N E W S
OF THE NATIONAL ACADEMY OF SCIENCES OF THE REPUBLIC OF KAZAKHSTAN
SERIES OF GEOLOGY AND TECHNICAL SCIENCES
ISSN 2224-5278
Volume 4, Number 436 (2019), 181 – 187 **https://doi.org/10.32014/2019.2518-170X.113**

UDC 004.056
IRSTI 81.93.29

M. Kalimoldayev[1], S. Tynymbayev[1], S. Gnatyuk[2], S. Khokhlov[3], M. Magzom[1], Y. Kozhagulov[3]

[1]Institute of Information and computational technologies, Almaty, Kazakhstan,
[2]National aviation university, Kyiv, Ukraine,
[3]al-Farabi Kazakh national university, Almaty, Kazakhstan.
E-mail: mnk@ipic.kz, s.tynym@mail.ru, s.gnatyuk@nau.edu.ua,
skhokh88@gmail.com, magzomxzn@gmail.com, kazgu.kz@gmail.com

MATRIX MULTIPLIER OF POLYNOMIALS MODULO ANALYSIS STARTING WITH THE LOWER ORDER DIGITS OF THE MULTIPLIER

Abstract. The advantage of an unconventional data encryption system using non-positional polynomial number systems (NPNS), known as polynomial residue number system, is considered. When hardware and software-hardware implementations of cryptosystems based on the NPNS, circuit solutions are needed multipliers of polynomials modulo an irreducible polynomial. In this paper, we present the design of matrix multiplier of polynomials modulo irreducible polynomial. The correct operation of the proposed multiplier is verified by implementing it on the FPGA of the company Xilinx of model Artix 7. In conclusion, a comparative analysis of the matrix multipliers considered is given in terms of time parameters and hardware costs for their implementation.

Keywords: non-positional polynomial number system, partial residual formers, modulo two.

Introduction. The development of information and communication systems increases the need to ensure data protection. At the same time, due to the specifics of the application, restrictions on physical size and power consumption, individual devices have small computational resources [1]. For devices with limited resources, standard cryptographic algorithms may be too complex, too slow, or too energy intensive. The issues of creating and applying methods to improve the efficiency of cryptosystems with hardware implementation remain relevant [2, 3].

Searching for ways to improve the efficiency of software and hardware calculations, methods for detecting and correcting errors and creating highly reliable computer systems, research is being carried out in the field of non-positional notation systems, such as the residual number system (RNS). In the classical positional number system, the value of each digit in the designation of a number depends on its position. In non-positional numeration systems, a large-digit integer in positional notation is represented as a sequence of several positional numbers of small bitness. These numbers are the residues of dividing the original number by moduli of RNS.

The basis for creating the proposed models of cryptosystems [4-9] are non-traditional encryption systems and digital signatures. These systems are developed on the basis of an algebraic approach using non-positional polynomial number systems (NPNS), known as polynomial RNS.

Improving the efficiency of the hardware implementation of these systems is provided by the rules of the NPNS, in which all arithmetic operations can be performed in parallel using the base modules of the NPNS. The features of the NPNS give significant advantages over the positional number system when performing modular operations of addition, subtraction and multiplication. This is especially true if large-digit numbers act as operands [10].

In non-positional cryptosystems, the cryptographic strength of the encryption algorithms and digital signature generation, which is characterized by a complete secret key, is used as a criterion for cryptographic strength. This key depends not only on its length, but also on the selected system of the polynomial bases of the NPNS, as well as on the number of all possible permutations of the bases in the system.

With increasing order of irreducible polynomials with binary coefficients, their number is rapidly growing. In this regard, a wide choice of polynomial bases is possible.

In [4], arithmetic of non-positional number systems with polynomial bases and its applications to problems of increasing reliability were developed. It is shown that the algebra of polynomials over a field modulo an irreducible polynomial over this field is a field and the representation of a polynomial in non-positional form is the only one (an analogue of the Chinese remainder theorem for polynomials). The rules for performing arithmetic operations in the NPNS and restoring a polynomial from its residues are also defined.

