Devices for Modularizing Numbers into a Square in Applications based on Public-Key Cryptography

Sakhybay Tynymbayev[1], Sergiy Gnatyuk[2, 3, 4], Margulan Ibraimov[5], Timur Namazbayev[5], and Assel Mukasheva[6]

[1] *International University of Information Technology, 34/1 Manas str., Almaty, 050000, Kazakhstan*
[2] *National Aviation University, 1 Liubomyra Huzara ave., Kyiv, 03058, Ukraine*
[3] *State Scientific and Research Institute of Cybersecurity Technologies and Information Protection, 3 Maksyma Zaliznyaka str., Kyiv, 03142, Ukraine*
[4] *Yessenov University, 32 microdistrict, Aktau, 130000, Kazakhstan*
[5] *Al-Farabi Kazakh National University, 71 Al-Farabi ave., Almaty, 050040, Kazakhstan*
[6] *Kazakh-British Technical University, 59 Tole Bi str., Almaty, 050000, Kazakhstan*

Abstract

Public-key cryptography solves the problem of key distribution, which is inherent in symmetric cryptography. However, there are problems associated with the high resource consumption of such crypto algorithms, which necessitates new methods to increase speed. The work is dedicated to the development of devices that perform modular square-building, which is one of the basic operations in the implementation of public-key cryptosystems. The resolution of the implementation approach is being analyzed. The method is considered where the multiplication of a number by bits thereof is carried out starting from its senior digits. Two variants are proposed for the implementation of the central unit of the Partial Residue Shaper (PRF). The workability of the developed devices has been tested on the FPGA Artix-7 using the Verilog hardware description language.

Keywords

Public key cryptography, hardware implementation of cryptosystems, device, modular square construction, partial residue shaper.

1. Introduction

Cryptographic security is one of the most reliable ways to address data security problems in computer systems and networks. Cryptographic protection ensures the conversion of open text into micro text by encrypting source text using cryptographic algorithms [1, 2].

In modern cryptosystems, asymmetric encryption [3] is widely used. This is because public-key cryptosystems have potentially high security as compared to symmetric (one key) cryptosystems with a private key: there is no need to transfer and authenticate secret keys. A disadvantage of public-key cryptosystems is their low speed, as encryption and decryption procedures use much more complex and cumbersome mathematical calculations over large numbers [4–7]. Encryption can be done with software, hardware, and hardware [8].

Hardware encryption has several significant advantages over software encryption [9]:

- Hardware encryption tools are faster (hardware implementation of any algorithm, including cryptographic algorithm, provides faster action than software implementation).
- Encryption equipment is easier to physically protect from external intrusion than software.

CPITS-2024: Cybersecurity Providing in Information and Telecommunication Systems, February 28, 2024, Kyiv, Ukraine
EMAIL: s.tynym@gmail.com (S. Tynymbayev); s.gnatyuk@nau.edu.ua (S. Gnatyuk); Margulan.Ibraimov@kaznu.kz (M. Ibraimov); t.namazbayev@gmail.com (T. Namazbayev); mukasheva.a.82@gmail.com (A. Mukasheva)
ORCID: 0000-0002-9326-9476 (S. Tynymbayev); 0000-0003-4992-0564 (S. Gnatyuk); 0000-0002-8049-3911 (M. Ibraimov); 0000-0002-2389-2262 (T. Namazbayev); 0000-0001-9890-4910 (A. Mukasheva)
© 2024 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

CEUR-WS.org/Vol-3654/short22.pdf

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

- Hardware implementation of crypto-systems guarantees its integrity.
- Encryption and storage of keys is carried out in the cipher board itself and not in the computer's RAM.
- It is possible to create systems based on hardware ciphers to protect information from unauthorized access and to delimit access to a computer.
- The use of paraphrase tires in the microprocessor architecture eliminates the threat of removing key information on electromagnetic radiation oscillations in the chains "Earth—Power" microcircuits, etc., arising during cryptographic transformations.

