UDC 004.056.53

MARKOVSKYI O.,
RUSANOVA O.,
AL-MRAYAT G.A.J,
KOT O.

ONE APPROACH TO ACCELERATE THE EXPONENTIATION ON GALOIS FIELDS FOR DATA PROTECTION CRYPTOGRAPHIC SYSTEMS

The new approach to accelerate the computational implementation of the basic for a wide range of cryptographic data protection mechanisms operation of exponentiation on Galois Fields have been proposed. The approach is based on the use of a specific property of a polynomial square and the Montgomery reduction. A new method of squaring reduces the amount of computation by 25% compared to the known ones. Based on the developed method, the exponentiation on Galois Fields procedure has been modified, which allows to reduce the amount of calculations by 20%.

Keywords: multiplication operation on Galois fields, cryptographic algorithms based on Galois Fields algebra, Galois Fields exponentiation, Montgomery reduction.

1. Introduction

The dynamic development of the Internet and computer technology has led to the emergence and widespread use of cloud technologies. These technologies provide a wide range of users with access to virtually unlimited computing power, large amounts of memory and modern software. Thus, cloud technologies can significantly increase the capabilities of the widest range of users to solve their scientific and applied problems.

On the other hand, access to these technologies has become more accessible not only to ordinary users, but also to villains, who were among the first to join the opportunities offered by cloud technology [1]. The tasks of selecting keys to existing cryptographic mechanisms for information security are well parallelized and, accordingly, effectively solved on powerful multiprocessor remote computer systems [2]. Thus, the advent of cloud technology has objectively upset the balance between the level of cryptocurrency and the resources available to villains [2]. To counteract this, there is a need to find new ways to increase the level of cryptocurrency, first of all, public key information protection algorithms. Most of these algorithms are based on the mathematical operation of modular exposition, which is performed on large numbers (2048 or 4096).

One possible solution to this problem is to increase the bit size of the keys used in the corresponding cryptographic algorithms. However, such a decision will result in a significant slowdown in the computational implementation of cryptographic protection mechanisms. In particular, doubling the bit size slows down the execution of algorithms eight times [3].

Another solution to the problem of accelerating the computational implementation of cryptographic algorithms with a public key is to move to another algebraic basis, in particular to the algebra of Galois fields [4]. Operations in these fields are performed an order of magnitude faster due to the lack of hyphens. Further acceleration can be achieved through the use of additional resources. To use them, it is necessary to develop new methods aimed at accelerating the execution of the exposure operation in the Galois fields.

Thus, the scientific task of accelerating the execution of the exposure operation in the Galois fields is relevant at the present stage of development of information technology.

2. Problem statement and review of methods for its solution

The tendency to expand the use of exposure to Galois fields in modern mechanisms of cryptographic protection of information stimulates intensive research aimed at accelerating the performance of multiplicative operations on numbers whose bit size far exceeds the bit size of the processor [5].

In the transition in cryptographic using to the algebra of Galois fields from traditional algebra, in order to distinguish operations in each of these algebras use different notation. In particular, in the algebra of Galois fields, the traditional addition is replaced by the addition operation in the Galois fields, which is denoted by the symbol '⊕' is a bitwise operation "Excluding OR" (XOR). In the algebra of Galois fields there is no subtraction operation usual in traditional algebra. The basic multiplication operation in the Galois field algebra consists of two operations: polynomial multiplication and reduction using a base polynom of the field [6]. The polynomial multiplication is denoted by the symbol '⊗', and the reduction operation consists in calculating the remainder of the polynomial division of the result of the polynomial multiplication by the Galois field forming the polynomial P (x) [6]. The operation of calculating the remainder of the polynomial division of the number A by P is denoted as A rem P. The product of the numbers A and B in the Galois fields is denoted as A⊗B rem P. The operation of calculating exponents on Galois fields, is the calculation of the polynomial remainder from the division of the product E of the numbers A by the polynomial field P is denoted as $A|^E$ rem P in contrast to the modular exposition A^E mod M adopted in traditional algebra [6].