The implementation of cryptosystems based on the NPNS can be implemented in software, hardware or software-hardware methods. The main advantage of the software implementation is their flexibility, which makes it possible to quickly rebuild cryptoalgorithms, the main disadvantage is a significantly lower speed compared to the hardware implementation. Software and hardware implementation of cryptosystems combines the advantages of software and hardware implementation. With hardware and software-hardware implementations of cryptosystems based on the NPNS, the central unit is the multipliers of polynomials modulo an irreducible polynomial, where repeated routine calculations are performed on encryption and decryption of data. Therefore, the development of devices for multiplying polynomials modulo an irreducible polynomial is relevant. In such multipliers, the multiplier is full A(x), having degree *m*, the binary image of which is part of the plaintext, the multiplier is polynomial B(x), having degree *m*, which is the key for encrypting the polynomial A(X). The module is an irreducible polynomial P(x), which is randomly selected from the set of irreducible polynomials with degree *m*. After multiplying modulo polynomials, we obtain the polynomial R(x) which is part of the ciphertext.

When decrypted, the polynomial of the ciphertext R(x) acts as a multiplicand, and the multiplier is the reverse key $B^{-1}(x)$. After multiplying R(x) by $B^{-1}(x)$ modulo P(x), we get a part of the plaintext - the polynomial A(x).

There are two ways to multiply polynomials modulo an irreducible polynomial. In the first method of multiplying polynomials, multiplication begins with an analysis of the higher order of the multiplier. At the same time, in each multiplication step, the next partial remainder is shifted one digit to the left. And in the second method, multiplication begins with an analysis of the lower order of the multiplier with a shift of the next partial remainder by one digit towards the older one.

The matrix multiplier of polynomials modulo an irreducible polynomial, where multiplication begins with an analysis of the higher order of the multiplier was considered in [11].

The matrix multiplier scheme of polynomials modulo an irreducible polynomial, where multiplication begins with an analysis of the lower order bits of the multiplier. In the matrix multiplier of polynomials modulo is performed in N-1 stages according to the number of digits of the multiplier. Each stage consists of three sub-steps. In the first sub-step, the partial remainder r_i is calculated by modifying twice the previous partial residual $2r_i$ modulo, i.e. $r_i=2r_{i-1}$modP. In the second sub-step, the partial residues r_i logical are multiplied by the corresponding bits of the b_i of the multiplier, starting with the lower order digit. In the third sub-step, an intermediate residue R_i is formed by modifying the sum $(r_i*b_i)+R_{i-1}$ modulo.

Figure 1 shows a block diagram of the matrix multiplier of polynomials modulo an irreducible polynomial, where multiplication begins with the analysis of the lower digits of the polynomial multiplier with a shift of partial residues by one bit in the direction of the higher digit. The multiplier consists of four blocks: 1 - the block is a block of registers, which includes the register of the module P(x) and the register of the multiplier B(x), the block of the PRS_2 ($PRS_1 \div PRS_{N-1}$) , block of circuits AND 3 ($AND_1 \div AND_{N-1}$), block of adders modulo two ($MA_{21} \div MA2_{N-1}$), delay lines 5.

Consider the operation of the device. The signal "START", which is fed into the circuit through input 6 to the register Pr P(x) from input 7, the binary coefficients of the polynomial P(x) are received – the module, and to the register Pr B(x) from input 9, the binary coefficients of the polynomial B(x) is a multiplier. Binary coefficients of the irreducible polynomial P(x) - module from the outputs of the register P(x) are fed to the first inputs of the formers $PRS_1 \div PRS_{N-1}$. The multiplicand A(x) (input 8) with a shift

