. Аналіз небезпеки впровадження вірусного програмного забезпечення в зображення / Гриньов Р.С., Севєрінов О.В. // Комп'ютерні та інформаційні системи і технології: міжнар. наук.-техн. конф. Харків, 2019. С. 75.

9. Гриньов Р.С. Шкідливий USB HID-емулятор / Гриньов Р.С., Сєвєрінов О.В. // Радіоелектроніка та молодь у XXI столітті: міжнар. форум. Харків, 2018. С. 120-121.

10. Гриньов Р.С. Аналіз небезпеки апаратних закладних пристроїв / Гриньов Р.С., Сєвєрінов О.В. // Радіоелектроніка та молодь у XXI столітті: міжнар. форум. Харків, 2019. С. 93-94.

11. Офіційна документація Parrot Security OS. URL: https://docs.parrotsec.org/doku.php (дата звернення: 22.10.2018).

Харківський національний
університет радіоелектроніки

Надійшла до редколегії 04.09.2019

УДК 004.056 **DOI:10.30837/rt.2019.3.198.15**

В.А. КУЛІБАБА

ПОРІВНЯЛЬНИЙ АНАЛІЗ КРИПТОПЕРЕТВОРЕНЬ НА ЕЛІПТИЧНИХ КРИВИХ ТА КРИВИХ ЕДВАРДСА

Вступ

Останнім часом все більш гостро постає проблема швидкодії асиметричних криптографічних механізмів і, передусім, цифрових підписів. Це питання постає через те, що неминучим є збільшення для забезпечення криптографічної стійкості в умовах постійно зростаючих можливостей порушників довжин параметрів. Реальна поява квантового комп'ютера ставить перед дослідниками нові задачі: розробити, стандартизувати та впровадити нові механізми захисту інформації, стійкі, в тому числі, в постквантовий період. Проте, в так званий перехідний період до появи цих нових стандартів будуть використовуватися механізми цифрового підпису, в основі яких лежить математика еліптичних кривих. Новим запропонованим математичним апаратом є криві Едвардса, які мають певні переваги перед еліптичними кривими [2, 6]. Метою даної статті є огляд та порівняння криптографічних перетворень в групі точок еліптичних кривих та кривих Едвардса, а також попередня оцінка криптографічної стійкості кривих Едвардса.

Криві Едвардса та їх переваги

В оригінальному вигляді [2] криві Едвардса були запропоновані у вигляді

$$x^2 + y^2 = e^2(1 + x^2 y^2) \tag{1}$$

Закон складання точок для кривої у формі Едвардса визначений як

$$P_1(x_1, y_1) + P_2(x_2, y_2) = P_3\left(\frac{x_1 y_2 + x_2 y_1}{e(1 + x_1 x_2 y_1 y_2)}, \frac{y_1 y_2 - x_1 x_2}{e(1 - x_1 x_2 y_1 y_2)}\right) \tag{2}$$

Також було введено поняття так званого нейтрального елемента $0 = (0, e)$. Операція знаходження зворотної точки ЕК визначається як

$$-P(x_p, y_p) = P(-x_p, y_p) \tag{3}$$

на відміну кривих в класичній формі, де

$$-P(x_p, y_p) = P(x_p, -y_p) \tag{4}$$

Операція подвоєння точки із (2) визначається як

$$2P(x, y) = \left(\frac{2xy}{e(1 + x^2, y^2)}, \frac{y^2 - x^2}{e(1 - x^2 y^2)}\right) \tag{5}$$

Проте, як зазначається в [5], криві в оригінальній формі Едвардса мають ряд вагомих недоліків і не можуть бути застосовані в криптографічних додатках.

Для практичного застосування в криптографії в [1] обгрунтовано модифікацію кривих Едвардса. Зокрема, для циклічності групи точок кривої та видалення особливих точок було введено додатковий коефіцієнт *d*, такий, що символ Лагранжа $\left(\frac{d}{p}\right) = -1$, тобто *d* э квадратичним невичетом, а також на нього накладено обмеження $d(1 - de^4) \neq 0$. Рівняння кривої в модифікованому вигляді має вид

$$x^2 + y^2 = e^2(1 + dx^2 y^2) \tag{6}$$

Аналогічно з (2) закон складання точок кривої Едвардса визначається як

$$P_1(x_1, y_1) + P_2(x_2, y_2) = P_3(\frac{x_1y_2 + x_2y_1}{e(1 + dx_1x_2y_1y_2)}, \frac{y_1y_2 - x_1x_2}{e(1 - dx_1x_2y_1y_2)}), \quad (7)$$

А операція подвоєння точки визначається як

$$2P(x, y) = \left(\frac{2xy}{e(1 + dx^2, y^2)}, \frac{y^2 - x^2}{e(1 - dx^2y^2)} \right) \quad (8)$$

Операція знаходження оберненої точки визначається згідно (3).

Порівняння складності перетворень в групі точок еліптичних кривих в класичному вигляді та кривих Едвардса

Груповий закон у проективних координатах для класичних еліптичних кривих.

Особливістю проективного базису для класичних ЕК є те, що при використанні проективних координат необхідно виконувати більше операцій множення, але немає операції ділення за модулем (інверсії), що дає виграш у швидкодії. Після виконання скалярного множення в проективному базисі необхідно зробити зворотне перетворення на афінні координати [6].

Проективний аналог короткого афінного рівняння Вейєрштраса (4) визначається [6]:

$$Y^2Z = X^3 + aXZ^2 + bZ^3, \quad a, b \in F(q). \quad (9)$$

Еліптична крива, що задається в проективних координатах, складається з усіх точок $R = (X, Y, Z)$ рівняння (9) так, що трійка (X, Y, Z) є розв'язком рівняння.

У проективних координатах груповий закон задається наступним чином:

1) точка $(0_F, 1_F, 0_F)$ є одиничним елементом 0_E відносно операції «+»;

2) точка $R = (X, Y, Z) \neq (0_F, 1_F, 0_F)$ є точкою на кривій E, що задана в проективних координатах, тоді обернена точка $-R = (X, -Y, Z)$;

3) нехай $R_1 = (X_1, Y_1, Z_1)$ і $R_2 = (X_2, Y_2, Z_2)$ є дві різні точки на E – такі, що $R_1 \neq R_2$ і $R_1, R_2 \neq (0_F, 1_F, 0_F)$, тоді сума R_1 та R_2 є $R_3 = (X_3, Y_3, Z_3)$. Координати X_3, Y_3 і Z_3 можуть бути обчислені як

$$\begin{aligned} X_3 &= -su, \\ Y_3 &= t(u + s^2X_1Z_2) - s^3Y_1Z_2, \\ Z_3 &= s^3Z_1Z_2, \end{aligned} \quad (10)$$

де $s = X_2Z_1 - X_1Z_2$, $t = Y_2Z_1 - Y_1Z_2$, і $u = s^2(X_1Z_2 + X_2Z_1) - t^2Z_1Z_2$; якщо $R = (X, Y, Z) \neq (0_F, 1_F, 0_F)$ є точкою на E, тоді її подвоєння є $2R = (X_3, Y_3, Z_3)$.

Координати точки $2R - (X_3, Y_3, Z_3)$ можуть бути обчислені за допомогою таких формул:

$$\begin{aligned} X_3 &= -su, \\ Y_3 &= t(u + s^2X) - s^3Y, \\ Z_3 &= s^3Z, \end{aligned} \quad (11)$$

де $t = 3X^2 + aZ^2$, $s = 2YZ$, а також $u = 2s^2X - t^2Z$.

Згідно [1] повні криві Едвардса визначені як

$$x^2 + y^2 = 1 + dx^2y^2; d(1 - d) \neq 0; \left(\frac{d}{p}\right) = -1, \quad (12)$$

а також було введено обмеження $\left(\frac{a}{p}\right) = \left(\frac{d}{p}\right) = -1$.

Для повної кривої Едвардса (12) закон складання точок визначено у вигляді [4]

$$P(x_1, y_1) + Q(x_1, y_2) = Z\left(\frac{x_1 ч_2 + y_1 y_2}{1 - dx_1 x_2 y_1 y_2}, \frac{x_1 y_1 + x_2 y_2}{1 + dx_1 x_2 y_1 y_2}\right) \tag{13}$$

$$2P(x, y) = \left(\frac{x_1^2 - y_1^2}{1 - dx_1^2 y_1^2}, \frac{2x_1 y_1}{1 + dx_1^2 y_1^2}\right) \tag{14}$$

Так як в афінних координатах присутнє знаходження оберненого елемента в полі, то прийнято переходити до проективних координат. Це пояснюється тим, що [4] операція інверсії в полі з обчислювальної точки зору є найбільш складною. Причому, заміна $x = \frac{x}{z}, y = \frac{y}{z}$ дозволяє перейти в проективні координати. В такому випадку рівняння кривої має вигляд

$$(X^2 + Y^2)Z^2 = Z^4 + dX^2 Y^2, X = xZ, Y = yZ\,. \tag{15}$$

Закон складання точок в проективних координатах має вигляд

$$P(X_1, Y_1, Z_1) + Q(X_2, Y_2, Z_2) = (X_3, Y_3, Z_3)\,.$$

Після підстановки [4] маємо:

$$y_3 = \frac{Y_3}{Z_3} = \frac{\left(\frac{X_1 Y_2}{Z_1 Z_2} + \frac{X_2 Y_1}{Z_1 Z_2}\right)\left(1 - d\frac{X_1 X_2 Y_1 Y_2}{Z_1^2 Z_2^2}\right)}{\left(1 + d\frac{X_1 X_2 Y_1 Y_1}{Z_1^2 Z_2^2}\right)\left(1 - d\frac{X_1 X_2 Y_1 Y_2}{Z_1^2 Z_2^2}\right)} = \frac{Z_1 Z_2 (Z_1^2 Z_2^2 - dX_1 X_2 Y_1 Y_2)(X_1 Y_2 + X_2 Y_1)}{(Z_1^2 Z_2^2 + dX_1 X_2 Y_1 Y_2)(Z_1^2 Z_2^2 - dX_1 X_2 Y_1 Y_2)} \tag{16}$$

$$x_3 = \frac{X_3}{Z_3} = \frac{Z_1 Z_2 (Z_1{}^2 Z_2{}^2 + dX_1 X_2 Y_1 Y_2)(X_1 X_2 - Y_1 Y_2)}{(Z_1{}^2 Z_2{}^2 + dX_1 X_2 Y_1 Y_2)(Z_1{}^2 Z_2{}^2 - dX_1 X_2 Y_1 Y_2)}\ . \tag{17}$$

Нехай $A = Z_1 Z_2; B = A^2; C = X_1 X_2; D = Y_1 Y_2; E = dCD; F = B - E; G = B + E$, тоді координати точок виражаються через

$$\begin{aligned} Y_3 &= A \cdot F((X_1 + Y_1) \cdot (X_2 + Y_2) - C - D) \\ X_3 &= A \cdot G \cdot (D - C) \\ Z_3 &= F \cdot G \end{aligned} \tag{18}$$

Після підрахунку елементарних операцій в полі маємо, що $V_E = 10M + 1S + 1U$ операцій в полі. При цьому можна прийняти[5], що $1S \approx \frac{2}{3}M$.

За аналогічною методикою виконуються оцінки подвоєння точок:

$$x_3 = \frac{X_3}{Z_3} = \frac{\left(\left(\frac{X_1}{Z_1}\right)^2 - \left(\frac{Y_1}{Z_1}\right)^2\right)\left(\left(\frac{X_1}{Z_1}\right)^2 + \left(\frac{Y_1}{Z_1}\right)^2\right)}{\left(2 - \left(\frac{X_1}{Z_1}\right)^2 - \left(\frac{Y_1}{Z_1}\right)^2\right)\left(\left(\frac{X_1}{Z_1}\right)^2 + \left(\frac{Y_1}{Z_1}\right)^2\right)} = \frac{\left(X_1^2 - Y_1^2\right)\left(X_1^2 + Y_1^2\right)}{\left(2Z_1^2 - X_1^2 - Y_1^2\right)\left(X_1^2 + Y_1^2\right)} \tag{19}$$

$$y_3 = \frac{Y}{Z} = \frac{2\frac{X_1 Y_1}{X_1}\left(2-\left(\frac{X_1}{Z_1}\right)^2-\left(\frac{Y_1}{Z_1}\right)^2\right)}{\left(2-\left(\frac{X_1}{Z_1}\right)^2-\left(\frac{Y_1}{Z_1}\right)^2\right)\left(\left(\frac{X_1}{Z_1}\right)^2+\left(\frac{Y_1}{Z_1}\right)^2\right)} = \frac{2X_1Y_1(X_1^2+Y_1^2)}{(2Z_1^2-X_1^2-Y_1^2)(X_1^2+Y_1^2)} \quad (20)$$

Після виконання заміни [3, 5] та підрахунку кількості операцій складність подвоєння точки на ЕК в формі Едвардса оцінюється як $T_E = 3M + 4S$. За аналогічною методикою оцінюється складність виконання операцій додавання та подвоєння точок в проективних координатах для еліптичних кривих в канонічній формі.

Зведені дані щодо складності операцій наведено в табл. 1, де для спрощення використано такі співвідношення [5]: 1S = 0.67M та $1U$ = 0.5M.

Таблиця 1

Порівняння складності операцій множення і додавання точок ЕК
в кривих Едвардса та канонічних кривих

	Криві у формі Вейерштрасса		Криві у формі Едвардса	
	Додавання	Множення	Додавання	Множення
1	*12M+2S*	*7M+5S*	*10M+1S+1U*	*3M+4S*
2	*13.33M*	*10.33M*	*11.57M*	*5.67M*

Як зазначено в роботі [5], криві у формі Едвардса мають значний виграш у складності виконання операцій в групі точок еліптичної кривої, приріст швидкодії сягає 1,5 рази, що, безумовно, є важливим з точки зору практичних застосувань.

Порівняльний аналіз стійкості класичних ЕК та ЕК Едвардса та складності атак

Відомо [6], що найбільш універсальною атакою на асиметричні крипто перетворення є атаки типу «повне розкриття», яка зводиться до дискретного логарифмування в групі точок еліптичної кривої, а також для електронного підпису до атаки «повне розкриття» на основі підписаних даних. Тому в якості основної загрози будемо розглядати загрозу визначення особистого ключа *d* шляхом дискретного логарифмування у відповідних групах кривих.

Нехай порушникові відомо порядок базової точки еліптичної кривої, відкритий ключ Q, а також всі загальносистемні параметри, включаючи модуль перетворення q. Він також знає, що особистий та відкритий ключі пов'язані між собою співвідношенням

$$Q_i = d_i \cdot G(\mathrm{mod} q) . \quad (21)$$

У [3] показано, що еліптичні криві Едвардса, які є придатними до застосувань у крипторгафії, є ізоморфними до ЕК у формі Вейерштрасса, до них можуть бути застосовані ті ж самі методи криптоаналізу, тобто дискретного логарифмування в групі точок.