Define the basic operations over the numbers used in asymmetric cryptographic encryption algorithms. The construction of integers A to the power X modulo P (A^x mod P) is carried out by carrying out operations such as modular multiplication, and modular squaring. One approach to increasing the performance of public-key cryptosystems is to speed up the execution of these operations.

This work deals with the development of a modular square device. Various approaches to its development are analyzed.

2. Literature Review and Problem Statement

In the first approach, a device for placing a number A in a square is separately synthesized, forming a 2N-bit number. Then, with the individual device, the 2N-bit number is given modulo. In the works [10, 11] the methods of synthesis of a small-bit square and a multiplier are disclosed. They are then extended to produce the required 2N-bit devices.

With this approach, the complexity of synthesized devices increases dramatically as the number A decreases. The works [12–16] consider various variants of the synthesis of devices for squaring a number on different digital nodes and blocks, whereas the number A increases the complexity of the device also increases sharply. In the works [17–22] various variants of synthesis of autonomous devices of numbers modulo are considered.

In the second approach, the elementary operations of squaring and modular multiplication are combined in one step. The squared number A in each step is multiplied by the polynomial members A = $a_0 + a_1 2^1 \ldots . + a_{N-2}2^{N-1} + a_{N-1}2^{N-1}$ from the lower or higher bits with the result of multiplication modulo P.

This article discusses a second approach—the development of a device for modular squaring of numbers, where multiplication is carried out from the higher bits of the factor.

3. Development of the Devices

3.1. Devices for Modularizing into a Square

Fig. 1 shows a functional modular squaring circuit where multiplication A begins with a higher bit—a_{N-1}.

The device comprises RgA1 and RgA2 registers for receiving and storing the number A. RgA2 has a left-to-left shift circuit, RgP serves for storing the module P, RgR for storing the current residue values and the result of the operation. There are also Add1 binary and Partial Residue Shaper (PRF), Synchronization Block (BSIN), which contains a subtracting counter (Count), delay lines DL.1, DL.2, DL.3. BSIN inputs are provided with Clock pulse signals, the binary N-1 shift number code. Clock signals are emitted at BSIN outputs and are directed to the input of the moving RgA2. Count = 0 BSIN generates the "End of operation" signal.

Fig. 2 shows the functional circuit of the PRF partial residue shaper (1-variant), which consists of two binary Add3 and Add2 and a block of AND8-AND10 circuits, and a block of OR1 circuits. Left-shifting $\overline{\mathrm{P}}$ is applied to the right inputs of Add3, and $\overline{\mathrm{P}}$ is shifted to the right inputs of Add2. The input of the lower bits of adders is given a level of +1. On the left inputs of Add1 the sum $C_i = 2R_{i-1} + A * a_i$is given from the outputs of the Add1 adder. In PRF, the operation $R_i = S_i \, mod \, P.$ is executed [23–25].