The existing methods of exposition on Galois fields are based on two classical algorithms: from the lower and upper bits of the code the exponents $E = \{e_n, e_{n-1}, \ldots, e_0\}$ for any $j \in \{1, 2, \ldots, n-1\}$; $e_i \in \{0, 1\}$. In both mentioned algorithms it is impossible to perform several cycles simultaneously [7]. The advantage of the lower-bit exponential algorithm is the ability to partially parallelize calculations within a single cycle. This allows you to increase the speed of the algorithm by 1.5 times.

Further acceleration of the exponent operation in Galois fields is carried out by reducing the execution time of multiplication in the field [8]. This operation consists of polynomial multiplication and reduction. The operation of polynomial multiplication of n-bit numbers requires $0.5 \cdot n$ logical addition operations and n shift operations to calculate the product. Taking into account that the execution time of the logical addition command is approximately the same as the execution time of the shift command, we can assume that the implementation of polynomial multiplication is determined by the execution time of $1.5 \cdot n$ logical operations [9]. As the main reserve for accelerating multiplication in Galois fields, most researchers consider the reduction operation [10].

The polynomial reduction operation is performed by adding a number corresponding to the forming polynomial to the current remainder. This operation includes determining the position of the highest digit of the current remainder, shifting the code forming the polynomial, logically adding it to the current remainder [11]. Thus, to perform the reduction, you need to perform an average of n bit testing operations, $2 \cdot n$ offset operations (offset code of the forming polynomial and test code containing one unit), as well as $0.5 \cdot n$ logical addition operations. The total average number of logical operations to perform reduction by dividing polynomials is $3.5 \cdot n$.

Further increase in speed is achieved by accelerating the reduction. Most of the known methods are based on the use of precalculations dependent on the unchanging polynomial P [12], which in cryptographic information protection systems is part of the public key and, accordingly, rarely changes.

In acceleration methods based on the use of this property of the forming polynomial, the residues from the division of the codes $2^{n+1}, \ldots, 2^{2 \cdot n}$ by the forming polynomial P(x): $T_1 = 2^{n+1}$ rem P, $T_2 = 2^{n+2}$ rem $P, \ldots, T_n = 2^{2 \cdot n}$ rem P are preliminarily calculated. The calculated codes are stored in the table memory of precalculations. The reduction is reduced to the addition of tabular codes, which correspond to the units in the highest n digits of the code of the polynomial product. Thus, due to the use of precalculations, it is possible to reduce the average number of logical operations to implement the reduction to $1.5 \cdot n$. The total average number of logical operations for multiplication on Galois fields is $3 \cdot n$.

Another way to accelerate the reduction in Galois fields is proposed in [13,14] and its essence is to adapt the Montgomery technology known in traditional algebra to the features of the algebra of Galois fields. Using Montgomery technology, the average number of logical operations for the computational implementation of multiplication in Galois fields was reduced to $2 \cdot n$ rounds [15].

An analysis of both classical Galois finite-field exposition algorithms shows that 2/3 of the computational volume is accounted for by the square operation. Therefore, the most promising way to accelerate these important cryptographic calculations is to conduct research aimed at reducing the computational complexity of squaring in the Galois fields.

3. Purpose and objectives of research

The aim of the study is to accelerate the calculation of the exponent on the finite Galois fields in software and hardware implementation by reducing the number of logical operations required for squaring in the Galois fields.

To achieve this goal, the study solves the following tasks:

- analysis of the features of symmetry of operations when squaring in Galois fields and finding ways to use them to accelerate squaring - the basic operation of exposition in Galois fields;
- development of a method of accelerated squaring in the Galois fields, the difference of which is to eliminate duplication of calculations, thereby reducing the computational complexity;
- development of a modified exposition procedure in Galois fields using the accelerated method of squaring;
- evaluation of the effectiveness of the proposed method of squaring in the Galois fields and exposition in terms of accelerating their computational implementation.