За цієї умови для знаходження особистого ключа d скористаємося методом ро-Поларда як одним з найбільш ефективних [4]. Суть його полягає у наступному.

Нехай для деякого кінцевого набору W є відображення $F: W \to W$. Сама послідовність формується за правилом $w_0 \in W, w_{k+1} = F(w_k)$. Далі, нехай n – порядок базової точки G, а відкритий ключ Q є точкою на цій кривій. Тоді для знаходження секретного параметра *d* Поллардом було введено інтерполяційну функцію

$$F(Y) = \begin{Bmatrix} G+Y, 1 \le Y(x) \le n \\ 2Y, n/3 \le Y(x) \le 2n/3 \\ Q+Y, 2n/3 \le Y(x) \le n \end{Bmatrix}. \quad (22)$$

Правило розбиття за інтервалами може бути обрано інше, а замість *x*-координати точки використовувати *y*-координату. Як можна зазначити, у 2/3 випадків використовується додавання точок, а в 1/3 – подвоєння.

Існує також альтернативна форма завдання рекурентних співвідношень для обчислення наступної точки:

$$Y_k = \alpha_k G + \beta_k Q,$$
$$\alpha_{k+1} = \begin{cases} \alpha_k, Y_k \in S_1 \\ 2\alpha_k, Y_k \in S_2 \\ \alpha_k + 1, Y_k \in S_3 \end{cases} \qquad (23)$$
$$\beta_{k+1} = \begin{cases} \beta_k + 1, Y_k \in S_1 \\ 2\beta_k, Y_k \in S_2 \\ \beta_k, Y_k \in S_3 \end{cases}$$

Використання (22) чи (23) дозволяє побудувати дві послідовності, збіг значень в яких $Y_i = Y_j$ при різних індексах дозволяє створити колізію та визначити особистий ключ [6]:

$$d = \frac{\alpha_j - \alpha_i}{\beta_i - \beta_j} G (\mathrm{mod}\, n). \qquad (24)$$

Також показано, що ймовірність знаходження колізії при *k* спробах складає [6]:

$$P(n,k) = 1 - \frac{n}{n} \cdot \left(\frac{n-1}{n}\right) \cdot \left(\frac{n-2}{n}\right) \ldots \left(\frac{n(k-1)}{n}\right) = 1 - \left(1 - \frac{1}{n}\right) \cdot \left(\frac{2}{n}\right) \ldots \left(\frac{n(k-1)}{n}\right). \qquad (25)$$

При $k << n$ маємо

$$P_k = 1 - e^{-\frac{k(k-1)}{2n}} \qquad (26)$$

Тому (26) дає можливість також обчислити *k* необхідних елементів послідовності (22), щоб отримати колізію з необхідною ймовірністю.

Тепер оцінимо складність обчислення послідовності довжиною *k* для кривих у формі Вейерштрасса та у формі Едвардса.

Побудування послідовності довжиною *k* потребує *k -1* викликів функції F(Y). Позначимо складність виконання однієї операції додавання точок ЕК як I_+, а складність подвоєння точки на ЕК як I_*. Так як послідовність $w_0 \in W, w_{k+1} = F(w_k)$ можна вважати випадковою, маємо

$$I_{sum} = (k-1)\left(\frac{1}{3} I_* + \frac{2}{3} I_+\right). \qquad (27)$$

Підставимо із табл. 1 значення для різних кривих та оцінимо складність генерації послідовності довжиною *k* для ЕК у формі Едвардса та канонічних кривих.

Для канонічних кривих:

$$I_{sum.canonocal} = (k-1)\left(\frac{1}{3} \cdot 10.33M + \frac{2}{3} \cdot 13.33M\right) = (k-1) \cdot 12.33M$$

Для кривих Едвардса:

$$I_{sum.edwards} = (k-1)\left(\frac{1}{3} \cdot 5,67M + \frac{2}{3} \cdot 11.57M\right) = (k-1) \cdot 9.603M$$

Тобто, побудування послідовності точок для криптоаналізу методом ро-Полларда є більш ефективним для кривих Едвардса, ніж для кривих в канонічному вигляді. Слід зазначити, що оскільки в (22) було прийнято, що множення точок виконується тільки в третині випадків, тобто коли $n/3 \le Y(x) \le 2n/3$, а для випадку додавання точок різниця в складності операцій не настільки суттєва, як у випадку множення, то приріст швидкості крипто аналізу складає приблизно 22 %, проте для інших алгоритмів, які більш активно використовують множення точок ЕК, він може бути суттєво вищим. Таким чином, дійне збільшення швидкодії також дозволяє збільшити швидкодію, а значить зменшити складність криптоаналізу, тобто розв'язання дискретного порівняння в групі точок кривих Едвардса. Вказаний факт необхідно враховувати при переході від використання точок еліптичних кривих до кривих Едвардса.

Висновки та рекомендації

1. Еліптичні криві Едвардса, як новий вид представлення еліптичних кривих, має суттєві переваги, основними серед яких є швидкодія. В цьому випадку можливість прискорити виконання підпису у 1,5 рази є дуже привабливою, проте, на наш погляд, потребує подальшого дослідження в частині стійкості проти атак як з використанням класичних обчислювальних пристроїв, так і після появи квантового комп'ютера.

2. Проведений аналіз також дозволив дійти висновку, що для кривих Едвардса можливе прискорення порівняно с канонічними кривими виконання алгоритмів крипто аналізу, яке залежить від кількості використання у конкретному алгоритмі операцій множення точок ЕК. Також слід враховувати, що застосування механізмів, які базуються на ЕК, можливе тільки обмежено, адже після появи квантового комп'ютера їх використання вже не забезпечить необхідного рівня стійкості.

3. Проблемним також є питання впровадження реалізацій, що використовують криві Едвардса в існуючу інфраструктуру відкритих ключів. На наш погляд, доцільним є розгляд питання щодо вдосконалення моделей, методів та засобів побудови крипто стійких загальносистемних параметрів для криптографічних перетворень в групах точок еліптичних кривих в напрямку підвищення їх стійкості, а також обґрунтування можливостей використання таких параметрів у постквантовий період. Вказане дозволить, насамперед, зосередитися на стандартизації нових криптографічних механізмів постквантового періоду.

Список літератури:

1. Bernstein D.J., Lange T. Faster Addition and Doubling on Elliptic Curves // Advancesin Cryptology—ASIACRYPT'2007 (Proc. 13th Int. Conf. On the Theory and Applicationof Cryptology and Information Security. Kuching, Malaysia. December 2–6, 2007). Lect. Notes Comp. Sci. V. 4833. Berlin : Springer, 2007. PP. 29–50.

2. Edwards H.M. A normal form for elliptic curves // Bulletin of the American Mathematical Society, Volume 44, Number 3, July 2007, PP. 393-422

3. Hisil Huseyin, Koon-Ho Wong Kenneth, Carter Gary, Dawson Ed. Twisted Edwards Curves Revisited // ASIACRYPT. 5350. New York: Springer, 2008. PP. 326-343

4. Балагура Д.С. Методы оценки сложности криптоанализа для криптографических приложений в группе точек эллиптической кривой, учитывающие вероятность коллизий / Д.С. Балагура, Ю.И. Горбенко // Радиотехника. 2005. Вып. 142. С. 205 – 214.

5. Бессалов А.В. Эллиптические кривые в форме Эдвардса и криптография: монография. Киев : ІВЦ «Видавництво «Політехника»», 2017. 272с.

6. Горбенко І. Д. Прикладна криптологія : підручник / І. Д. Горбенко, Ю. І. Горбенко ; вид. 2-ге. Харків : Форт, 2013. 878 с.

7. Горбенко Ю.І., Єсіна М.В., Кулібаба В.А. Сутність та умови здійснення атаки на зв'язаних ключах відносно ектронних підписів IBS-1 та IBS-2 ДСТУ ISO/IEC 14888-3 // Системи обробки інформації. 2016. № 7. С. 113-118. (http://www.hups.mil.gov.ua/periodic-app/article/16934)

Харківський національний
університет імені В.Н. Каразіна;

Надійшла до редколегії 06.09.2019

UDC 004 056 55 **DOI:10.30837/rt.2019.3.198.16**

A. BESSALOV, Dr. Sc. (Engineering), L. KOVALCHUK, Dr. Sc. (Engineering),
N. KUCHYNSKA, Ph.D. (Engineering), O. TELIZHENKO

SECURITY OF MODIFIED DIGITAL PUBLIC-KEY SIGNATURE EDDSA

Introduction

The Ukrainian National Standard for Digital Signature DSTU 4145-2002 has been in use about 17 years.

During this time, significant changes have occurred in the field of information technology:

- new, more powerful cryptanalysis techniques have emerged and with the growing computing capabilities, are forcing the world to look for ways to increase the resilience of existing cryptosystems;
- in the last few years, practically all cryptographic algorithms used at this level are analyzed and evaluated in terms of their security with respect to the modern and perspective post-quantum methods of cryptanalysis, which leads to revision of these algorithms (usually, with increasing size of the parameters of such algorithms);
- DSTU 4145-2002 standard uses and refers to outdated Soviet and Russian block encryption and hash functions standards, while Ukraine's 2014 adopted DSTU 7564 and DSTU 7624 National Standards, which have significantly higher performance and cryptographic security;
- new algebraic objects have appeared the use of which has many advantages (both in performance and cryptographic security) in building different cryptosystems;
- a new cryptographic algorithms, including auxiliary ones, were offered which have numerous advantages over older ones and are gradually displace them from use.

In addition, a significant drawback of DSTU 4145-2002 is that it (the only one all over the world) recommends the use of only elliptic curves over the finite field of characteristic 2, which makes the signature of this standard more than 20% slower than any other states, including Russia and Belarus.

All these changes directly affect the implementation of the current National Standard for Digital Signature DSTU 4145-2002 and indicate the need for its modernization.

An important task in choosing asymmetric crypto-algorithms is selection of parameters size and its justification. Today an international organizations, such as NIST, ETSI, and ENISA, have raised appropriate parameter requirements that can be used in the transitional to post-quantum and post-quantum period. [1], [2].

Due to the need to revise and update national digital signature standard DSTU 4145-2002 [3], the authors considered several digital signature constructions. Among the requirements to modern public-key signatures it is worth to highlight at least 128-bit security, fast signing and fast signature verification, fast keys generation, foolproof session keys, collision resistance, secure software implementation, etc. There are a lot of obvious variants in classic and elliptic signature systems, ElGamal, Schnorr`s, ECDSA, etc, which can be used in transitional to post quantum period.

This paper introduces one of possible modifications for signature schemes based on The Edwards-curve Digital Signature Algorithm (EdDSA) proposed in [4] and specified in IETF RFC 8032 [5], which is a variant of Schnorr's signature system with (possibly twisted) Edwards curves. The main advantages of the modification proposed in this work are:

1) secure even in case of generator faults [6];
2) signature performance doesn't depend on message length;
3) security against related-key attacks.

Related work

The equation of the elliptic curve in the form, which later took the name "Edwards form", was suggested in paper [11].The isomorphism (under certain conditions) between the curves in the Weierstrass form and in the Edwards form was proved. Howewer the curves, proposed in [11] were weak from the cryptographic point of view. And paper [11] was quickly followed by paper [12], where the Edwards curves were modified by the introduction of the certain parameter.

Hereafter, for simplification, we will consider curve E, assigned over finite field F_p, for an odd prime number p and $a,d \in F_p^* \; a \neq d, d \neq 1$:

$$E: ax^2 + y^2 = 1 + dx^2 y^2, \tag{1}$$

The main differences (almost all of which are advantages) of the Edwards curve compared with the Weierstrass curves are the following.

1. Universality of the addition law. Indeed, the operations of different points addition and doubling a point are assigned by he same formulas:

$$(x_1, y_1) + (x_2, y_2) = \left(\frac{x_1 y_2 + x_2 y_1}{1 + d x_1 x_2 y_1 y_2}, \frac{x_1 y_2 - a x_2 y_1}{1 - d x_1 x_2 y_1 y_2} \right) \tag{2}$$

2. The absence of "the point on infinity". Thus, the neutral element is a usual point of the Edwards curve with coordinates $O = (0,1)$, which obviously fulfill equation (1).

3. Group E_p of points of elliptic curve (1) with Legendre symbol $\left(\frac{ad}{p} \right) = -1$ is always cyclic.

4. The order of the group E_p is always divided by 4. The property of the Edwards curve can be considered an insignificant disadvantage due to the fact that its subgroup of the large prime order, on the basis of which cryptosystems are constructed, has at least 4 times less points than E_p, in other words at least three-quarters of the points of the group are "extra".

5. The high performance of points addition. This property is one of the most important advantages of the Edwads curve. Thus, approximately 1.5 bit operations less required for two (different) points addition of the Edwads curve than for the same operation for the Weierstrass curve. Meanwhile for the doubling of the points the bit operation number is even less.

6. Uniformity of the addition law. The formulas for the addition and doubling of points are the same for the Edwards curve. This increases security of cryptosystems on the Edwards curves to timing and capacitive attacks, aimed to determine the number of such operations.

In this part we briefly specify EdDSA signature system according to [4], [5], but in more formal way. As the authors of algorithm pointed, the advantages with EdDSA are as follows:

- EdDSA provides high performance on a variety of platforms and the use of a unique random number for each signature is not required;
- specified in RFC 8032 EdDSA uses relatively small (in comparison with postquantum algorithms) public keys (32 or 57 bytes) and signatures (64 or 114 bytes) for Ed25519, providing approximately 128 bits of security (uses Edwards version of Curve25519) and Ed448, which provides approximately 224 bits of security, respectively;
- the formulas are "complete", i.e., they are valid for all points on the curve, with no exceptions. This obviates the need for EdDSA to perform expensive point validation on untrusted public values;
- EdDSA provides collision resilience, meaning that hash-function collisions do not break this system (only holds for PureEdDSA) and it is more resilient to side-channel attacks.

EdDSA needs to be instantiated with certain parameters. The scheme has next parameters:

- an odd prime number p;
- an integer b with $2^{b-1} > p$,
- a $(b-1)$-bit encoding of elements of finite field F_p,
- a cryptographic hash function H producing $2b$-bit output;
- a non-square element d of F_p and a non-zero square element a of F_p (the usual recommendation for best performance is

$$a = \begin{cases} -1, \text{if } p \equiv 1 \bmod 4, \\ 1, \text{if } p \equiv 3 \bmod 4; \end{cases} \tag{3}$$

- a prime n between 2^{b-4} and 2^{b-3} under special condition and the point $P \neq (0,1)$.