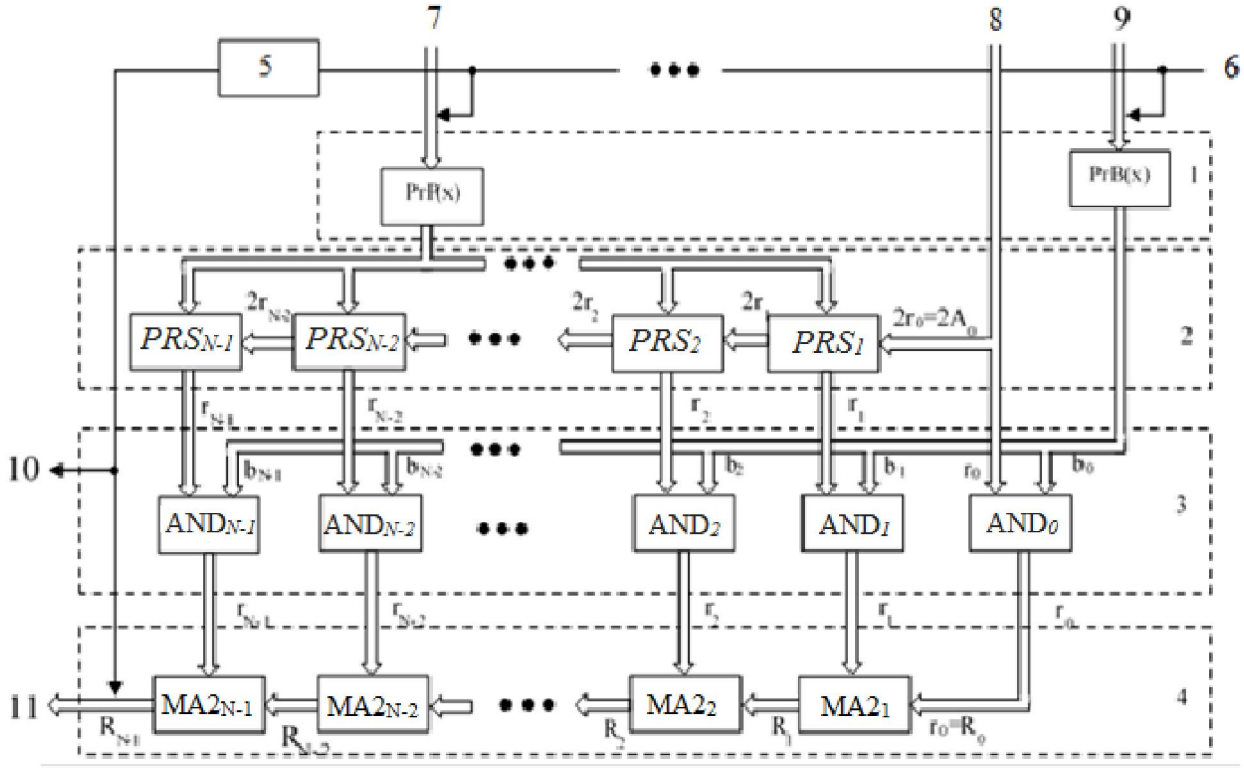

Figure 1 – Block diagram of the matrix multiplier of polynomials modulo an irreducible polynomial, where multiplication begins with the analysis of the lower order of the polynomial – multiplier

by one bit in the direction of the higher discharge, i.e. 2*A(x) = 2*r_o is fed to the second inputs PRS_1 and without shift is transmitted to the information inputs of the AND_0 circuit, the value of the b_o bit is fed to its control input from the output of the B(x) register. At the outputs of PRS_1, a partial residual $r_1 = 2 * r_o$ *mod P(x)* is formed, which is fed with a shift by one bit to the second input of the PRS_2 and without a shift is transmitted to the information inputs of the AND_1 circuit, to the control input of which the bit b_1 value is fed from Pr B(x). When $b_1 = 1$, the value of r_1 from the output of AND_1 is transmitted to the first inputs of the adder modulo two $MA2_1$, and the second information inputs of which are fed the value $r_0 = R_0 = A(x)$ and the intermediate balance is formed at the output of the $MA2_1$ by calculating $R_1 = r_1 \oplus r_0$, which is transmitted to the second inputs of the $MA2_2$. The PRS_2 having received the value $2*r_1$ from the output of the PRS_1 at its output forms a partial residual r_2, which with a shift of one digit to the left is transmitted to the input of the PRS_3 and without a shift to the information inputs of the AND_2 circuit. To the control input of which is fed bit b_2 from the register B(x). When $b_2 = 1$, the value of r_2 is transmitted to the information inputs of the $MA2_2$, the other information input is supplied with the value R_1 from the outputs of the $MA2_1$ and forms the intermediate remainder R_2, which is transmitted to the information inputs of the $MA2_3$.