4. The method of accelerated elevation to the square in the Galois fields using the Montgomery reduction

Two-thirds of the computational volume that makes up the Galois field exponentiation, as well as the traditional modular exposition, is the squaring operation [8]. Therefore, it is important to find opportunities to accelerate this dominant component of the implementation of cryptographic mechanisms in the transition from traditional modular exposure to perform this operation in Galois fields.

The property of a polynomial square and the application of the Montgomery reduction can be considered as the main reserves for the acceleration of calculations related to the squaring in the Galois fields.

The property of a polynomial square is that the addition to the square of the number $A = a_{n-1}\cdot 2^{n-1} + a_{n-2}\cdot 2^{n-2} +\ldots+ a_1\cdot 2 + a_0$, where $\forall i\in\{0,1,\ldots,n-1\}$: $a_i \in\{0,1\}$ is reduced to the insertion of "zeros" between the binary digits $a_0, a_1,\ldots,a_{n-1}$ of the number A: $A\otimes A = A|^2 = a_{n-1}\cdot 2^{2\cdot(n-1)} + a_{n-2}\cdot 2^{2\cdot(n-2)} +\ldots+ a_1\cdot 4 + a_0$ [11]. For example, if $A = 9_{10} = 1001_2$, then its polynomial representation has the form: $A(x) = x^3 + 1$. Accordingly, the polynomial square of this number can be represented as: $A(x)\otimes A(x) = (x^3+1)\cdot(x^3+1)= x^6+x^3+x^3+1=x^6+1$. Inserting “zeros” between binary digits gives a similar result: $A|^2 = 1\ 0\ 0\ 0\ 0\ 0\ 1_2 = 65$.

Thus, the first component of squaring in the Galois fields - polynomial multiplication does not require for its implementation any operations other than shifts. Montgomery technology adapted to Galois field algebra can be used to accelerate the computational implementation of the second component, the reduction of a polynomial square [12].

To realize the above possibilities, the following method of squaring in Galois fields is proposed.

There is a number A such that $A = a_{n-1}\cdot 2^{n-1} + a_{n-2}\cdot 2^{n-2} +\ldots+ a_1\cdot 2 + a_0$, and $\forall i\in\{0,1,\ldots,n-1\}$: $a_i \in\{0,1\}$. It is necessary to perform the operation of squaring this number to the square, ie to calculate $A\otimes A|^2$ rem P, where P is the number that corresponds to the forming polynomial of the Galois field: $P = p_n\cdot 2^n + p_{n-1}\cdot 2^{n-1} + p_{n-2}\cdot 2^{n-2} +\ldots+ p_1\cdot 2 + p_0$; $\forall j\in\{0,1,\ldots,n\}$: $p_j\in\{0,1\}$. Montgomery's technology involves the use of an auxiliary polynomial R, $R = 2^n$ for which the multiplicative inversion R^{-1} is determined so that $R\cdot R^{-1}$ rem P=1.

The proposed method involves the following sequence of actions:

1. The counter j cycles is set to zero: $j = 0$.

2. The number B is formed: B: $B = b_{2n-1}\cdot 2^{2n-1} + b_{2n-2}\cdot 2^{2n-2} +\ldots+ b_1\cdot 2 + b_0$ and $\forall k\in\{0,1,\ldots, 2n-1\}$: $b_k = a_{k/2}$, if k mod 2 = 0 and $b_k = 0$ if k mod 2 = 1.

3. If $b_0 = 0$, then proceed to claim 5.

4. To the current value of the code B is added modulo 2 the value of the code P, which corresponds to the forming polynomial of the code: B = B⊕P.