Note, that the set (1) defines complete Edwards curve (according to classification [7]), because $a \in Q_p$, according to (3), where Q_p is the set of quadratic residues modulo p. Due to the condition $d \notin \{0,-1\}$ and $d \notin Q_p$, the set (3) forms a group of points with affine coordinates with neutral element $O = (0,1)$ under the addition law (2) [7], [8].

There is also extra constraint for the base point P of elliptic curve E, with order n of the point P, i.e. $nP = 0$, where n is a large prime. So the number of points on the curve is $|E| = 2^c n$ and a cofactor is 2^c with integer c, $c \in \{2,3\}$ (according to choice of elliptic curve).

An EdDSA secret key is a b-bit string k. The hash $H(k) = (h_0, h_1, ..., h_{2b-1})$ determines an integer $e = 2^{b-2} + \sum_{c \le i < (b-c)} 2^i h_i$. The knowledge of e is sufficient for producing valid signatures, which justifies considering e as the signing key. The public key $Q = eP$ is a point on the curve (1).

An element $\overline{(x,y)} \in E$ is encoded as a b-bit string (x,y), namely y is encoded by $(b-1)$-bit string and concatenated with one bit that is 1 if x is negative and 0 if x is not negative [4].

Note that the encoding of elements of the finite field F_p is defined specifically as, x is "negative" if the (b-1)-bit encoding of x is lexicographically larger than the (b-1)-bit encoding of $-x$ (in little-endian form).

The signature of a message M under this secret key k is defined as follows.

Signature algorithm:

1) compute $r = H(h_b, ..., h_{2b-1}, M) \in \{0, ..., 2^{2b-1}\}$;
2) compute $R = rP$;
3) compute $h = H(\overline{R}, \overline{Q}, M)$ and convert it into integer;
4) compute $s = (r + eh) \bmod n$.

The signature of M is $DS = (\overline{R}, s)$

Verification algorithm:

1) compute $h' = H(\overline{R}, \overline{Q}, M)$ and convert into integer;
2) check $sP = R + h'Q$.

The verifier rejects the alleged signature if the parsing fails or if the group equation does not hold.

EdDSA is based on digital signature scheme that was first designed by Schnorr [9]. A main demand when using this kind of signatures is that r has to be chosen unpredictably [4]. Indeed, if r can be guessed correctly for an existing signature, then the signing key a_k can be simply computed as $e = (s - r)h^{-1} \bmod n$ using the extended Euclidean algorithm.

Legitimate users choose $Q = eP$, where e is a random secret; the derivation of e from $H(k)$ ensures adequate randomness. These users have negligible chance of generating any particular multiple of P targeted by the attacker. The chance of the attacker randomly guessing e is much smaller than the chance of the attacker computing e by known discrete-logarithm algorithms; standard elliptic-curve security criteria are designed so that the latter algorithms have negligible chance of succeeding in any reasonable amount of time.

Furthermore, if the same nonce value k (with unknown hash $H(k) = (h_0, h_1, ..., h_{2b-1})$) has been used for generating signatures of different messages M_1, M_2, and $h_1 = H(\overline{R}, \overline{Q}, M_1)$, $h_2 = H(\overline{R}, \overline{Q}, M_2)$ the signing key e can be tried to recover by eavesdropper as

$$e = ((s_1 - s_2) - (r_2 - r_1))(h_1 - h_2)^{-1} \bmod n,$$

where $r_1 = H(h_b, ..., h_{2b-1}, M_1)$ and $r_2 = H(h_b, ..., h_{2b-1}, M_2)$. But values r_1 and r_2 are still different and unknown. So the secret key cannot be found even in case of generator`s faults, when it produces the same nonce k for two different messages. That is one of the essential differences of EdDSA from ECDSA [10].

Modification of EdDSA

This work considers signature construction, with longtime signature key. In our notation private key is $e \in F_p$, $e \neq \pm 1$ and public key $Q = eP$ is a point on the elliptic curve E over F_p. Note that not only the curve (1), but any suitable Edwards elliptic curve with $|E| = 2^c n$ and small cofactor 2^c can be used in this construction, i.e.

$$E = \left\{(x, y) \in F_p \times F_p : x^2 + ay^2 = 1 + dx^2 y^2\right\}. \quad (4)$$

Under the condition $(ad) \notin Q_p$ the set (4) forms a group of points with affine coordinates with neutral element $O = (1,0)$ according the addition law [7]:

$$(x_1, y_1) + (x_2, y_2) = \left(\frac{x_1 x_2 - a y_2 y_1}{1 - d x_1 x_2 y_1 y_2}, \frac{x_1 y_2 + x_2 y_1}{1 + d x_1 x_2 y_1 y_2}\right). \quad (5)$$

The modified scheme of EdDSA make use of next parameters:

- an odd prime number p, defining the underlying field;
- an integer b with $2^{b-1} > p$ and a $(b-1)$-bit encoding of elements of finite field F_p,
- a cryptographic hash function H producing bit strings of the length not less than b.

The signature and verification algorithm for a message M under this secret key e defined as follows.

Signature algorithm:

1) generate random k : $1 < k < n$;
2) compute $r = H(k \,||\, H(M))$;
3) compute point $R = rP$;
4) encode $\overline{R}, \overline{Q}$ and compute $h = H(\overline{R}, \overline{Q}, H(M))$;

5) compute $s = (r + eh) \bmod n$.

The signature of M is $DS = (\overline{R}, s)$.

Verification algorithm:

1) compute $h' = H(\overline{R}, \overline{Q}, H(M))$ and convert into integer;

2) check $sP = R + h'Q$.

Correctness of proposed signature can be proved with next equality

$$R + H(\overline{R}, \overline{Q}, H(M))Q = rP + H(\overline{R}, \overline{Q}, H(M))dP = (r + dH(\overline{R}, \overline{Q}, H(M)))P = sP.$$

Reducing the secret key search to solution of DLP for Modification of EdDSA

In this section, we show that the task of the secret key recovery in the proposed digital signature algorithm is polynomially reduced to the DLP problem./

Theorem: the problem of obtaining secret key from given pair message M and its signature DS in Modification of EdDSA is not easier (not more efficient) than solution of discrete logarithm problem.

Proof: let we have an Oracle O, which for given message M and its signature $DS = (\overline{R}, s)$ returns secret key e. Let we have some point $R = rP$ for some unknown r for elliptic curve base point P, with $ordP = n$, i.e. $nP = 0$. Then we can construct the next algorithm, which finds r in polynomial time, using an Oracle O.

Algorithm input: P, R.

Algorithm output: r (where $R = rP$).

Algorithm.

1) generate random $s : 1 < s < n$ and $h : 1 < h < n$;

2) compute $h^{-1} \bmod n$;

3) compute $Q = h^{-1}sP - h^{-1}P$;

4) query the oracle O and gets its answer $O(P, R, s, h, Q) = a_k$;

5) compute $r = s - a_k h$.

It should be marked, that according to the Algorithm it follows that $Q = a_k P$, where $s = r + eh$. Then $sP = rP - hQ$ and $Q = h^{-1}(s - r)P$.◄

Conclusion

The signature algorithm, considered in this work, is promising and provable secure against leakage of secret key. The main advantage over the original signature EdDSA is to reduce the time of its work, as well as that the time of work does not depend on the length of the message.

Besides it is still secure even in case of PRNG faults. In further analysis, it is desirable to show that the algorithm is resistant to a keyless subscription.

References:

1. ETSI GR QSC 001 V.1.1.1 (2016-07). Quntum-Safe Cryptography (QSC); Quantum-safe algorithmic framework. Access mode: https://portal.etsi.org/webapp/workProgram/Report_WorkItem.asp?wki_id=46690 30.10.2016.
2. Quantum Safe Cryptography and Security. An introduction, benefits, enablers and challenges. ETSI White Paper No. 8, 2015. Access mode: http://www.etsi.org/images/files/ETSIWhitePapers/QuantumSafeWhitepaper.pdf 30.10.2016.
3. DSTU 4145-2002. Information Technology. Cryptographic protection of information. Digital signature based on elliptic curves.
4. D. J. Bernstein, N. Duif, T. Lange, P. Schwabe, and B.-Y. Yang. High-speed high-security signatures // Proc. of the 13th International Workshop on Cryptographic Hardware and Embedded Systems (CHES'11), Nara, Japan, ser. Lecture Notes in Computer Science, vol. 6917. Springer-Verlag, September 2011, pp.124–142.

5. S. Josefsson, I. LiusvaaraRFC 8032 Edwards-Curve Digital Signature Algorithm (EdDSA). January 2017 DOI: 10.17487/RFC8032

6. Ambrose, Christopher & Bos, Joppe & Fay, Björn & Joye, Marc & Lochter, Manfred & Murray, Bruce. (2018). Differential Attacks on Deterministic Signatures.

7. Bessalov A.V. (2017). Ellipticheskie krivyie v forme Edvardsa i kriptografiia: monografiya. Kyiv : KPI im. Igoria Sikorskogo ; Politekhnik». 272.

8. Daniel J. Bernstein, Peter Birkner, Marc Joye, Tanja Lange, Christiane Peters. Twisted Edwards curves // Africacrypt 2008, 389–405. http://eprint.iacr.org/2008/013

9. Claus P. Schnorr. Efficient Identification and Signatures forSmart Cards // Advances in Cryptology. CRYPTO ’89. NewYork: Springer, 1990, pp. 239–252.

10. Hartl Alexander & Annessi Robert & Zseby Tanja. (2017). A Subliminal Channel in EdDSA: Information Leakage with High-Speed Signatures. 67-78.

11. Edwards H.M. (2007). A normal form for elliptic curves. Bulletin of the American Mathematical Society, V. 44, 393-422.

12. Bernstein D.J., Lange T. (2007) Faster Addition and Doubling on Elliptic Curves // Kurosawa K. (eds) Advances in Cryptology – ASIACRYPT 2007. Lecture Notes in Computer Science, vol 4833. Springer, Berlin, Heidelberg.

Institute of Physics and Technology NTUU
“Igor Sikorsky Kyiv polytechnic institute”

Received 11.09.2019

UDC 004 056 55 **DOI:10.30837/rt.2019.3.198.17**

D. TELEVNYI

THE KUPYNA HASH FUNCTION APPLICATION TO SPHINCS+ SIGNATURES

Introduction

Digital signatures (DSAs) are crucial elements in any system that requires data protection. The most used signatures are based on asymmetric pairs.

In recent years, there has been a substantial amount of research on quantum computers – machines that exploit quantum mechanical phenomena to solve mathematical problems that are difficult or intractable for conventional computers. If large-scale quantum computers are ever built, they will be able to break many of the public-key cryptosystems currently in use.

Hash-based signature schemes were developed as one-time signature schemes in the late 1970s by Lamport and extended to more signatures by Merkle.

As for the 2nd round there are 9 Digital signature candidates. SPHINCS+ (former SPHINCS) is in the list. The algorithm can be briefly described as a stateless hash-based signature scheme. It uses many components from XMSS but works with larger keys and signature to eliminate state.

The scheme can be used with different hash functions.

The main goal of this paper is to analyze the application of the national standard hash function the scheme of the NIST submission candidate SPHINCS+.

The NIST candidate SPHINCS+

In 40 years since Lamport's scheme, many ideas were introduced to improve performance, practicality and theoretical aspects of hash-based signatures. As a result, XMSS were introduced that's in the last stage of being standardized by the CFRG as the first post-quantum signature scheme. The only downside of XMSS is being stateful, which makes it not fit the standard definition of signature schemes.

In 2017 NIST (National institute of standards and technology) announced the 1st round candidate submissions for both post quantum Public-key Encryption and Key-establishment Algorithms and Digital Signatures.

As for the 2nd round there are 9 Digital signature candidates. SPHINCS+ (former SPHINCS) is in the list. The algorithm can be briefly described as a stateless hash-based signature scheme. It uses many components from XMSS but works with larger keys and signature to eliminate state.

SPHINCS [1] was designed by Bernstein, Hopwood, Hülsing, Lange, Niederhagen, Papachristodoulou, Schneider, Schwabe, and Wilcox-O'Hearn as a stateless hash-based signature scheme and was the first signature scheme to propose parameters to resist quantum cryptanalysis.

At a high level, SPHINCS works the following way. The basic idea is to authenticate a huge number of few-time signature (FTS) key pairs using a so-called hyper-tree. FTS schemes are signature schemes that allow a keypair to produce a small number of signatures, e.g., in the order of ten for our parameter sets. For each new message, a (pseudo)random FTS key pair is chosen to sign the message. The signature consists then of the FTS signature and the authentication information for that FTS key pair. The authentication information is a hyper-tree signature, i.e. a signature using a certification tree of Merkle tree signatures.

SPHINCS uses several parameters and functions [1]. The main security parameter is $n \in N$. The functions include two short-input cryptographic hash functions $F:\{0,1\}^n \rightarrow \{0,1\}^n$ *and* $H:\{0,1\}^{2n} \rightarrow \{0,1\}^n$; one arbitrary-input randomized hash function $H:\{0,1\}^n \times \{0,1\}^* \rightarrow \{0,1\}^m$, for $m = poly(n)$; a family of pseudo-random generators $G_\lambda:\{0,1\}^n \rightarrow \{0,1\}^{\lambda n}$ for different values of λ; an ensemble of pseudo-random function families $F_\lambda:\{0,1\}^\lambda \times \{0,1\}^n \rightarrow \{0,1\}^n$; and a pseudo-random function family $F:\{0,1\}^* \times \{0,1\}^n \rightarrow \{0,1\}^{2n}$ that supports arbitrary input lengths.

The security parameter n is also the output length of all cryptographic function families besides $H_{msg.}$

The Winternitz parameter w determines the number and length of the hash chains per WOTS+ instance. A greater value for w linearly increases the length of the hash chains.

The algorithm uses a hyper-tree with a total height of h and the hyper-tree consists of d layers of trees, each having height h/d. The height of the hyper-tree h determines the number of FORS instances.

The tree-based few-time signature scheme HORST has a space-time tradeoff which is controlled by two parameters $k \in N$ and $t = 2\tau$ with $\tau \in N$ and $k\tau = m$.

More detailed description of WOTS+ and HORST structures can be found in Specification paper [5]. The developers also propose SPHINCS-256 as a tested and verified set of parameters for the digital signature of the security strength *128* bit.

The hyper-tree as much as the whole signature can be visualized as it's shown in figure 1. It contains d trees (each consisting of a binary hash tree that authenticates the root nodes of $2^{h/d}$ L-Trees which in turn each have the public key no des of one WOTS+ key pair as leaves). Each tree authenticates the tree below using a WOTS+ signature σ_W, i. The only exception is $Tree_0$ which authenticates a HORST public key using a WOTS+ signature. Finally, the HORST key pair is used to sign the message.