Further, partial residues $r_3, r_4, \dots, r_{N-1}$ and intermediate residues $R_3, R_4, \dots, R_{N-1}$ are formed in the same way. After the formation of the intermediate residue R_{N-1}, the information output $MA2_{N-1}$ forms the result, which by the signal 10 ("end of operation") outputs it through the output of the device 11.

Figure 2 shows the block diagram of the PRS_i, which consists of a modulo-two adder and an MS multiplexer. The multiplexer, in turn, consists of AND_1, AND_2 schemes and *OR* scheme. The first partial adder r_{i-1} is fed to the first inputs of the adder with a shift by one bit in the direction of the higher digit, which is equivalent to multiplying r_{i-1} by two. In addition, the value of $2 * r_{i-1}$ is also fed to the information inputs of the AND_1 circuits. The information inputs of the $И_2$ scheme are fed with the result of the addition of $2 * r_{i-1} \oplus P(x)$. Switching of values $2 * r_{i-1}$ from input AND_1 or AND_2 depends on the values of the most significant digit (S_r) of the value of doubled partial residue $2 * r_{i-1}$. At S_r=1, the output of the MS circuit through the AND_2 and *OR* circuits is the result of the sum modulo two result $(2 * r_{i-1} \oplus P(x))$,

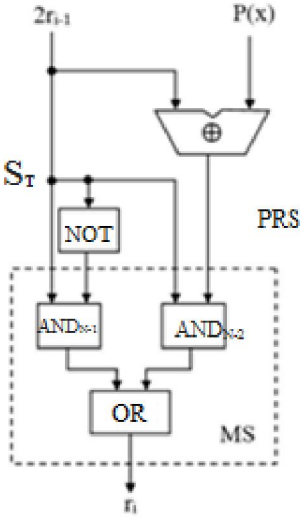

Figure 2 –
The structure of the partial residual shaper

and at S_T=0, this signal passing through the inverter is NOT fed to input circuit AND_1, allowing the passage of the values $2 * r_{i-1}$ to the output MS. Thus, for values of $2 * r_{i-1} < P(x)$ (while S_T=0), the output of the *PRS* circuit produces the value $r_i = 2 * r_{i-1}$, and for values of $2 * r_{i-1} > P(x)$ (at that S_T=1) at the output PRS_{N-1} the value $r_i = 2 * r_{i-1} \oplus P(x)$ is shaped.

The $MA2_i$ is an modulo-two n-bit adder, where the operation $R_i = r_i + R_{i-1}$ is performed.

Consider the work of the multiplier on a specific example.

Let $P(x) = x^5 + x^3 + 1$, the binary image of P(x): 101001;

$A(x) = x^4 + x + 1$, binary image A(x): 10011;

$B(x) = x^4 + x^2 + x + 1$, binary image B(x): 10111.

The order of calculation of $R = [A(x) * B(x)] mod\ P(x)$ is given in table.