5. A shift to the right by one bit of the value of the code B of the current result: B >> = 1;

6. The unit is added to the cycle counter: $j = j + 1$. If $j < n$, then there is a return to re-execution of claim 3.

Next will be show that as a result of the proposed procedure it obtained the value of the result B, which is equal to $A\otimes A\otimes R^{-1}$ rem P. If we denote by D the polynomial square D = A⊗A, which is obtained by inserting "zeros" between each pair of binary digits of the number A, the value of D is equal to the initial value of B, which is formed in paragraph 2 of the developed procedure. In the process of its implementation to the value of D are added h values of the number P, where $0 < h \le n$. The addition of the numbers P is carried out in such a way that their logical sum with the code D has zeros in n lower digits. That is, the code B′, which is obtained as a result of the above procedure, excluding offsets, can be represented as: B′ = A⊗A ⊕ S, where S is the logical sum h of shifted codes P. Offset of the obtained result B′ by n positions to the right , provided that the lower n bits of B′ are equal to zero, equivalent to the multiplication of B′ by the multiplicative inversion R-1 of the code R = 2n, the multiplication by which is identical to the shift operation to the left by n bits. Thus, the code B obtained as a result of the procedure described above is a reduction of the product: $B'\otimes R^{-1} = (A\otimes A \oplus S)\otimes R^{-1} =(A\otimes A)\otimes R^{-1}\oplus S\otimes R^{-1)}$. In other words, $B = B'\otimes R^{-1}$ rem $P = (A\otimes A)\otimes R^{-1}$ rem $P \oplus S\otimes R^{-1}$ rem P. Due to the fact that the second component of the sum includes as a component of the product the sum of codes P, then the value of its remainder from the polynomial division by P is zero. This means that the obtained, as a result of the proposed and described above procedure is equal to $B = (A\otimes A)\otimes R^{-1}$ rem P, which had to be proved.

The proposed algorithm can be illustrated by the following numerical example.

Let n = 4, $A = 11_{10} = 1011_2$, forming polynomial $P = 19_{10} = 10011_2$, an auxiliary polynomial $R = 2^n = 16$, and its multiplicative inversion is equal to $R^{-1} = 14_{10} = 1110_2$. Indeed, R·R-1 rem P = 16·14 rem 19 = 1. It is necessary to raise to the square of the number A on the Galois field with the forming polynomial P (x) = x4 + x + 1, which is related to the number P = 19: A⊗A rem P = 11⊗11 rem 19 = 9.

Before performing the calculations, according to claim 1 of the above procedure, the counter j cycles is set to zero, and the initial value of the number B is formed from a given number A by inserting zeros: B = 1000101.

The dynamics of transformations of variable B in the process of performing cycles of the above procedure is presented in table 1.

Table 1

Dynamics of transformations of variable B in the process of performing cycles of the proposed procedure of accelerated squaring in the Galois fields

j	The value of the variable B			
	At the beginning of the cycle	b_0	After performing step 4	After performing step 5
0	1000101	1	1000101 ⊕ $\underline{10011}$ 1010110	101011
1	101011	1	101011 ⊕ $\underline{10011}$ 111000	11100
2	11100	0	-	1110
3	1110	0	-	111

The number B obtained as a result of the proposed procedure of accelerated ascent to the square is equal to $A\otimes A\otimes R^{-1}$ rem P = 11⊗11⊗14 rem 19 = 7.

The true value of U of the square A on the Galois field with the forming polynomial P(x)=x^4+x+1, which corresponds to the number P = 19 can be obtained by multiplying B by R=2^n =2^4 = 16: U = 7⊗16 rem 19 = 9.

According to conducted research, the average multiplication time in Galois fields depends on the number of logical addition and shift operations.