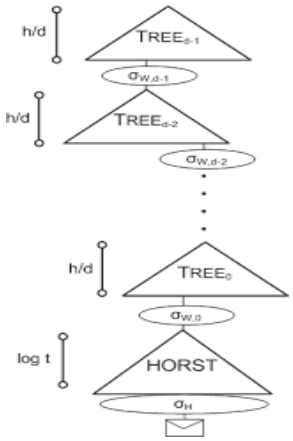


Fig. 1. SPHINCS hyper-tree

The general structure of SPHINCS can be described as a superposition of trees sets.

– *The SPHINCS hyper-tree* of height h. The root of tree is a part of the public key. The leaves are the *HORST* instances, the tree is divided in d layers containing so-called Merkle-trees.

– *Merkle-trees* of height h/d. The leaves are roots of the WOTS+ public key compression trees.

– *The WOTS public key compression trees* are L-trees of height $log_2 l$, for l – leaves amount. The leaves of this tree are components of a WOTS public key. The associated WOTS instance signs a tree root at the next layer.

– *The HORST public key compression trees* are Merkle trees of height $\tau = log_2 t$ where t is the number of public key elements in the HORST instances.

There were some changes made since the 2nd round submissions. SPHINCS+ introduced several details changes: multi-target attack protection [3], tree-less WOTS+ Public Key Compression (instead of tree-based compression), FORS (forest of random subsets) replaced HORST. [2].

The National Standard Hash Kupyna

In 2014 Ukraine released the new hash standard DSTU 7564:2014. It [dstu] defines the parameter set and modes of the Kupyna hash. As mentioned in the description the construct is based on the Even-Mansour scheme with Davies-Mayer compress function and inner permutation block from Kalyna (DSTU 7664:2014). The hash function supports several modes, defined as Kupyna-n. The standard [dstu] defines the following modes of Kupyna-256, 384, 512. [4]

The Kupyna hash is resistant to the second-preimage search attack. [4] The collision attacks described in «Analysis of the Kupyna-256 Hash Function» paper [5] showed that collision attacks are possible on the round-reduced hash up to 5 rounds. The time complexity of the attacks on round-reduced hash is shown in Table 1.

Table 1

Complexity of collision attack
on the reduced hash function

rounds	Time complexity
4	2^{67}
5	2^{120}

The collision attack on the compression function up to 7 rounds included semi-free-start collisions and was based on the rebound attack on Grøstl using SuperBox matching [5 – 7].

The time complexity results are shown in Table 2.

Table 2

Time complexity of the collision attack
on the compression function

rounds	Time complexity
6	2^{70}
7	2^{125}

In the SPHINCS design description [1] authors announced that the hash functions must be collision-resilient to be adopted in the signature scheme. In the updated version SPHINCS+ [2] hash functions must also be second-preimage search resistant due to possible multi-target attack [3].

The Kupyna hash is collision-resilient and preimage resistant if it works in modes as defined in the national standard [4]. Thus it can be applied to the SPHINCS+ signature scheme.

Kupyna-n application and benchmarking

There are tables of benchmark results given in [5]. The authors used the following Hash function families: SHA-256, SHAKE-256, Haraka-256.

The reference code can be found in the project site. [8] Each implementation case contains the signature scheme implementation alongside with hash function implementation. Authors released two versions: with suffix “s” – generates rather small signature and keys and “f” – for more quicker signature.

There are test cases applied to each reference implementation. The cases includes testing keypair and signature generation for WOTS+, FORS and the whole scheme. The PQCGenKAT_sign generates signatures for different message lengths.

The implementation is defined in a separate file and linked to signature within a source file. The header file “*hash.h*” contains the following interface listed below.

```
void prf_addr(unsigned char *out, ...);

void gen_message_random(unsigned char *R, const unsigned char *sk_seed..);

void hash_message(unsigned char *digest, uint64_t *tree, uint32_t *leaf_idx,
                  const unsigned char *R..);

void thash(unsigned char *out, const unsigned char *in, unsigned int
inblock..);
```

The src files contains the implementation of given functions with calls to the specific hash function. The Kupyna hash was implemented by the national standard description specs. [1]

The benchmark was held for signature schemes using 256 bit hash functions. The list included SHA-256, Haraka-256, SHAKE-256.

The Kupyna-256 function was used for implementation of the `hash_message()` function that computes the message hash using randomness, the public key, and the message. `gen_message_random` computes the message-dependent randomness R, using a secret seed as a key for HMAC, and an optional randomization value prefixed to the message. HMAC contains Kupyna-256 as a hash.

The signature testing runs on the following specs: *Core i3 6006U, 4GB RAM, Ubuntu 18.04, kernel 4.19.72 SMP x86_64.* The SHA-256 function was loaded from *openssl* shared lib from *libssl-dev* package. The compiler is *gcc 7*, CFLAGS = -Wall-Wextra-Wpedantic -O3

The test results for both “small signature length” and “fast signing” is listed in the Table 3 below.

Table 3

Run-time results

Signature name	Time (keygen + sign + ver)	Signature length
sphincs-shake256-256s	7.3851s	~ 29 kBytes
sphincs-shake256-256f	0.86s	~ 49 kBytes
sphincs-sha256-256s	6.83s	~ 29 kbytes
sphincs-sha256-256f	0.84s	~ 49 kBytes
sphincs-haraka256-256s	13.21s	~29 kBytes
sphincs-haraka256-256f	1.49s	~ 49kBytes
sphincs-sha-kup256-256s	6.82s	~29 kBytes
sphincs-sha-kup256-256f	0.81s	~ 49kBytes

The time measuring is performed by `clock_gettime()` function calls with CLOCK_PROCESS_CPUTIME_ID resolution. This function is common to Gnu libc and utilizes Intel processors TCS.

The measurements include key pair generation, message signing, verifying the signature with public key.

Conclusions

There has been made some changes in SPHINCS since the first submission to NIST as a Digital Signature candidate. The new signature called SPHINCS+ now requires hash function families to be second-preimage search resistant as well due to possible multi-targeting attacks. The Wirnetnitz signatures for public key compressions is now tree-less which became possible with introducing inner tweakable hash functions. The HORST scheme was also replaced by FORS (forest of random subsets) concept.

The Kupyna hash is collision-resilient and preimage resistant if it works in modes as defined in the national standard [4]. So it can be applied to the SPHINCS+ signature scheme.

The SHINCS+ signature scheme declared different hash families could be used for message signing. For practical part Kupyna-256 function was applied for randomness generation as a part of HMAC and message hashing. Authentication path calculation utilizes the SHA-256 hash.

The signature scheme was presented for 2 cases. The first case is generating rather small signatures (approximately 29 kbytes) for much longer time (a few seconds). This case is marked with -s suffix. The fast signing (marked with suffix -f) produces a longer signature (about 49 kBytes). The signature in this case contains all necessary data (seeds, full authentication path) to make verification more flawless.

The results shows the signatures have better performance when applying more lightweight hashes (SHA-2 /256, Kupyna -256) rather than using Keccak sponge (SHAKE). Haraka is a fast postquantum hash built to work on processors with AES-NI instructions. When used as a reference implementation (un-optimized) it shows itself less perfomant. In case when optimization is applied by vectorizing calculations and utilizing instruction sets the perfomance can be drastically increased.

The National standard [dstu] hash function shows good performance when used with other hashes for authentication in the signature scheme.

References:

1. Daniel J. Bernstein, Daira Hopwood, Andreas Hülsing, Tanja Lange, Ruben Niederha-gen, Louiza Papachristodoulou, Michael Schneider, Peter Schwabe, and Zooko Wilcox-O'Hearn. SPHINCS: Practical Stateless Hash-Based Signatures. In Elisabeth Oswaldand Marc Fischlin, editors, EUROCRYPT 2015, volume 9056 of LNCS, pages 368–397.Springer Berlin Heidelberg, 2015.

2. Jean-Philippe Aumasson, Daniel J. Bernstein, Christoph Dobraunig, Maria Eichlseder, Scott Fluhrer, Stefan Lukas Gazdag, Andreas Hülsing, Panos Kampanakis, Stefan Kölbl, Tanja Lange, Martin M. Lauridsen, Florian Mendel, Ruben Niederhagen, Christian Rechberger, Joost Rijneveld, Peter Schwabe. SPHINCS$^+$ – Submission to the 2nd round of the NIST post-quantum project. Specification document (part of the submission package). 2019-03-14

3. Andreas Hülsing, Joost Rijneveld, and Fang Song. Mitigating multi-target attacks inhash-based signatures. In Chen-Mou Cheng, Kai-Min Chung, Giuseppe Persiano, andBo-Yin Yang, editors,PKC 2016, volume 9614 of LNCS, pages 387–416. Springer BerlinHeidelberg, Berlin, Heidelberg, 2016.

4. Roman Oliynykov, Ivan Gorbenko, Oleksandr Kazymyrov, Victor Ruzhentsev, Oleksandr Kuznetsov, Yurii Gorbenko, Artem Boiko, Oleksandr Dyrda, Viktor Dolgov, Andrii Pushkaryov. A New Standard of Ukraine: The Kupyna Hash Function. Cryptology ePrint Archive. Report 2015/885, 2015. https://eprint.iacr.org/2015/885.pdf

5. Christoph Dobraunig, Maria Eichlseder, and Florian Mendel. Analysis of the Kupyna-256 Hash Function, Graz University of Technology, Austria, Cryptology ePrint Archive. Report 2015/956, 2015. https://eprint.iacr.org/2015/956.pdf

6. Mendel F., Rechberger C., Schl affer M., Thomsen S.S.: Rebound attacks on the reduced Grøstl hash function. In: Pieprzyk, J. (ed.) Topics in Cryptology – CT-RSA 2010. LNCS. vol. 5985. P. 350–365. Springer (2010)

7. Jean J., Naya-Plasencia M., Peyrin T. Improved rebound attack on the finalist Grøstl. In: Canteaut, A. (ed.) Fast Software Encryption – FSE 2012. LNCS. vol. 7549. P. 110–126. Springer (2012)

8. Peter Schwabe (September 23, 2019) SPHINCS+ Stateless hash-based signatures. Software Reference Implementation. Retrieved from https://sphincs.org/software.html

9. Klintsevich K., Okeya, Vuillaume C., Buchmann J., Dahmen E. Merkle signatures with virtually unlimited signature capacity. 5th International Conference on Applied Cryptography and Network Security. ACNS07, 2007.

Kharkiv National University of Radio Electronics *Received 11.09.2019*

РЕФЕРАТЫ РЕФЕРАТИ ABSTRACTS

ПЕРСПЕКТИВНЫЕ КРИПТОГРАФИЧЕСКИЕ ПРЕОБРАЗОВАНИЯ И ИХ ПРИМЕНЕНИЕ

ПЕРСПЕКТИВНІ КРИПТОГРАФІЧНІ ПЕРЕТВОРЕННЯ ТА ЇХ ЗАСТОСУВАННЯ

PERSPECTIVE CRYPTOGRAPHIC TRANSFORMATIONS AND THEIR APPLICATION

УДК 004.056.55

Алгоритмы асимметричного шифрования и инкапсуляции ключей постквантового периода 5-7 уровней стойкости и их применение / *И.Д. Горбенко, О.Г. Качко, А.Н. Олексейчук, А.А. Кузнецов, Ю.И. Горбенко, В.В. Онопрієнко, М.В. Есина, С.А. Кандий* // Радиотехника : Всеукр. межвед. науч.-техн. сб. 2019. Вып. 198. С. 5 – 18.

Подаются и рассматриваются построенные алгоритмы асимметричного шифрования и инкапсуляции ключей в кольцах полиномов (алгебраических решетках), анализируется сущность криптографических преобразований асимметричного шифрования и протоколов инкапсуляции ключей, которые применяемых. Рассматриваются механизмы шифрования и инкапсуляции с различными наборами параметров, определяющих устойчивость.

Ключевые слова: асимметричный шифр; инкапсуляция ключей; постквантовый период; уровни стойкости.

Табл. 6. Библиогр.: 17 назв.

УДК 004.056.55

Алгоритми асиметричного шифрування та інкапсуляції ключів постквантового періоду 5-7 рівнів стійкості та їх застосування / *І.Д. Горбенко, О. Г. Качко, А. М. Олексійчук, О.О.Кузнєцов, Ю.І. Горбенко, В.В.Онопрієнко, М. В. Єсіна, С. О. Кандій* // Радіотехніка : Всеукр. міжвід. наук.-техн. зб. 2019. Вип. 198. С. 5 – 18.

Подаються та розглядаються побудовані алгоритми асиметричного шифрування та інкапсуляції ключів в кільцях поліномів (алгебраїчних решітках), аналізується сутність криптографічних перетворень асиметричного шифрування та протоколів інкапсуляції ключів, що застосовуються. Розглядаються механізми шифрування та інкапсуляції з різними наборами параметрів, що визначають стійкість.

Ключові слова: асиметричний шифр; інкапсуляція ключів; постквантовий період; рівні стійкості.

Табл. 6. Бібліогр.: 17 назв.

UDC 004.056.55

Algorithms of asymmetric encryption and encapsulation of keys of post-quantum period of 5 -7 stability stability levels and their applications / *I.D. Gorbenko, O.G. Kachko, O.M. Oleksijchuk, O.O. Kuznetsov, Yu.I. Gorbenko, V.V. Onoprienko, M.V. Yesina, S.O. Kandy* // Radiotekhnika : All-Ukr. Sci. Interdep. Mag. 2019. №198. P. 5 – 18.

The asymmetric encryption and keys encapsulation in polynomial rings (algebraic lattices) built algorithms are presented and considered, the essence of used asymmetric encryption transformations and key encapsulation protocols are analyzed. Encryption and encapsulation mechanisms with different sets of parameters that determine stability are considered.

Key words: asymmetric cipher; key encapsulation; post-quantum period; stability levels.

6 tab. 17 items.

УДК 004.056.5

Стеганоаналитический метод, эффективный в условиях малой пропускной способности скрытого канала связи / *И.И. Бобок, А.А. Кобозева* // Радиотехника : Всеукр. межвед. науч.-техн. сб. 2019. Вып. 198. С. 19 – 31.