To implement the above example on programmable logic integrated circuits (FPGA), consider the logical chain of operations performed. The binary image values of the polynomials A(x) and B(x) are fed to the input of the programmable logic integrated circuit. At the output of $MA2_{n-1}$, the result of the multiplication R=[A(x)*B(x)]mod(P(x)) is formed. The current values of R_1 , R_2 … R_{n-2} is formed at the outputs of the corresponding adder circuits modulo $MA2_1$, $MA2_2$ … $MA2_{n-2}$. The first step is set separately, according to the description of the zero stage from Table 1, where r_0 takes the values of the input signal *A*, also in the $MA2_i$ block, the first value R_0 is equal to r_0. From the next step, system operations are performed according to the above description, using the modulo multiplier operator. To check the correctness of the proposed algorithm on the integrated circuit, a time diagram was built on the FPGA of the Artix 7 model shown in Figure 3. On the time diagram (aigure 3), one can observe the results of the calculation on each clock signal whose numerical values correspond to the values shown in Table 1. The program is written in the Verilog language, consisting of a procedural block, a register (data type) of calculation parameters, a ternary operator, a shift operator, and a continuous assignment operator [12-14].

The number of used FPGA resources of the Artix 7 model does not exceed 1%: the number of registers is 44 out of 126,800, the number of logical cells is 82 out of 63 400. The results obtained confirm the correctness of the proposed algorithm and the proposed theory on the FPGA.

In conclusion, a comparative analysis of the algorithm with a matrix multiplier with the results of [11] was carried out. The comparison is carried out according to the time of multiplication – T_{mult} and the hardware cost Q_{mult} required to build them.

From figure 1, it is easy to determine the components of the circuit from the total delays at which the multiplication time can be determined: PRS_1 χ PRS_{N-1} – AND AND_{N-1} – MA_{N-1}. Then the multiplication time can be determined by the following relationship:

$$T_{mult}^{SM} = N\text{-}1(\tau_{PRS}) + \tau_{AND} + \tau_{MA2} \quad (1)$$

where τ_{PRS} is the amount of delay on one partial residual shaper; τ_{MA2} – delay time on the adder modulo two; τ_{AND} is the delay time in the circuit AND_{N-1}.

Order of calculation of R

Stages	PRS_i (r_i)	$b_i * r_i$	$MA2_i$ (R_i)
0	r_0=10011	b_0*r_0=10011	R_0= r_0=10011
1	r_1=2 r_0 *mod* P(x) $2r_0 = 100110$ $\oplus$ $P(x) = 101001$ $r_1 = 001111$	b_1*r_1=01111	R_1= $r_1 \oplus R_0$ R_0=10011 $\oplus$ $r_1 = 01111$ $R_1 = 11100$
2	r_2=2 r_1 *mod* P(x) $2r_1 = 011110$ $\oplus$ $P(x) = 101001$ $r_2 = 011110$	b_2*r_2=011110	R_2= $r_2 \oplus R_1$ R_1=11100 $\oplus$ $r_2 = 11110$ $R_2 = 00010$
3	r_3=2 r_2 *mod* P(x) $2r_2 = 111100$ $\oplus$ $P(x) = 101001$ $r_3 = 010101$	b_3*r_2=0	R_3=R_2=00010
4	r_4=2 r_3 *mod* P(x) $2r_3 = 101010$ $\oplus$ $P(x) = 101001$ $r_4 = 000011$	b_4*r_4=00011	R_4=$r_4 \oplus R_3$ 00011 $\oplus$ 00010 $R = R_4 = 00001$

Check: $(x^4 + x + 1)(x^4+x^2 + x + 1) = x^8 + x^6 + x^3 + 1$

$$\begin{array}{r} x^8 + x^6 + x^3 + 1 \\ \oplus \\ x^8 + x^6 + x^3 \\ \hline 00001 \end{array} \qquad \begin{array}{|r} \\ \\ x^5 + x^3 + 1 \\ \hline x^3 \end{array}$$