In a known variant of multiplication in the fields of the Montgomery Reduction [14], the shift is performed on each of *n* cycles, so is performed *n* times. Operations of logical adding the multiplicand to current result are carried out when the current bit of the multiplier is equal to one. On large length, the probability that the current bit will be a one or zero, is equal to 50%, thus, the multiplicand logical addition will occur only in half of all cycles, that is 0.5·*n* times. Operations of logical adding of Galois fields base polynomial depends on low bit of current result. Thus, in average this operation is executed also 0.5·*n* times [16]. Then, the average total number of logical addition operators consist of *n*. Logical addition and shift operations require approximately the same time to execute. Therefore, it is advisable to calculate all operations together when calculating the time of the algorithm together. And the total number of all operations will be 2·*n*.

In the developed method, the rise time to the square depends on number of shifts and number of logical addition operations. Adding a multiplicand in proposed method was replaced by the previous and disposable insertion of zeros. Shifts occur on each cycle, that is their number *n*. As in a well-known method, the logical addition operation number depends on the low bit of current result. Therefore, the average numbers of such operations is equal to 0.5·*n*. Acceleration occurs due to the exclusion of logical adding of multiplicand. Thus, in the algorithm according to the proposed method, the average total number of operations for square on Galoise field calculation is 1.5·*n*.

Compared to the time of execution of a previously existing multiplication algorithm by Montgomery method [15], the number of operations decreased from 2·*n* to 1.5·*n*, that is, by 25%. The conducted experimental studies were obtained by the theoretical evaluation.

5. Organization of the calculation of the exponent in the Galois field using the proposed method of squaring

As noted above, squaring in the Galois field is about 2/3 of the process of calculating the exponent in the Galois field - the basic operation of a wide class of cryptographic algorithms.

Accordingly, the squaring by the proposed method, which combines the use of a polynomial square and the Montgomery reduction, can be effectively used to accelerate the exposure in the Galois fields. The squaring operation performed by the proposed and described method is hereinafter referred to as KM (Montgomery Square) in contrast to the known multiplication scheme in Galois fields using Montgomery recursion [13 which is denoted as MM.

Modified in this way the exposition procedure on Galois fields, the calculation of $A|^{E}$ rem P involves performing precalculations before the start of cycles of sequential processing of bits of the exponent code. Montgomery technology determines the use of the auxiliary polynomial R(x) and its multiplicative inversion R^{-1}(x). The number R corresponding to the polynomial $R(x) = x^n$ is defined as R = 2^n; accordingly, the multiplicative inversion R^{-1}(x) is correlated with the number R^{-1} such that $R\otimes R^{-1}$ rem P = 1. In addition, the reduction technology according to the Montgomery method involves performing precalculations before exponentiation, namely: calculation G = R rem P = R⊕P and D = $R|^2$ rem P, as well as Z = MM (A, D) = $A\otimes D\otimes R^{-1}$ rem *P*. It is obvious that the values of G, D depend only on the Galois polynomial field, so they are calculated only once and can be used to expose different numbers provided that the Galois polynomial field is constant. The calculation of the number Z precedes each exposure in the Galois fields due to the fact that it depends on A.

Formally, modified as above, the Galois finite field exposure procedure using Montgomery reduction and accelerated squaring consists of the following sequence of actions:

1. The counter of h cycles is set in *n*: *h* = *n* so that it indexes the highest unit digit of the code of the exponent E.

2. Using the developed method of accelerated ascent to the square, the value of the square G is calculated: G = KM (G).

3. If the current h-th bit of the code of the exponent E is equal to one $e_h = 1$, then the multiplication operation with Montgomery recursion obtained in the previous step of the result G on Z: G = MM (Z, G).

4. The value of the cycle counter $h = h$-1 is decremented. If $h \geq 0$, then the return is performed for re-execution of claim 2.