Одним из основных стеганографических методов, используемых при организации скрытого канала связи, остается на сегодняшний день метод модификации наименьшего значащего бита (LSB-method). Возможной особенностью современного использования LSB-метода является малая пропускная способность организуемого с его помощью скрытого канала связи. В таких условиях подавляющее большинство существующих стеганоаналитических методов являются малоэффективными. В работе на основе теории возмущений и матричного анализа разработан новый стеганоаналитический метод выявления наличия вложенной методом модификации наименьшего значащего бита дополнительной информации в цифровое изображение, эффективный в условиях малой пропускной способности скрытого канала связи. Основой метода является анализ нормированной отделенности максимальных сингулярных чисел непересекающихся блоков матрицы изображения, полученных путем ее стандартного разбиения. Показано, что для цифровых изображений, хранимых в формате без потерь, при их пересохранении в формат с потерями с различными коэффициентами качества будет иметь место монотон-

ное возрастание количества блоков, для которых увеличивается нормированная отделенность максимального сингулярного числа блока, с уменьшением коэффициента качества, используемого при сжатии исходного изображения. Указанная монотонность будет нарушаться в случае, когда пересохранению с потерями подвергается изображение, которое первоначально хранилось в формате с потерями. Сделанный вывод является основой для разработанного стеганоаналитического метода и реализующего его алгоритма, являющегося полиномиальным степени 2. Предложенный алгоритм превосходит по эффективности существующие аналоги в условиях пропускной способности скрытого канала связи меньше 0,1 бит/пиксель, эффективен как для цветных, так и для монохромных изображений. Выводы подтверждаются приведенными результатами вычислительного эксперимента, в котором было задействовано более 5000 цифровых изображений.

Ключевые слова: стеганоаналитический метод; цифровое изображение; малая пропускная способность скрытого канала связи; сингулярные числа; отделенность сингулярного числа; метод модификации наименьшего значащего бита.

Табл. 6. Ил. 4. Библиогр.: 28 назв.

УДК 004.056.5

Стеганоаналітичний метод, ефективний в умовах малої пропускної спроможності прихованого каналу зв'язку / *І.І. Бобок, А.А. Кобозєва* // Радіотехніка : Всеукр. міжвід. наук.-техн. зб. 2019. Вип. 198. С. 19 – 31.

Одним з основних стеганографічних методів, що використовуються при організації прихованого каналу зв'язку, залишається на сьогоднішній день метод модифікації найменшого значущого біта (LSB-method). Можливою особливістю сучасного використання LSB-метода є мала пропускна спроможність прихованого каналу зв'язку, що організується з його допомогою. У таких умовах переважна більшість існуючих стеганоаналітичних методів є малоефективними. У роботі на основі теорії збурень і матричного аналізу розроблений новий стеганоаналітичний метод виявлення наявності вбудованої методом модифікації найменшого значущого біта додаткової інформації в цифрове зображення, ефективний в умовах малої пропускної спроможності прихованого каналу зв'язку. Основою методу є аналіз нормованої відокремленості максимальних сингулярних чисел непересічних блоків матриці зображення, отриманих шляхом її стандартної розбивки. Показано, що для цифрових зображень, збережених у форматі без втрат, при їхнім перезбереженні у формат із втратами з різними коефіцієнтами якості буде мати місце монотонне зростання кількості блоків, для яких збільшується нормована відокремленість максимального сингулярного числа блоку, зі зменшенням коефіцієнта якості, використовуваного при стиску вхідного зображення. Зазначена монотонність буде порушуватися у випадку, коли перезбереженню із втратами піддається зображення, яке спочатку зберігалося у форматі із втратами. Зроблений висновок є основою для розробленого стеганоаналітичного методу і алгоритму, що його реалізує, який є поліноміальним ступеня 2. Запропонований алгоритм перевищує по ефективності існуючі аналоги в умовах пропускної спроможності прихованого каналу зв'язку менше 0,1 біт/піксель, є ефективним як для кольорових, так і для монохромних зображень. Висновки підтверджуються наведеними результатами обчислювального експерименту, у якому було задіяне більш 5000 цифрових зображень.

Ключові слова: стеганоаналітичний метод; цифрове зображення; мала пропускна спроможність прихованого каналу зв'язку; сингулярні числа; відокремленість сингулярного числа; метод модифікації найменшого значущого біта.

Табл. 6. Іл. 4. Бібліогр.: 28 назв.

UDC 004.056.5

Steganalysis method efficient for the hidden communication channel with low capacity / *I.I. Bobok, A.A. Kobozeva //* Radiotekhnika : All-Ukr. Sci. Interdep. Mag. 2019. №198. P. 19 – 31.

The Least Significant Bit (the LSB-method) remains one of the main steganalysis methods used nowadays for the hidden communication channel organization. One of the features of the current use of the LSB method is an organizing of the hidden communication channel with low capacity. Under such conditions, the vast majority of existing steganalysis methods is ineffective. This paper is dedicated to the development of a new steganalysis method for detection of additional information in digital images embedded by the least significant bit modification. The method is based on the perturbation theory and matrix analysis and effective under the low capacity of the hidden communication channel. This method is based on the analysis of the normalized gap of maximum singular numbers for non-intersecting blocks of an image matrix, obtained by its standard splitting. It is shown that conversion of a digital image from the lossless format to the lossy format with different quality factors will lead to a monotonous increase in the number of blocks for which the normalized gap of block's maximum singular number increases with a decrease in the quality factor used in compression of the source image. This monotony will be broken in the case when the image originally stored in lossy format is being re-stored in lossy format. The conclusion made is the basis for the developed steganalysis method and the algorithm that implements it, which has polynomial complexity of degree 2. The proposed algorithm exceeds in efficiency the existing analogues when the embedding rate is less then 0.1 bits per pixel and effective for color and grayscale images. The conclusions are confirmed by the given results of a computational experiment, which have involved more than 5,000 digital images.

Key words: steganalysis method; digital image; low capacity of the hidden communication channel; singular numbers; singular number gap; the Least Significant Bit method.

6 tab. 4 fig. Ref.: 28 items.

УДК 681.3.06:519.248.681

Математическая модель сигналов с ортогональным частотным разделением и мультиплексированием (OFDM) / *И.Д. Горбенко, А.А. Замула, В.Л. Морозов, С.В. Родионов* // Радиотехника : Всеукр. межвед. науч.-техн. сб. 2019. Вып. 198. С. 32 – 43.

Рассматриваются проблемные вопросы построения помехозащищенных систем связи на основе использования технологии мультиплексирования сигналов с ортогональным частотным разделением каналов (OFDM). Предоставлено описание технологии формирования сигналов с OFDM, используемых в системах связи и телекоммуникаций, а также приводится анализ перспективных технологий, которые могут найти применение в системах широкополосной беспроводной связи с множеством несущих, к которым предъявляются повышенные требования по информационной безопасности, помехоустойчивости приема сигналов, скорости приема-передачи данных. Основной целью публикации является достаточно детальное рассмотрение некоторых проблем, связанных с разработками физически систем OFDM, получение математических моделей преобразований при реализации OFDM. Особое внимание при этом обращается на возможности обмена между качественными характеристиками системы связи и ее сложностью.

Ключевые слова: помехозащищенность; информационная безопасность; широкополосный доступ; сигнал; целостность; модуляция; преобразование Фурье; частотное разделение.

Ил. 6. Библиогр.: 13 назв.

УДК 681.3.06:519.248.681

Математична модель сигналів з ортогональним частотним розподілом і мультиплексуванням (OFDM) / *І.Д. Горбенко, О.А. Замула, В.Л. Морозов, С.В Родіонов* // Радіотехніка : Всеукр. міжвід. наук.-техн. зб. 2019. Вип. 198. С. 32 – 43.

Розглядаються проблемні питання побудови завадозахищених систем зв’язку на основі використання технології мультиплексування сигналів з ортогональним частотним розділенням каналів (OFDM). Надано опис технології формування сигналів з OFDM, що використовуються в системах зв'язку і телекомунікацій, а також наводиться аналіз перспективних технологій, які можуть знайти застосування в системах широкосмугового бездротового зв'язку з багатьма несійними, до яких висуваються підвищені вимоги щодо інформаційної безпеки, завадостійкості прийому сигналів, швидкості прийому-передачі даних. Основною метою публікації є достатньо детальний розгляд деяких проблем, пов'язаних з розробками на фізичному рівні систем OFDM, отримання математичних моделей перетворень при реалізації OFDM. Особу увагу при цьому звертається на можливості обміну між якісними характеристиками системи зв'язку і її складністю.

Ключові слова: завадо захищеність; інформаційна безпека; широкосмуговий доступ; сигнал; цілісність; модуляція; перетворення Фур’є; частотне розділення.

Іл. 6. Бібліогр.: 13 назв.

UDC 681.3.06:519.248.681

Mathematical model of orthogonal frequency distribution and multiplexing (OFDM) signals / *I.D. Gorbenko, O.A. Zamula, V.L. Morozov, S.V. Rodionov* // Radiotekhnika : All-Ukr. Sci. Interdep. Mag. 2019. №198. P. 32 – 43.

The problematic issues of building noise immunity systems based on the use of orthogonal frequency division-division (OFDM) signal multiplexing are considered. The description of OFDM signaling technology used in telecommunication and telecommunication systems is given, as well as the analysis of promising technologies that can be used in multi-carrier broadband wireless systems, which have high requirements for information security, noise immunity signal reception, data rate. The main purpose of the publication is a sufficiently detailed discussion of some of the problems associated with the development at the physical level of OFDM systems, obtaining mathematical models of transformations in the implementation of OFDM. Particular attention is paid to the possibility of exchange between the quality characteristics of the communication system and its complexity.

Key words: noise immunity; information security; broadband access; signal; integrity; modulation; Fourier transform; frequency division.

6 fig. Ref.: 13 items.

УДК 004.056.5

Алгоритмы криптографического хеширования, которые применяются в современных блокчейн-системах / *А.А. Кузнецов, Ю.И. Горбенко, В.В. Оноприенко, И.В. Стельник, Д.В. Мялковский* // Радиотехника : Всеукр. межвед. науч.-техн. сб. 2019. Вып. 198. С. 44 – 53.

Проводится анализ функций хеширования, которые применяются или могут применяться в различных блокчейн-системах. В частности, рассматриваются наиболее распространенные национальные и международные стандарты, в которых приведены спецификации всемирно известных алгоритмов криптографического хеширования, и исследуются различные проекты по построению децентрализованных блокчейн-систем, где эти функции могут быть применены.

Ключевые слова: хеширования; криптографический алгоритм; блокчейн; криптовалюта.

Табл. 2. Ил. 1. Библиогр.: 31 назв.

УДК 004.056.5

Алгоритми криптографічного гешування, які застосовуються в сучасних блокчейн-системах / *О.О. Кузнецов, Ю.І. Горбенко, В.В. Онопрієнко, І.В. Стельник, Д.В. Мялковський* // Радіотехніка : Всеукр. міжвід. наук.-техн. зб. 2019. Вип. 198. С. 44 – 53.

Проводиться аналіз функцій гешування, які застосовуються або можуть застосовуватися в різних блокчейн-системах. Зокрема, розглядаються найбільш поширені національні та міжнародні стандарти, в яких наведено специфікацію всесвітньо відомих алгоритмів криптографічного гешування, та досліджуються різні проекти з побудови децентралізованих блокчейн-систем, де ці функції можуть бути застосовані.

Ключові слова: гешування; криптографічний алгоритм; блокчейн; криптовалюта.

Табл. 2. Іл. 1. Бібліогр.: 31 назв.

UDC 004.056.5

Cryptographic hashing algorithms used in modern blockchain systems / *A.A. Kuznetsov, Yu.I. Gorbenko, V.V. Onoprienko, I.V. Stelnik, D.V. Myalkovsky* // Radiotekhnika : All-Ukr. Sci. Interdep. Mag. 2019. №198. P. 44 – 54.

The analysis of hashing functions that are applied or can be used in various blockchain systems is carried out. In particular, the most common national and international standards are considered, which contain specifications of world-famous cryptographic hashing algorithms, and various projects for the construction of decentralized blockchain systems where these functions can be applied are investigated.

Key words: hashing; cryptographic algorithm; blockchain; cryptocurrency.

2 tab. 1fig. Ref.: 31 items.

УДК 004.056.5

Исследование алгоритмов криптографического хеширования, которые применяются в современных блокчейн-системах / *А.А. Кузнецов, Ю.И. Горбенко, В.В. Ономриенко, И.В. Стельник, Д.В. Мялковский* // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2019. Вып. 198. С. 54 – 74.

Исследуются современные алгоритмы хеширования, которые применяются или могут применяться в различных блокчейн-системах. В частности, рассматриваются наиболее распространенные и применяемые алгоритмы криптографического хеширования, которые стандартизированы на международном и национальном уровнях, а также алгоритмы, хотя и не стандартизированные, но которые применяются в большинстве современных децентрализованных системах, построенных по технологии блокчейн.

Ключевые слова: хеширование; криптографический алгоритм; блокчейн; криптовалюта.

Табл. 1. Библиогр.: 94 назв.

УДК 004.056.5

Дослідження алгоритмів криптографічного гешування, які застосовуються в сучасних блокчейн-системах / *О.О. Кузнецов, Ю.І. Горбенко, В.В. Онопрієнко, І.В. Стельник, Д.В. Мялковський* // Радіотехніка : Всеукр. міжвід. наук.-техн. зб. 2019. Вип. 198. С. 54 – 74.

Досліджуються сучасні алгоритми гешування, які застосовуються або можуть застосовуватися в різних блокчейн-системах. Зокрема, розглядаються найбільш поширені та застосовувані алгоритми криптографічного гешування, які стандартизовані на міжнародному та національному рівнях, а також алгоритми, які хоча і не стандартизовані, але застосовуються у більшості сучасних децентралізованих системах, побудованих за технологією блокчейн.

Ключові слова: гешування; криптографічний алгоритм; блокчейн; криптовалюта.

Табл. 1. Бібліогр.: 94 назв.

UDC 004.056.5

The study of cryptographic hashing algorithms used in modern blockchain systems / *A.A. Kuznetsov, Yu.I. Gorbenko, V.V. Onoprienko, I.V. Stelnik, D.V. Myalkovsky* // Radiotekhnika : All-Ukr. Sci. Interdep. Mag. 2019. №198. P. 54 – 74.

Modern hashing algorithms that are or can be used in various blockchain systems are studied in this work. In particular, the most common and used cryptographic hashing algorithms are considered, which are standardized at the international and national levels, as well as algorithms, although not standardized, but used in most modern decentralized systems built on blockchain technology.

Key words: hashing; cryptographic algorithm; blockchain; cryptocurrency.

1 tab. Ref.: 94 items.

УДК 004.056.5

Исследование быстродействия и статистической безопасности алгоритмов криптографического хеширования / *А.А. Кузнецов, В.А. Тимченко, К.Е. Лисицкий, М.Ю. Родинко, М.С. Луценко, К.Ю. Шеханин, А.А. Колгатин* // Радиотехника : Всеукр. межвед. науч.-техн. сб. 2019. Вып. 198. С. 75 – 95.