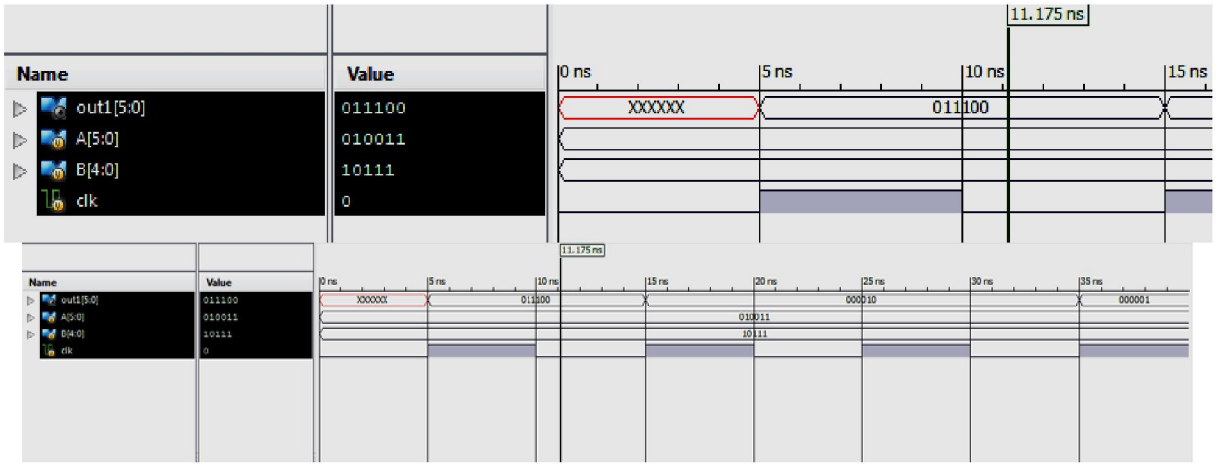

Figure 3 – Timing coding diagram for a 5-bit binary information message in an Artix 7 FPGA

From figure 1, it is also not difficult to determine the ratios, with which you can calculate the hardware cost of the multiplier:

$$Q_{mult}^{SM} = N\text{-}1(Q_{PRS+} Q_{MA2}) + NQ_{AND} \quad (2)$$

where Q_{PRS} - the cost of logic circuits for building one partial residual shaper; Q_{MA2} - the cost of logic circuits for building one adder modulo two; NQ_{AND} - the cost of N logic circuits I.

From the matrix multiplier scheme of polynomials modulo an irreducible polynomial, where multiplication begins with an analysis of the most significant bits [10], it is possible to determine the route of the input signal, which determines the maximum delay, i.e. multiplication time of polynomials, where multiplication begins with the analysis of the most significant digit: $AND_0 - MA2_1 - PRS_1 - MA2_2 - - PRS_2 \chi MA2_{N-2} - PRS_{N-2} - MA2_{N-1} - PRS_{N-1}$

then:

$$T_{mult}^{BG} = \tau_{AND} + N\text{-}1(\tau_{MA2} + \tau_{PRS}) \quad (3)$$

The magnitude of the hardware costs can be determined from the following relationship:

$$Q_{mult}^{BG} = NQ_{AND} + N\text{-}1(Q_{MA2} + Q_{PRS}) \quad (4)$$

From relations (2) and (4) it can be seen that the considered matrix multipliers are equal in hardware costs, i.e. $Q_{mult}^{SM} = Q_{mult}^{BG}$ and they differ in speed.

Let us consider in more detail the components of formulas (1) and (2). From figure 2 that

$\tau_{PRS} = \tau_{MA2} + \tau_{MS}$; in turn, $\tau_{MA2} = 3\,\tau_{L3}$ and $\tau_{MS} = 2\tau_{L3}$;

where, τ_{L3} - the delay time on the logical elements AND-NOT, OR-NOT

Then $T_{mult}^{SM} = N\text{-}1\,(3\tau_{LE} + 2\,\tau_{LE}) + \tau_{LE} + 3\tau_{LE} = N\text{-}1(5\tau_{LE}) + 4\tau_{LE} = N5\tau_{LE} - 5\tau_{LE} + 4\tau_{LE}$

$$T_{mult}^{SM} \approx N5\tau_{LE}. \quad (5)$$

$$T_{mult}^{BG} = N\text{-}1(8\tau_{LE}) + \tau_{LE} \approx N7\tau_{LE}. \quad (6)$$

From the relation (5) and (6) it is seen that with the same hardware costs of the matrix multiplier polynomials modulo an irreducible polynomial, where multiplication begins with the lower order of the multiplier has a significant advantage in speed.