5. The final result is obtained by multiplying in the Galois field using the Montgomery reduction of the obtained value of G per unit: G = MM (G, 1)

The operation of the described modified Galois field exposure procedure using the accelerated squaring based on the Montgomery reduction can be illustrated by the following example. Let it be necessary to calculate $A|^{E}$ rem P, and A = 12, E = 13, and the Galois field-forming polynomial has the form P (x) = $x^4 + x + 1$, ie corresponds to the number P = 19, then n = 4. Then Montgomery technology involves the use of an auxiliary polynomial R(x) = x^4. Its multiplicative inversion $R^{-1} = 14$. It is easy to calculate the value $12|^{13}$ rem 19 = 8 by performing exponentiation by the classical algorithm without using Montgomery reduction.

The values $G = R \oplus P = 16 \oplus 19 = 3$ and $D = R|^2$ rem P = 5 are calculated in advance. Z = MM (A, D) = $12 \otimes 5 \otimes 14$ rem 19 = 7 is calculated immediately before the exposition.

After setting the counter h to the initial value 4 within stage 2 of the procedure, the square of the constant initial value G = 3: G = KM(3) = 3. Since the current digit e_4 of the code of the exponent E is equal to one, ie $e_4 = 1$, the result under item 3 is multiplied by the Montgomery method by the value of Z: G = MM (Z, G) = MM (3,7) = 7. Then decreases by one the value of the counter h = 3 and, since it is not equal to one, the return for re-execution of paragraph 2.

In this paragraph 2 is the elevation to the square of the value obtained in the previous cycle G: G = KM (7) = 2. Since $e_3 = 1$, it is performed in paragraph 3, in which the result is multiplied by Z: G = MM(Z, G) = MM(2,7) = 11. Again subtract the unit from the counter h and return to claim 2.

When h = 2 is raised to the square of the result: G = KM (11) = 7. Since $e_2 = 0$, then paragraph 3 is skipped and decrements the value of the counter, resulting in h becomes equal to 1. Accordingly, the return to re-execution is realized item 2 within which the previously obtained result is squared: G = KM (7) = 2. Since the least significant bit of the exponent is equal to one, ie $e_1 = 1$, the multiplication is performed: G = MM (Z, G) = MM(2, 7) = 11. Then decreases by one the value of h, which becomes equal to zero. This means that the exposure cycles in the Galois fields are complete. Finally, item 5 is performed - correction of the obtained result G = 11 by multiplying it by one: G = MM (G, 1) = MM (11,1) = 8. The obtained value corresponds to the true value $12|^{13}$ rem 19 = 8.

The exposition operation consists of n cycles. In each cycle, the developed procedure of accelerated squaring to the square is performed, which is realized, on average, in $1.5 \cdot n$ logical operations. In addition, on average, $0.5 \cdot$n multiplication operations are performed on Galois fields with Montgomery reduction, each of which, on average, requires $2 \cdot n$ logical operations to implement. In general, the average number of logical operations required for exposition on Galois fields using the proposed method is $2.5 \cdot n^2$

Based on the fact that in the known scheme [11] of exponentiation on Galois fields with Montgomery reduction, the average number of logical operations is $3 \cdot n^2$ we can conclude that the proposed method can speed up the process of exposure to Galois fields by 20%, due to saving $0.5 \cdot n^2$ logical operations. With a typical value of n = 2048 for practical applications, this is 254,000 operations. Experimental studies have shown that the real acceleration of exposure in Galois fields is in the range of 18-22%.

5. Conclusions

As a result of research aimed at accelerating the execution of the basic for a wide range of cryptographic mechanisms of the exponentiation operation in Galois fields, a new method of accelerated squaring using Montgomery reduction is proposed.

The developed method is based on the properties of a polynomial square in Galois fields and allows to reduce by 25% the number of logical operations in comparison with the use for calculation of the multiplication square in Galois fields with Montgomery reduction. Based on the developed method, a modified exposition procedure on Galois fields with Montgomery reduction is proposed. Theoretically and experimentally it is shown that the use of a modified procedure can reduce by 20% the number of logical operations and accordingly accelerate the exposure.