Проводятся сравнительные исследования алгоритмов криптографического хеширования, которые применяются (или могут применяться) в современных децентрализованных блокчейн системах. В частности исследуется быстродействие хеширования на разных десктопных системах, оценивается количество тактов вычисли-

тельной системы на один байт (Cycles / byte), объем гешованого сообщения за одну секунду (MB / s) и количество сформированных хеш-кодов в секунду (KHash / s). Дополнительно исследуется быстродействие отдельных криптографических функций хеширования на графических вычислителях. Для оценки статистической безопасности исследуются исходные последовательности криптографических функций хеширования при обработке ими чрезмерных входных данных (которые сформированы с помощью обычного счетчика). Для сравнительных исследований показателей статистической безопасности используется методика NIST STS (Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications), которая рекомендована Национальным институтом стандартов и технологий США для исследования генераторов случайных и псевдослучайных чисел для криптографических приложений.

Ключевые слова: хеширование; быстродействие; криптографический алгоритм; блокчейн; криптовалюта.

Табл. 5. Ил. 42. Библиогр.: 21 назв.

УДК 004.056.5

Дослідження швидкодії та статистичної безпеки алгоритмів криптографічного гешування / *О.О. Кузнецов, В.А. Тимченко, К.Є. Лисицький, М.Ю. Родінко, М.С. Луценко, К.Ю. Шеханін, А.О. Колгатін* // Радіотехніка : Всеукр. міжвід. наук.-техн. зб. 2019. Вип. 198. С. 75 – 95.

Проводяться порівняльні дослідження алгоритмів криптографічного гешування, які застосовуються (або можуть застосовуватися) в сучасних децентралізованих блокчейн системах. Зокрема досліджується швидкодія гешування на різних десктопних системах, оцінюється кількість тактів обчислювальної системи на один байт (Cycles/byte), обсяг гешованого повідомлення за одну секунду (MB/s) та кількість сформованих геш-кодів за секунду (KHash/s). Додатково досліджується швидкодія окремих криптографічних функцій гешування на графічних обчислювачах. Для оцінки статистичної безпеки досліджуються вихідні послідовності криптографічних функцій гешування при обробці ними надмірних вхідних даних (які сформовано за допомогою звичайного лічильника). Для порівняльних досліджень показників статистичної безпеки використовується методика NIST STS (Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications), яку рекомендовано Національним інститутом стандартів і технологій США для дослідження генераторів випадкових і псевдовипадкових чисел для криптографічних застосувань.

Ключові слова: гешування; швидкодія; криптографічний алгоритм; блокчейн; криптовалюта.

Табл. 5. Іл. 42. Бібліогр.: 21 назв.

UDC 004.056.5

The study of the speed and statistical security of cryptographic hashing algorithms / *A.A. Kuznetsov, V.A. Timchenko, K.E. Lisitzky, M.Yu. Rodinko, M.S. Lutsenko, K.Yu. Shehanin, A.A.* // Radiotekhnika: All-Ukr. Sci. Interdep. Mag. 2019. №198. P. 75 – 95.

Comparative studies of cryptographic hashing algorithms are being carried out, which are used (or can be applied) in modern decentralized blockchain systems. In particular, hashing speed of action is studied on different desktop systems, the number of clock cycles of the computing system per byte (Cycles / byte), the volume of the hashed message per second (MB / s) and the number of generated hash codes per second (KHash / s) are estimated. Additionally, the speed of action speed of action of individual cryptographic hashing functions on graphical computers is investigated. To evaluate statistical security, we study the initial sequences of cryptographic hash functions when they process excessive input data (which are generated using a conventional counter). For comparative studies of statistical security indicators, the NIST STS (Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications) technique is used, which is recommended by the National Institute of Standards and Technology for the study of random and pseudorandom number generators for cryptographic applications.

Key words: hashing; speed of action performance; cryptographic algorithm; blockchain; cryptocurrency.

5 tab. 42 fig. Ref.: 21 items.

АНАЛИЗ И ИСПОЛЬЗОВАНИЕ КРИПТОГРАФИЧЕСКИХ МЕТОДОВ В ДЕЦЕНТРАЛИЗОВАННЫХ ТЕХНОЛОГИЯХ

АНАЛІЗ ТА ЗАСТОСУВАННЯ КРИПТОГРАФІЧНИХ МЕТОДІВ В ДЕЦЕНТРАЛІЗОВАНИХ ТЕХНОЛОГІЯХ

ANALYSIS AND USE OF CRYPTOGRAPHIC METHODS IN DECENTRALIZED TECHNOLOGIES

УДК 004.056.5

Установление протоколов доверия в сети взаимного недоверия путем формирования консенсуса / *Е.В. Исирова, А.В. Потий, Jens Christian Claussen* // Радиотехника : Всеукр. межвед. науч.-техн. сб. 2019. Вып. 198. С. 96 – 104.

Любое взаимодействие между субъектами происходит через сети связей между ними. Важной целью является обеспечение безопасности таких взаимодействий, особенно при появлении технологий квантовых вычислений. Возможно, что в постквантовом периоде наиболее выгодными архитектурами сетей для проведения верификации будут именно распределенные. В работе приводится обоснование данного вопроса, подробно

проводится аналогия между распределенным формированием доверия согласно предложенным протоколам и формированием консенсуса в социальных сетях для различных топологий сетей. Иерархические сети в обеих областях демонстрируют самые медленные временные рамки формирования консенсуса. Сделан вывод, что это может служить аргументом в пользу создания механизмов распределенных протоколов, где важна масштабируемость с размером сети.

Ключевые слова: Формирование консенсуса в социальных сетях; voter model; формирования консенсуса в компьютерных сетях; иерархическая инфраструктура открытых ключей; распределенная инфраструктура открытых ключей; распределенные протоколы верификации; постквантовый период; технология Blockchain.

Іл. 7. Бібліогр.: 17 назв.

УДК 004.056.5

Встановлення протоколів довіри в мережі взаємної недовіри шляхом формування консенсусу / *К.В. Ісірова, О.В. Потій, Jens Christian Claussen* // Радіотехніка : Всеукр. міжвід. наук.-техн. зб. 2019. Вип. 198. С. 96 – 104.

Будь-яка взаємодія між суб'єктами відбувається через мережі зв'язків між ними. Важливою метою є забезпечення безпеки таких взаємодій, особливо при появі технологій квантових обчислень. Можливо, що в постквантовому періоді найбільш вигідними архітектурами мереж для проведення верифікації будуть саме розподілені. У роботі наведено обґрунтування даного питання, докладно проведено аналогію між розподіленим формуванням довіри згідно із запропонованими протоколам і формуванням консенсусу в соціальних мережах для різних топологій мереж. Ієрархічні мережі в обох областях демонструють самі повільні тимчасові рамки формування консенсусу. Зроблено висновок, що це може служити аргументом на користь створення механізмів розподілених протоколів, де важлива масштабованість з розміром мережі.

Ключові слова: Формування консенсусу в соціальних мережах; voter model; формування консенсусу в комп'ютерних мережах; ієрархічна інфраструктура відкритих ключів; розподілена інфраструктура відкритих ключів; розподілені протоколи верифікації; постквантовий період; технологія Blockchain.

Іл. 7. Бібліогр.: 17 назви.

UDC 004.056.5

Establishing trust protocols in mutual distrust network by consensus formation / *K. Isirova, O. Potii, J. Claussen* // Radiotekhnika : All-Ukr. Sci. Interdep. Mag. 2019. №198. P. 96 – 104.

Any interactions between actors take place through networks of connections between them. An important goal is to ensure the security of such interactions, especially in advent of quantum computing technologies. It might be that in the post-quantum world, the avatar of verification architectures will be manifested through distributed protocols. Here we augment the discussion by explicitely drawing the analogy between distributed protocol consensus formation and consensus formation in social networks in various topologies. Hierarchical networks, in both domains, exhibit slowest timescale of consensus formation. We conclude this supports universal argument towards establishment of distributed protocol mechanisms, wherever the scaleability with network size is of relevance.

Key words: Consensus Formation in Social Networks; Voter Model; Trust Formation in Computer networks; Hierarchical PKI; Distributed PKI; Distributed Verification Protocols; Post-quantum Period; Blockchain Technology.

7 fig. Ref.: 17 items.

УДК 004.773.2

Метод сравнения Proof of Work алгоритмов консенсуса / *М.О. Осадчук, Р.В. Олейников* // Радиотехника : Всеукр. межвед. науч.-техн. сб. 2019. Вып. 198. С. 105 – 112.

Алгоритм консенсуса представляет собой наиболее важную часть любой блокчейн системы. Существует множество алгоритмов консенсуса, которые разработчики могут использовать для своих решений, однако принятие такого решения не может быть полностью формализировано из-за неопределенности в требованиях и среде приложения. Предложен метод, который позволяет выбрать наиболее оптимальный Proof of Work алгоритм консенсуса для новых блокчейн-систем, который основан на процессе анализа иерархий. Применение этого метода для разных PoW алгоритмов и привлечение независимых экспертов позволяет выбрать dPoW в качестве наилучшего решения для существующих условий.

Ключевые слова: блокчейн; алгоритм консенсуса; децентрализованные вычисления; Proof of Work; атака двойной траты.

Табл. 7. Библиогр.: 25 назв.

УДК 004.773.2

Метод порівняння Proof of Work алгоритмів консенсусу / *М.О. Осадчук, Р.В. Олійников* // Радіотехніка : Всеукр. міжвід. наук.-техн. зб. 2019. Вип. 198. С. 105 – 112.

Алгоритм консенсусу є найбільш важливою частиною будь-якої блокчейн-системи. Існує багато алгоритмів консенсусу, які розробники можуть використовувати для своїх рішень, але прийняття такого рішення не може бути повністю формалізованим через невизначеність у вимогах та у середовищі додатку. Ми запропонували метод, який дозволяє обрати оптимальний Proof of Work алгоритм консенсусу для новостворених блокчейн-систем, який заснований на процесі аналізу ієрархій. Застосування цього методу для різних PoW алгорит-

мів із залученням незалежних експертів дозволяє обрати dPoW у якості найкращого рішення для існуючих умов.

Ключові слова: блокчейн; алгоритм консенсусу; децентралізовані обчислення; Proof of Work; атака подвійного витрачання.

Табл. 7. Бібліогр.: 25 назв.

UDC 004.773.2

Method of Proof of Work consensus algorithms comparison / *M. Osadchuk, R. Oliynykov* // Radiotekhnika : All-Ukr. Sci. Interdep. Mag. 2019. №198. P. 105 – 112.

A consensus algorithm is the most important part of any blockchain system. There are available various consensus algorithms that developers can utilize in their solutions, and such a decision making cannot be fully formalized due uncertainty in requirements and application environment. We propose a method that allows selecting of an optimal Proof of Work (PoW) consensus algorithm for newly developed blockchain system based on Analytic Hierarchy Process. Application of this method to various PoW algorithms with involvement of independent experts allowed to select dPoW as the best solution for the given conditions.

Key words: blockchain; consensus algorithm; decentralized computation; Proof of Work; double spend attack.

7 tab. Ref.: 25 items.

УДК 004.056.5

Некоторый подход к маскированию данных как средство противодействия угрозе логического вывода / *В.И. Есин, В.В. Вилигура* // Радиотехника : Всеукр. межвед. науч.-техн. сб. 2019. Вып. 198. С. 113 – 130.

Цель статьи – раскрытие сути некоторого подхода к маскированию данных, хранящихся в базе данных, как средства противодействия угрозе логического вывода. В основу подхода положены принципы случайной перестановки элементов поля данных столбца строки таблицы производственной базы данных и динамического маскирования. Отличительной особенностью предлагаемого решения является подход к процессу перемешивания данных, а именно, перемешиванию элементов значения данных внутри требуемого поля строки. С помощью данного решения возможно маскирование как всего значения поля столбца строки таблицы, так и его части. Предлагаемый подход отличается от большей части типичных коммерческих инструментов маскирования критических данных тем, что в базе данных выполняются предварительные физические изменения конфиденциальных данных, и эти изменения при необходимости можно отменить пользователем, который имеет соответствующие права на это. Легитимный пользователь получает доступ к конфиденциальным данным за счет возможности осуществить преобразование (перезапись) запроса «на лету», а злоумышленник может только считать хранящиеся в базе заранее измененные определенным образом с сохранением исходного формата данные. Предлагаемый подход к маскированию данных может быть использован как в производственных, так и в непроизводственных базах данных, расширяя возможности, так называемого, статического маскирования данных.

Ключевые слова: безопасность данных; база данных; маскирование данных; конфиденциальные данные.

Табл. 4. Ил. 2. Библиогр.: 34 назв.

УДК 004.056.5

Деякий підхід до маскування даних як засіб протидії загрозі логічного висновку / *В.І. Єсін, В.В. Вілігура* // Радіотехніка : Всеукр. міжвід. наук.-техн. зб. 2019. Вип. 198. С. 113 – 130.

Мета статті – розкриття суті деякого підходу до маскування даних, що зберігаються в базі даних, як засобу протидії загрозі логічного висновку. В основу підходу було покладено принципи випадкової перестановки елементів поля даних стовпця рядка таблиці виробничої бази даних і динамічного маскування. Відмінною особливістю запропонованого рішення є підхід до процесу перемішування даних, а саме – перемішування елементів значення даних всередині потрібного поля рядка. За допомогою даного рішення можливо маскування як всього значення поля стовпця рядка таблиці, так і його частини. Запропонований підхід відрізняється від більшої частини типових комерційних інструментів маскування критичних даних тим, що в базі даних виконуються попередні фізичні зміни конфіденційних даних, і ці зміни при необхідності можна скасувати користувачем, який має відповідні права на це. Легітимний користувач отримує доступ до конфіденційних даних за рахунок можливості здійснити перетворення (перезапис) запиту «на льоту», а зловмисник може тільки зчитувати заздалегідь змінені певним чином зі збереженням вихідного формату дані, що зберігаються в базі. Запропонований підхід до маскування даних може бути використаний як в виробничих, так і в невиробничих базах даних, розширюючи можливості так званого статичного маскування даних.

Ключові слова: безпека даних; база даних; маскування даних; конфіденційні дані.

Табл. 4. Іл. 2. Бібліогр.: 34 назв.

UDC 004.056.5

Some approach to data masking as means to counteract the inference threat / *V.I. Yesin, V.V. Vilihura* // Radiotekhnika : All-Ukr. Sci. Interdep. Mag. 2019. №198. P. 113 – 130.

The goal of the article is to reveal the essence of some approach to data masking stored in the database as a means to counteract the inference threat. This approach is based on the principles of random permutation of the elements of a

data field of the row column of the production database table data and dynamic masking. A distinctive feature of the proposed solution is the approach to the process of data shuffling, namely, shuffling data value elements within the demanded row field. It is possible to mask both an entire value of the field of the table row column and its part using this solution. The proposed approach differs from most of the typical commercial tools for masking sensitive data in that a preliminary physical change of sensitive data is made in the production database, and a user who has the appropriate rights can cancel these changes if it is necessary. The legitimate user in the proposed approach gets access to sensitive data due to the ability to transform (rewrite) the query "on the fly", and the attacker can only read the previously modified data that is stored in the database. The proposed approach to data masking can be used in both production and non-production databases, expanding the possibilities of so-called static data masking.