М. Н. Калимолдаев[1], С. Т. Тынымбаев[1], С. Гнатюк[2], С. А. Хохлов[3], М. М. Мағзом[1], Е. Т. Кожагулов[3]

[1]Ақпараттық және есептеуіш технологиялар институты, Алматы, Қазақстан,
[2]Ұлттық авиациялық университеті, Киев, Украина,
[3]фль-Фараби атындағы Қазақ ұлттық университеті, Алматы, Қазақстан

КӨБЕЙТКІШТІҢ КІШІ РАЗРЯДЫНАН БАСТАП ТАЛДАЙТЫН МОДУЛІ БОЙЫНША ПОЛИНОМДАРДЫҢ МАТРИЦАЛЫҚ КӨБЕЙТКІШІ

Аннотация. Қалдықты кластың полиномдық жүйесі ретінде белгілі, есептеудің бейпозициялық полиномдық жүйесін (ЕБПЖ) пайдалану арқылы мәліметтерді шифрлаудың дәстүрлі емес жүйесінің артықшылықтары қарастырылды. ЕБПЖ негізінде криптожүйенің аппараттық және бағдарламалық-аппараттық іске асырылуы кезінде келтірілмейтін полиномның модулі бойынша полиномдардың көбейткіштерінің сұлбалық шешімі қажет. Осы жұмыста мәліметтерді шифрлап және шифрын ашып оқуға мүмкіндік беретін, модулі бойынша полиномдар көбейткішінің матрицалық сұлбасы келтірілген. Ұсынылған көбейткіштің жұмыс істеуінің дұрыстығы Xilinx фирмасының Artix 7 моделі негізіндегі бағдарламаланатын логикалық интегралдық сұлбасында (БЛИС) жүзеге асыру арқылы тексерілді. Қорытындысында қарастырылған матрицалық көбейткіштердің жүзеге асырылуы үшін қажетті аппараттық шығыны және уақыттық параметрлеріне байланысты салыстырмалы талдау келтірілген.

Түйін сөздер: есептеудің бейпозициялық полиномдық жүйесі, жартылай қалдықтарды қалыптастырғыштар, модулі екі бойынша сумматор.

М. Н. Калимолдаев[1], С. Т. Тынымбаев[1], С. Гнатюк[2], С. А. Хохлов[3], М. М. Мағзом[1], Е. Т. Кожагулов[3]

[1]Институт информационных и вычислительных технологий, Алматы, Казахстан,
[2]Национальный авиационный университет, Киев, Украина,
[3]Казахский национальный университет им. аль-Фараби, Алматы, Казахстан

МАТРИЧНЫЙ УМНОЖИТЕЛЬ ПОЛИНОМОВ ПО МОДУЛЮ С АНАЛИЗОМ НАЧИНАЯ С МЛАДШИХ РАЗРЯДОВ МНОЖИТЕЛЯ

Аннотация. Рассматривается преимущество нетрадиционной системы шифрования данных с использованием непозиционных полиномиальных систем счисления (НПСС), известный как полиномиальные системы остаточных классов. При аппаратной и программно-аппаратной реализаций криптосистем на базе НПСС необходимы схемные решения умножители полиномов по модулю неприводимого полинома. В данной работе приводиться матричная схема умножителя полиномов по модулю, которая позволяет шифровать и расшифровать данных. Правильность функционирование предложенного умножителя проверено путем реализации его на ПЛИС фирмы Xilinx модели Artix 7. В заключении дается сравнительный анализ рассмотренных матричных умножителей с точки зрения временных параметров и аппаратных затрат для их реализации.

Ключевые слова: непозиционная полиномиальная система счисления, формирователи частичных остатков, сумматор по модулю два.