The proposed solutions not only speed up the calculations, but also provide their simplification, which determines their focus primarily on the hardware implementation of cryptoprocessors.

References

1. Boroujerdi N. Cloud Computing: Changing Cogitation about Computing/ N. Boroujerdi, S. Nazem // IJCSI International Journal of Computer Science Issues. – Vol. 9. – Issue 4. – 2012. – №3. – PP. 169-180.
2. Armbrust M. A view of cloud computing / M. Armbrust, A. Fox, R. Griffith, R. Katz, A.A. Konwinski // International Journal Computer Technology.-2013.- No.4.- PP.50-58.
3. Schneier B. Applierd Cryptography. Protocols, Algorithms and Source Codes in C/ B.Schneier // John Wiley&Son,Inc.:N.Y. – 2009.- 816 p.
4. Zenzin O.S. Standard of cryptographic data protection of the XXI century –AES. Finite fields / O.S.Zenzin, M.F. Ivanov // M.: Kudic-Obraz.- 2002.- P.174.
5. Markovskyi O.P. Utilization of Galois Field algebra for zero knowledge identification and audentification of remote users /O.P. Markovskyi, Zahariudakis Lifteris, Maksimyk V.R. // Electronic modeling – 2017.- Vol.6.-No.39.-P.33-45/.
6. Postnikov M.M. Galois field theory / M.M. Postnikov //Sankt-Petersburg: BXV-Peterburg Press.- 2011.- 411 P.
7. Fitzpatrick P. Algorithm and Architecture for a Galois Fiels multiplicative Arithmetic Processor./ P. Fitzpatrick, Popovici E. M. // IEEE Trans. on Information Theory.- 2003 - Vol.49, - № 12, - P. 3303-3307.
8. Samofalov K.G. The method of accelerated implementation of exponentiation on Galois fields for data protection systems / K.G. Samofalov, O.P. Markovskyi, A.S. Sharshakov // Problems of informatization and control. NAU.- 2011.- Vol.2- No.33.-P.143-151.
9. Markovskyi O.P. Technology of digital signature DSA based on Galois Fields Arithmetics // O.P. Markovskyi, Saidreza Mehmali, G.V.Isachenko // Herald of of National Technical University of Ukraine “KPI” Informatica, control and computer technic.- 2012.- № 55. - P. 34 — 41.
10. Kalmikov I.A. Development of the method of nonlinear data encryption using an exponentiation in Galois Fields / I.A. Kalmikov, E.S. Stepanova, K.T. Titcherov // Modern Scientific Technologies.- 2019.- № 9. - P.84—89.
11. Osadchyy V. The Order of Edwards and Montgomery Curves «, / V.Osadchyy // WSEAS Transactions on Mathematics, - 2020.- Vol. 19.- № 25, - P. 253-264.
12. Wu H. Finite field multiplier using redundant representation./ H. Wu, M.A. Hasan, I.F. Blake, S.Gao // IEEE Trans. Computers. -2002 - Vol.51,- № 5.- P. 1306-1316.
13. Kot O.S. Organization of speed up exponentiation on Galois Field using Montgomery Reduction / O.S. Kot, O.P. Markovskyi // Almanac Science.-2020.- № 3 (36).- P.34-37.
14. Markovskyi Oleksandr. The Employment of Montgomery reduction for acceleration of exponent on Galoise fields calculation / O. Markovskyi, V. Masimyk, O.Kot // Proceeding of International Conference on Security, Fault Tolerance, Intelligence” (ICSFTI2020), 13-14 may 2020, Kyiv .- P.44-49.
15. Haches G. Montgomery multiplication with no final subtraction./ G. Haches, J.J. Quisquater // Cryptographic Hardware and Embedded System- CHES’2000. LNCS-1965, Springer-Verlag. — 2000.- P. 293-301.
16. Elfard S. Justification of Montgomery Modular Reductions / S. Elfard //Advanced Computing.- 2012. - № 11. – P.41-45.