Key words: data security; database; data masking; sensitive data.

4 tab. 2 fig. Ref.: 34 items.

УДК 004.056.55

Современные проблемы централизованных технологий типа «клиент-сервер» и возможности их усовершенствования на основе децентрализации / *Ю.И. Горбенко, М.В. Есина, Д.В. Мялковский, О.С. Акользина, В.А. Пономарь* // Радиотехника : Всеукр. межвед. науч.-техн. сб. 2019. Вып. 198. С. 131 – 145.

Приводится анализ основных принципов построения децентрализованных технологий с использованием технологии блокчейн и требования к ним в части безопасности, а также анализ особенностей и условий применения защищенных технологий блокчейн. Описываются и анализируются потенциальные атаки, когда применение блокчейна является существенным механизмом защиты от них. Приводится сущность и предложения относительно противодействия атакам специального вида.

Ключевые слова: децентрализация, информационные технологии, клиент-серверная технология, централизованная технология.

Табл. 5. Ил. 1. Библиогр.: 28 назв.

УДК 004.056.55

Сучасні проблеми централізованих технологій типу «клієнт – сервер» та можливості їх удосконалення на основі децентралізації / *Ю.І. Горбенко, М.В. Єсіна, Д.В. Мялковський, О.С. Акользіна, В.А. Пономарь* // Радіотехніка : Всеукр. міжвід. наук.-техн. зб. 2019. Вип. 198. С. 131 – 145.

Наводиться аналіз основних принципів побудування децентралізованих технологій з використанням технології блокчейн та вимоги до них в частині безпеки, а також аналіз особливостей та умов застосування захищених технологій блокчейн. Описуються та аналізуються потенційні атаки, коли застосування блокчейну є суттєвим механізмом захисту від них. Наводиться сутність та пропозиції відносно протидії атакам спеціального виду.

Ключові слова: децентралізація, інформаційні технології, клієнт-серверна технологія, централізована технологія.

Табл. 5. Іл. 1. Бібліогр.: 28 назв.

UDC 004.056.55

Modern problems of centralized technologies of the client-server type and possibilities of their improvement on the basis of decentralization / *Yu.I. Gorbenko, M.V. Yesina, D.V. Myalkovskiy, O.S. Akolzina, V.A. Ponomar* // Radiotekhnika : All-Ukr. Sci. Interdep. Mag. 2019. №198. P. 131 – 145.

The paper analyzes the basic principles of building decentralized technologies with the use of blockchain technology and their requirements in terms of security, as well as the analysis of the features and conditions of the use of secure blockchain technologies. Potential attacks are described and analyzed when the use of blockchain is a significant defense mechanism against them. The essence and suggestions concerning counteraction to attacks of a special kind are given.

Key words: decentralization, information technology, client-server technology, centralized technology.

5 tab. 1 fig. Ref.: 28 items.

УДК 004.056.5

Моделирование атаки двойной траты на протокол консенсуса «Proof of work» / *Н.А. Полуяненко, А.А. Кузнецов* // Радиотехника : Всеукр. межвед. науч.-техн. сб. 2019. Вып. 198. С. 146 – 161.

Проведен критический анализ известных аналитических оценок вероятности успешной реализации атаки двойной траты на протокол консенсуса «Proof of work». В частности, рассмотрена «задача о разорении игрока», показано, что базовые предположения о вероятностном пространстве (множество элементарных исходов и вероятности их наступления) не соответствуют реальным процессам, протекающим при установлении консенсуса «Proof of work» в блокчейн-системе. Предложена модель «независимых игроков», которая устраняет основные неточности и несоответствия. Показана сходимость результатов теоретических расчетов с данными экспериментов по имитации «гонки» между честными игроками и злоумышленниками.

Ключевые слова: блокчейн; протокол консенсуса; атака двойной траты; имитационное моделирование.

Табл. 1. Ил. 11. Библиогр.: 16 назв.

УДК 004.056.5

Моделювання атаки подвійної витрати на протокол консенсусу «Proof of work» / *М.О. Полуяненко, О.О. Кузнецов* // Радіотехніка : Всеукр. міжвід. наук.-техн. зб. 2019. Вип. 198. С. 146 – 161.

Проведено критичний аналіз відомих аналітичних оцінок ймовірності успішної реалізації атаки подвійної витрати на протокол консенсусу «Proof of work». Зокрема, розглянуто «завдання про розорення гравця», показано, що базові припущення про імовірнісний простір (безліч елементарних фіналів і ймовірності їх настання) не відповідають реальним процесам, що протікають при встановленні консенсусу «Proof of work» в блокчейн-системі. Запропоновано модель «незалежних гравців», яка усуває основні неточності і невідповідності. Показана збіжність результатів теоретичних розрахунків з даними експериментів з імітації «гонки» між чесними гравцями і зловмисниками.

Ключові слова: блокчейн; протокол консенсусу; атака подвійної витрати; імітаційне моделювання.

Табл. 1. Іл. 11. Бібліогр.: назв.

UDC 004.056.5

Simulation of double spend attack on the “Proof of Work” consensus protocol / *N.A. Poluyanenko, A.A. Kuznetsov* // Radiotekhnika : All-Ukr. Sci. Interdep. Mag. 2019. №198. P. 146 – 161.

A critical analysis of the well-known analytical estimates of the probability of successful implementation of a double-spending attack on the “Proof of work” consensus protocol has been carried out. In particular, the so-called “Player ruin problem” is considered, it is shown that the basic assumptions about the probability space (the set of elementary outcomes and the likelihood of their occurrence) do not correspond to the real processes that occur when the “Proof of work” consensus is established in the blockchain system. A model of “independent players” is proposed, which eliminates the main inaccuracies and inconsistencies. The convergence of the results of theoretical calculations with the data of experiments to simulate the "race" between honest players and attackers is shown.

Key words: blockchain; consensus protocol; double waste attack; simulation modeling.

1 tab. 11 fig. Ref.: 16 items.

УДК 004.056.55

Принципы построения и анализа инфраструктур открытого ключа на основе применения технологии блокчейн / *И.Д. Горбенко, А.В. Потий, Ю.И. Горбенко, А.И. Пушкарёв, М.В. Есина* // Радиотехника : Всеукр. межвед. науч.-техн. сб. 2019. Вып. 198. С. 162 – 181.

Обоснованы возможности и необходимость создания инфраструктуры открытых ключей на основе технологии блокчейн. Анализируется усовершенствованная модель инфраструктуры открытых ключей с прозрачностью сертификатов на основе блокчейна, а также основные проблемные вопросы перспективных инфраструктур открытых ключей на базе блокчейна. Проводится общая оценка устойчивости инфраструктуры открытых ключей на основе блокчейна к известным атакам.

Ключевые слова: блокчейн; децентрализация; информационные технологии; технология блокчейн.

Табл. 1. Ил. 3. Библиогр.: 40 назв.

УДК 004.056.55

Принципи побудування та аналізу інфраструктур відкритого ключа на основі застосування технології блокчейн / *І.Д. Горбенко, О.В. Потій, Ю.І. Горбенко, А.І. Пушкарьов, М.В. Єсіна* // Радіотехніка : Всеукр. міжвід. наук.-техн. зб. 2019. Вип. 198. С. 162 – 181.

Наводиться обґрунтування можливостей та необхідності створення інфраструктури відкритих ключів на основі технології блокчейн. Аналізується удосконалена модель інфраструктури відкритих ключів з прозорістю сертифікатів на основі блокчейну, а також основні проблемні питання перспективних інфраструктур відкритих ключів на базі блокчейну. Проводиться загальна оцінка стійкості інфраструктури відкритих ключів на основі блокчейну до відомих атак.

Ключові слова: блокчейн; децентралізація; інформаційні технології; технологія блокчейн.

Табл. 1. Іл. 3. Бібліогр.: 40 назв.

UDC 004.056.55

Principles of building and analyzing public key infrastructures based on the use of blockchain technology / *I.D. Gorbenko, O.V. Potii, Yu.I. Gorbenko, A.I. Pushkarov, M.V. Yesina* // Radiotekhnika : All-Ukr. Sci. Interdep. Mag. 2019. №198. P. 162 – 181.

The paper provides a rationale for the possibilities and the need to create blockchain-based public key infrastructure. An improved public key infrastructure model with blockchain-based certificate transparency as well as the main problematic issues of promising blockchain-based public key infrastructures are analyzed. A general assessment of the public key infrastructure based on the blockchain stability in conditions of well-known attacks is carried out.

Key words: blockchain, decentralization, information technology, blockchain technology.

1 tab. 3 fig. Ref.: 40 items.

УДК 004.728:004.728.3, 004.056.055

Оптимизация методов синтеза дискретных сложных сигналов в современных многопользовательских системах связи широкополосного доступа / *А.А. Замула* // Радиотехника : Всеукр. межвед. науч.-техн. сб. 2019. Вып. 198. С. 182 – 191.

Среди основных направлений улучшения показателей эффективности функционирования информационно-коммуникационных систем (ИКС), в частности помехозащищенности, скрытности, информационной безопасности, можно выделить направления, связанные с применением фазоманипулированных широкополосных сигналов (ФМ ШПС). Многочисленные приложения ИКС указывают на важность периодических автокорреляционной свойств сигналов: дальномерные системы с непрерывным излучением, пилотный канал и канал синхронизации в цифровых системах передачи данных, радарные и сонорные системы и др. Кроме того, хорошая периодическая автокорреляционная функция автокорреляции (АКФ) сигнала указывает на возможность отбора сигналов с хорошими апериодическими АКФ. В данной работе сформулирована и решена задача оптимизации синтеза нелинейных дискретных последовательностей, которые имеют улучшенные ансамблевые, структурные и автокорреляционные свойства. Применение нелинейных дискретных сигналов, образованные на основе таких последовательностей, позволит обеспечить необходимые значения помехозащищенности, информационной и структурной скрытности функционирования ИКС.

Ключевые слова: дискретная последовательность; криптографический сигнал; функция корреляции; конечное поле; база сигнала.

Табл. 2. Ил. 3. Библиогр.: 9 назв.

УДК 004.728:004.728.3, 004.056.055

Оптимізація методів синтезу дискретних складних сигналів у сучасних багатокористувачевих системах зв'язку широкосмугового доступу / *О.А. Замула* // Радіотехніка : Всеукр. міжвід. наук.-техн. зб. 2019. Вип. 198. С. 182 – 191.

Серед основних напрямків покращення показників ефективності функціонування інформаційно-комунікаційних систем (ІКС), зокрема завадозахищеності, скритності, інформаційної безпеки, можна виділити напрямки, пов'язані із застосуванням фазоманіпульованих широкосмугових сигналів (ФМ ШПС). Чисельні додатки ІКС вказують на важливість періодичних автокореляційних властивостей сигналів: дальномірні системи з безперервним випромінюванням, пілотний канал і канал синхронізації в цифрових системах передачі даних, радарні і сонорні системи і ін. Крім того, хороша періодична автокореляційна функція автокореляції (АКФ) сигналу вказує на можливість відбору сигналів з хорошими аперіодичними АКФ. У даній роботі сформульована і вирішена задача оптимізації синтезу нелінійних дискретних послідовностей, які мають покращені ансамблеві, структурні і автокореляційні властивості. Застосування нелінійних дискретних сигналів, які утворені на основі таких послідовностей, дозволить забезпечити необхідні значення завадозахищеності, інформаційної та структурної скритності функціонування ІКС.

Ключові слова: дискретна послідовність; криптографічний сигнал; функція кореляції; кінцеве поле; база сигналу.

Табл. 2. Іл. 3. Бібліогр.: 9 назв.

UDC 004.728:004.728.3, 004.056.055

Optimization of the method for the synthesis of discrete folding signals in the most common bag-box-and-bag systems / *A.A. Zamula* // Radiotekhnika : All-Ukr. Sci. Interdep. Mag. 2019. №198. P. 182 – 191.

Among the main areas of improvement of the performance indicators of information and communication systems (ICS), in particular, noise immunity, secrecy, and information security, it is possible to identify the areas associated with the use of phase-manipulated broadband signals (FM SHPS). Numerous ICS applications point to the importance of periodic autocorrelation properties of the signals used: continuous-beam systems with continuous radiation, pilot channel and channel of synchronization in digital data systems, radar and sonar systems, and others. In addition, a good periodic AKF signal indicates the ability to select signals with good aperiodic ACF. The minimization of the level of the side petals of the AKF is of greatest importance when designing a signal for such applications as measuring the lag time, time resolution, etc. In this paper, the problem of optimizing the synthesis of nonlinear discrete sequences, which have improved ensemble, structural and autocorrelation properties, is formulated and solved. The use of nonlinear discrete signals, which are formed on the basis of such sequences, will provide the necessary values of impedance protection, information and structural secrecy of the operation of the ICS.

Key words: discrete sequence; cryptographic signal; correlation function; isomorphism; finite field; base of signal.

2 tab. 3 fig. Ref.: 9 items.

УДК 004.491.4

Метод преодоления средств защиты с использованием уязвимостей графических файлов формата BMP / *Р.С. Гринев, А.В. Северинов* // Радиотехника : Всеукр. межвед. науч.-техн. сб. 2019. Вып. 198. С. 192 – 202.

Цель статьи – исследование уязвимостей современных систем защиты от атак с использованием графических файлов формата BMP. Рассматриваются особенности изображений формата BMP, способ их использования для внедрения компьютерных вирусов и проведения атак с целью преодоления средств защиты. Проанали-

зирована эффективность предложенного метода сокрытия компьютерных вирусов по сравнению с известными, показана возможность преодоления средств защиты.

Ключевые слова: файл изображения формата BMP; компьютерный вирус; шелл-код; преодоление систем защиты; сокрытия вируса; антивирус; IDS; IPS; уязвимость; эксплойт.

Табл. 3. Ил. 23. Библиогр.: 11 назв.

УДК 004.491.4

Метод подолання засобів захисту з використанням вразливостей графічних файлів формату BMP / *Р.С. Гриньов, О.В. Сєвєрінов* // Радіотехніка : Всеукр. міжвід. наук.-техн. зб. 2019. Вип. 198. С. 192 – 202.

Мета статті – дослідження вразливостей сучасних систем захисту від атак з використанням графічних файлів формату BMP. Розглядаються особливості зображень формату BMP, спосіб їх використання для впровадження комп'ютерних вірусів та проведення атак з метою подолання засобів захисту. Проаналізована ефективність запропонованого методу приховування комп'ютерних вірусів в порівнянні з відомими, показана можливість подолання засобів захисту.