Information about authors:

Kalimoldayev Maksat, Director general of Institute of Information and computational technologies, Doctor of sciences, professor, academician member of the National academy of science of the Republic of Kazakhstan, Almaty, Kazakhstan; mnk@ipic.kz; https://orcid.org/0000-0003-0025-8880

Tynymbayev Sakhybay, Chief researcher, Candidate of Technical Sciences, Institute of Information and computational technologies, Almaty, Kazakhstan; s.tynym@mail.ru; https://orcid.org/0000-0002-9326-9476

Gnatyuk Sergiy, Doctor of sciences, Associate Professor, Leading Researcher in Cybersecurity R&D Lab, Executive Secretary of Ukrainian Scientific Journal of Information Security, Scientific Adviser of Engineering Academy of Ukraine, IEEE Member, National aviation university, Kyiv, Ukraine; s.gnatyuk@nau.edu.ua; https://orcid.org/0000-0003-4992-0564

Khokhlov Serik, Lead researcher, PhD, Lecturer of Department of Physics and Technology, al-Farabi Kazakh national university, Almaty, Kazakhstan; skhokh88@gmail.com; https:/orcid.org/0000-0001-5163-508X

Magzom Miras – Senior researcher, PhD, Institute of Information and computational technologies, Almaty, Kazakhstan; magzomxzn@gmail.com; https://orcid.org/0000-0002-9380-1469

Kozhagulov Yeldos, Lead researcher, PhD, Lecturer of Department of Physics and Technology, al-Farabi Kazakh national university, Almaty, Kazakhstan; kazgu.kz@gmail.com; https://orcid.org/0000-0001-5714-832X

REFERENCES

[1] McKay K.A., Bassham L., Turan M.S. (2017) Report on lightweight cryptography, https://doi.org/10.6028/NIST.IR.8114

[2] Mouha N. (2015) The design space of lightweight cryptography. P. 1-19.

[3] William J. Buchanan and Shancang Li and Rameez Asif. (2017) Lightweight cryptography methods. https://doi.org/10.1080/23742917.2017.1384917

[4] Biyashev R.G. (1985) Development and research of end-to-end reliability enhancement methods in data exchange systems of distributed automated control systems: diss. Dr. technical sciences: 328 p.

[5] Amerbayev V.M., Biyashev R.G. (2005) Nysanbaeva S.E.E.Application of non-positional number systems in cryptographic protection of information // Izv. NAS RK. Ser. phys.-mat. sciences. 2005. N 3. P. 84-89.

[6] Biyashev R.G., Nysanbaeva S.E. (2012) Algorithm for generating electronic digital signature with the ability to detect and correct errors // Cybernetics and system analysis. P. 14-23.

[7] Biyashev R.G., Nysanbaeva S.E., Kapalova N.A. (2013) Development of cryptographic information protection systems with specified characteristics // Problems of Informatics. Novosibirsk. N 2(19). P. 30-36.

[8] Biyashev R.G., Nysanbaeva S.E., Begimbayeva Ye.Ye., Magzom M.M. (2015) Building modified modular cryptographic systems // International Journal of Applied Mathematics and Informatics. Vol. 9. P. 103-109.

[9] Nysanbayeva S.E., Magzom M.M. (2016) Model of an unconventional encryption algorithm based on nested networks of Feistel // Vestnik KazNTU. N 4.

[10] Akushsky I.Ya., Yuditsky D.I. (1968) Machine arithmetic in residual classes. M.: Soviet Radio. P. 440.

[11] Kalimoldayev M.N., Tynymbayev S., Gnatyuk S.A., Ibraimov M.K., Magzom M.M. (2019). The device for multiplying polynomials modulo an irreducible polynomial // News of National academy of sciences of the Republic of Kazakhstan. Vol. 2, N 434. P. 199-205. https://doi.org/10.32014/2019.2518-170X.55

[12] Zhanabaev Z.Zh., Kozhagulov Y.T., Zhexebay D. (2016) FPGA implementations of scale invariant models of neural networks // Turkish Journal Of Electrical Engineering & Computer Sciences (Turk J Elec Eng & Comp Sci), 24. P. 5090-5099. doi:10.3906/elk-1504-204