Ключові слова: файл зображення формату BMP; комп'ютерний вірус; шелл-код; подолання систем захисту; приховування вірусу; антивірус; IDS; IPS; вразливість, експлойт.

Табл. 3. Іл. 23. Бібліогр.: 11 назв.

UDC 004.491.4

The method of overcoming protection using vulnerabilities of graphic files in BMP / *R.S. Grynov, A.V. Severinov* // Radiotekhnika : All-Ukr. Sci. Interdep. Mag. 2019. №198. P. 192 – 202.

The aim of the article – study attacks on modern protection systems by using vulnerabilities of BMP image files. This article describes the features of BMP format images. The method of injecting computer viruses in BMP image and attacks in order to overcome the means of protection. The analysis of the efficiency of the proposed method of hiding computer viruses in comparison with the known ones showed the possibility of overcoming the means of protection.

Key words: BMP image file; computer virus; shell code; overcoming protection systems; virus hiding; antivirus; IDS; IPS; vulnerability; exploit.

3 tab. 23 fig. Ref.: 11 items.

УДК 004.056

Сравнительный анализ криптопреобразований на эллиптических кривых и кривых Эдвардса / *В.А. Кулибаба* // Радиотехника : Всеукр. межвед. науч.-техн. сб. 2019. Вып. 198. С. 203 – 208.

Выполнен сравнительный анализ основных преобразований на канонических эллиптических кривых и кривых Эдвардса. Приведены сравнения производительности групповых операций в группах точеки эллиптических кривых и кривых Эдвардса. Показана возможность использовать алгоритм Полларда для криптоанализа кривых Эдвардса, а также ускорение генерации последовательности для алгоритма ро-Поларда для кривых Эдвардса, что позволяет ускорить выполнение криптоанализа. Предложена оценка стойкости кривых Эдвардса против атаки типа «полное раскрытие» с использованием дискретного логарифма в группе точек кривых Эдвардса.

Ключевые слова: эллиптические кривые, кривые Эдвардса, криптоанализ, цифровая подпись.

Табл. 1. Библиогр.: 7 назв.

УДК 004.056

Порівняльний аналіз криптоперетворень на еліптичних кривих та кривих Едвардса / *В.А. Кулібаба* // Радіотехніка : Всеукр. міжвід. наук.-техн. зб. 2019. Вип. 198. С. 203 – 208.

Виконано порівняльний аналіз основних перетворень на канонічних еліптичних кривих та кривих Едвардса. Наведено порівняння швидкодії групових операцій в групах точок еліптичних кривих та кривих Едвардса. Показана можливість використання алгоритму Полларда для криптоаналізу кривих Едвардса, а також прискорення генерації послідовності для алгоритму ро-Поларда для кривих Едвардса, що дозволяє пришвидшити виконання криптоаналізу. Запропоновано оцінку стійкості кривих Едвардса проти атаки типу «повне розкриття» з використанням дискретного логарифму в групі точок кривих Едвардса.

Ключові слова: еліптичні криві, криві Едвардса, криптоаналіз, електронний підпис.

Табл. 1. Бібліогр.: 7 назв.

UDC 004.056

Comparative analysis of cryptoprimitives on canonical elliptic curves and Edwards curves / *V. Kulibaba* // Radiotekhnika : All-Ukr. Sci. Interdep. Mag. 2019. №198. P. 203 – 208.

The article provides a comparative analysis of the basic transformations on canonical elliptic curves and Edwards curves. Comparisons of the performance of group operations in groups of points of elliptic curves and Edwards curves are given. The possibility of using the Pollard algorithm for cryptanalysis of Edwards curves, as well as the acceleration of the sequence generation for the ro-Polard algorithm for Edwards curves is shown, which allows to accelerate the execution of cryptanalysis. The paper proposes an assessment of the resistance of Edwards curves against attacks of the type "full disclosure" using the discrete logarithm in the Edwards curve point group.

Key words: elliptic curves, Edwards curves, cryptanalysis, digital signature.

1 tab. Ref.: 7 items.

УДК 004.056 55

Стойкость модифицированной цифровой подписи EdDSA / *А. Бессалов, Л. Ковальчук, Н. Кучинская, А. Телиженко* // Радиотехника : Всеукр. межвед. науч.-техн. сб. 2019. Вып. 198. С. 209 – 214.

Украинский Национальный Стандарт Цифровой Подписи ДСТУ 4145-2002 используется уже почти 17 лет. За это время в области информационных технологий произошли существенные изменения, которые непосредственно влияют на применение этого стандарта и указывают на необходимость его модернизации.

В связи с необходимостью пересмотра и обновления Национального стандарта цифровой подписи ДСТУ 4145-2002 авторы рассматривают несколько конструкций цифровых подписей. Среди требований к современной цифровой подписи следует упомянуть как минимум 128-битовый уровень стойкости, быстрые алгоритмы подписи и её проверки, быструю генерацию ключей, надёжность сеансовых ключей, стойкость к коллизиям, безопасная программная реализация и тд. Существует множество очевидных вариантов среди классических и эллиптических систем подписи, Эль-Гамаля, Шнорра, ECDSA, другие, которые могут использоваться в переходный период к постквантовой криптографии.

Предлагается одна из возможных модификаций схемы цифровой подписи, базирующейся на алгоритме The Edwards-curve Digital Signature Algorithm (EdDSA). Основные преимущества предложенной в этой работе модификации состоят в следующем:

1) схема подписи является стойкой даже в случае сбоя генератора сеансовых ключей;

2) время реализации подписи не зависит от длины сообщения;

3) стойкость к атаке со связанными ключами.

Ключевые слова: кривая Эдвардса; цифровая подпись; EdDSA.

Библиогр.: 12 назв.

УДК 004.056 55

Стійкість модифікованого цифрового підпису EdDSA / *А. Бессалов, Л. Ковальчук, Н. Кучиньска, О. Теліженко* // Радіотехніка : Всеукр. міжвід. наук.-техн. зб. 2019. Вип. 198. С. 209 – 214.

Український Національний Стандарт Цифрового Підпису ДСТУ 4145-2002 використовується вже майже 17 років. За цей час у галузі інформаційних технологій відбулись суттєві зміни, які безпосередньо впливають на використання цього стандарту та вказують на необхідність його модернізації.

У зв'язку з необхідністю перегляду та оновлення Національного стандарту цифрового підпису ДСТУ 4145-2002 автори розглядають кілька конструкцій цифрових підписів. Серед вимог до сучасного цифрового підпису слід зазначити як мінімум 128-бітовий рівень стійкості, швидкі алгоритми підпису та його перевірки, швидку генерацію ключів, надійність сеансових ключів, стійкість до колізій, безпечну програмну реалізацію і т. і. Існує багато очевидних варіантів серед класичних та еліптичних систем цифрового підпису, Ель-Гамаля, Шнорра, ECDSA, інші, які можна використовувати у перехідний до постквантового періода.

Пропонується одна з можливих модифікацій схеми цифрового підпису, що базується на алгоритмі The Edwards-curve Digital Signature Algorithm (EdDSA). Основними перевагами модифікації, яка запропонована у цій роботі, є наступні:

1) схема підпису є стійкою навіть у випадку збоїв у роботі генератора сеансовіх ключів;

2) час реалізації підпису не залежить від довжини повідомлення;

3) стійкість до атаки зі зв'язаними ключами.

Ключові слова: крива Едвардса; цифровий підпис; EdDSA.

Бібліогр.: 12 назв.

UDC 004.056 55

Securuty of modified digital public-key signature EdDSA / *A. Bessalov, L. Kovalchuk, N. Kuchynska, O. Telizhenko* // Radiotekhnika : All-Ukr. Sci. Interdep. Mag. 2019. №198. P. 209 – 214.

The Ukrainian National Standard for Digital Signature DSTU 4145-2002 has been in use about 17 years. During this time, significant changes have occurred in the field of information technology, which directly affect the implementation of the current National Standard for Digital Signature DSTU 4145-2002 and indicate the need for its modernization.

Due to the need to revise and update national digital signature standard DSTU 4145-2002, the authors considered several digital signature constructions. Among the requirements to modern public-key signatures it is worth to highlight at least 128-bit security, fast signing and fast signature verification, fast keys generation, foolproof session keys, collision resistance, secure software implementation, etc. There are a lot of obvious variants in classic and elliptic signature systems, ElGamal, Schnorr`s, ECDSA, etc, which can be used in transitional to post quantum period.

This paper introduces one of possible modifications for signature schemes based on the Edwards-curve Digital Signature Algorithm (EdDSA). The main advantages of the modification proposed in this work are:

1) the signature scheme is secure even if the session key generator fails;

2) signature implementation time does not depend on message length;

3) security against related-key attacks.

Key words: Edwards curve; digital signature; EdDSA.

12 items.

УДК 004.056.55

Применение хэш-функции купина в схеме подписей SPHINCS+ / *Д. Телевний* // Радиотехника : Всеукр. межвед. науч.-техн. сб. 2019. Вып. 198. С. 215 – 219.

В последние годы проведено значительное количество исследований по квантовым компьютерам, использующим квантово-механические явления для решения математических задач, которые являются сложными или неразрешимыми для обычных компьютеров. Возможность квантовых атак сформировала новую главу в области криптологии – постквантовую криптологию, где схемы ЭЦП стали одним из основных векторов исследований. Наиболее представительными выборками являются схемы, основанные на хэш-преобразованиях. Схемы подписи на основе хэша разработаны Лемпортом в качестве схемы одноразовой подписи в конце 1970-х годов и расширены для большего числа подписей Мерклом. В дальнейшем были введены более сложные схемы. NIST объявил о конкурсе новых стандартов постквантовой криптографии как для шифрования (генерации ключей), так и для подписей. На 2-й тур претендуют 9 кандидатов цифровой подписи. SPHINCS + (бывший SPHINCS) находится в списке. Алгоритм может быть кратко описан как схема подписи на основе хэша без сохранения состояния. Он использует много компонентов из XMSS, но работает с большими ключами и сигнатурой для устранения состояния. Схема может быть использована с различными хеш-функциями. Цель статьи – проанализировать применение хэш-функции национального стандарта в схеме кандидата NIST-кандидата SPHINCS +. Исследование показало, что хэш национального стандарта может быть применен к генерации случайных чисел и хешированию входного сообщения. Поскольку функция Kupyna возвращает выходные данные фиксированного размера, ее применение выглядит подобно хэш-функции SHA-256.

Ключевые слова: постквантовая криптография; схема подписи; хэш-функция; Купина; SPHINCS +; Меркле деревья.

Табл. 3. Ил. 1. Библиогр.: 9 назв.

УДК 004.056.55

Застосування хеш-функції Купина в схемі підпису SPHINCS+ / *Д. Телевний* // Радіотехніка : Всеукр. міжвід. наук.-техн. зб. 2019. Вип. 198. С. 215 – 219.

В останні роки проведено значну кількість досліджень квантових комп'ютерів – машин, які використовують квантові механічні явища для вирішення математичних задач, важких або нерозв'язних для звичайних комп'ютерів. Можливість квантових атак сформувала нову главу в галузі криптології – постквантову криптологію, де схеми DSA стали одним з основних векторів досліджень. Найбільш репрезентативними вибірками є схеми, засновані на хеш-перетвореннях. Схеми підписів Лемпорта на основі хешу були розроблені як одноразові схеми підпису в кінці 1970-х і розширені Меркле на багаторазові підписи. Надалі були запроваджені складніші схеми. NIST заявив про конкуренцію нових стандартів potquantum як для шифрування (генерації ключів), так і для підписів. У II турі є 9 кандидатів у цифровий підпис. SPHINCS + (колишній SPHINCS) є у списку. Алгоритм можна коротко охарактеризувати як схему підписів без збереження стану. Він використовує багато компонентів з XMSS, але працює з більшими ключами та підписом для усунення стану. Схему можна використовувати з різними хеш-функціями. Мета роботи – проаналізувати застосування національної стандартної хеш-функції схеми кандидата, що подає NIST SPHINCS +. Дослідження показало, що національний стандарт хеш може бути застосований до генерації випадкових випадків насіння та хешування вхідного повідомлення. Оскільки Fuction fuction повертає вихід фіксованого розміру, його застосування виглядає аналогічно хешам SHA-256.

Ключові слова: постквантова криптографія; схема підпису; хеш-фукція; Купина; SPHINCS+; дерева Меркла.

Табл. 3. Іл. 1. Бібліогр.: 9 назв.

UDC 004.056.55

The Kupyna hash function application to SPHINCS+ signatures / *D. Televnyi* // Radiotekhnika : All-Ukr. Sci. Interdep. Mag. 2019. №198. P. 215 – 219.

In recent years, there has been a substantial amount of research on quantum computers – machines that exploit quantum mechanical phenomena to solve mathematical problems that are difficult or intractable for conventional computers. The possibility of quantum attacks formed a new chapter in cryptology field – postquantum cryptology, where DSA schemes became one of the main research vectors. The most representative samples are schemes based on hash transformations. Hash-based signature schemes were developed as one-time signature schemes in the late 1970s by Lamport and extended to more signatures by Merkle. In further more complicated schemes were introduced. NIST declared about the competition of new postquantum standards both for encryption (key generation) and signatures As for the 2nd round there are 9 Digital signature candidates. SPHINCS+ (former SPHINCS) is in the list. The algorithm can be briefly described as a stateless hash-based signature scheme. It uses many components from XMSS but works with larger keys and signature to eliminate state. The scheme can be used with different hash functions. The main goal of this paper is to analyze the application of the national standard hash function the scheme of the NIST submission candidate SPHINCS+. The research showed the national standard hash could be applied to the seed randomness generation and hashing the input message. Since Kupyna function returns fixed-size output, its application looks similar to SHA-256 hashes.

Key words: postquantum cryptography; signature scheme; hash function; Kupyna; SPHINCS+; Merkle trees.

3 tab. 1 fig. Ref.: 9 items.

ЗБІРНИК НАУКОВИХ ПРАЦЬ
РАДІОТЕХНІКА
Випуск 198
Російською, українською та англійською мовами

СБОРНИК НАУЧНЫХ ТРУДОВ
РАДИОТЕХНИКА
Выпуск 198
На русском, украинском и английском языках

Коректор Л.І. Сащенко

Підп. до друку 25.10.2019. Формат 60x90/8. Папір офсет. Гарнітура Таймс. Друк. ризограф.
Ум. друк. арк**.** 11,2. Обл.-вид. арк. 10,6. Тираж 300 прим. Зам. № 252. Ціна договір.

Харківський національний університет радіоелектроніки (ХНУРЕ)
Просп. Науки, 14, Харків, 61166.

Оригінал-макет підготовлено і збірник надруковано у ПФ „Колегіум”, тел. (057) 703-53-74.
Свідоцтво про внесення суб’єкта видавничої діяльності до Державного реєстру видавців.
Сер. ДК №1722 від 23.03.2004.