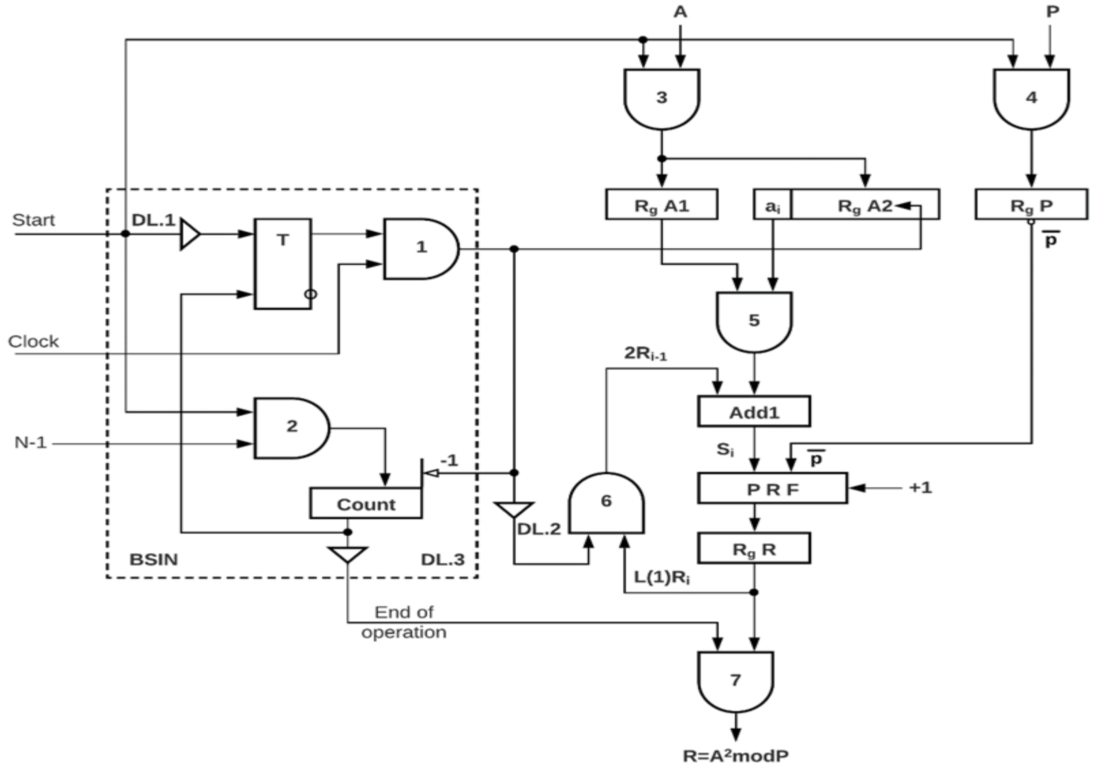

Figure 1: Functional circuit of the modular number A-squared device

In addition, Add2 executes $S_i + \bar{P} + 1$ and Add1 executes $S_i + 2\bar{P} + 1$. Furthermore, if $S_i > 2P$, then the transfer signal $C_3 = 1\, S_i - 2P$ difference from the Add3 output is transmitted to the output of the AND8 block as a result. Furthermore, Add3 takes the value Sign3 = 0, which blocks the transfer to the output AND9 of the difference $S_i - \mathrm{P}$ from the outputs of Add2.

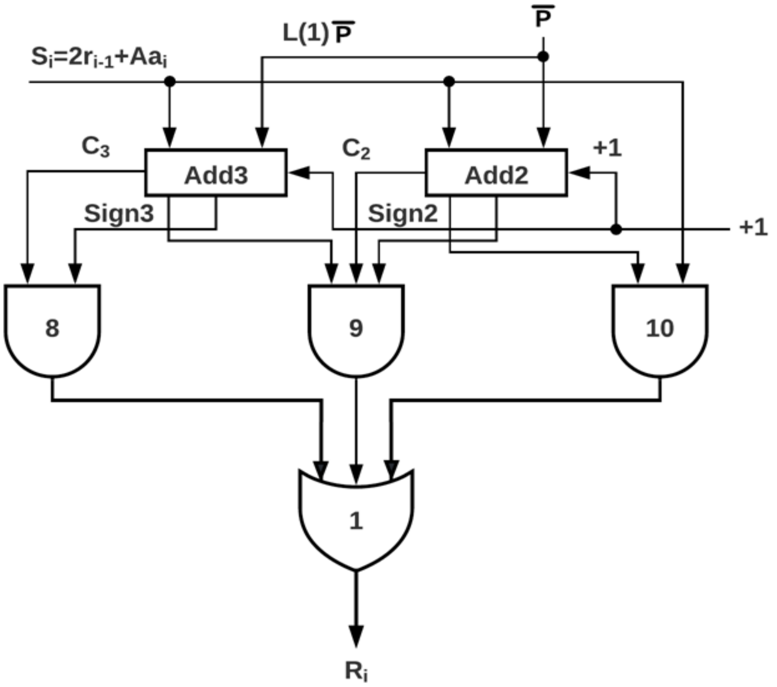

Figure 2: Functional circuit of the partial residue shaper (1—variant)

With $S_i > 2P$ the signal on the symbol output of the Add2 Sign2 = 0, which blocks the transfer to the output of AND10 of the value S_i.

With S_i<$2P$, Sign3 = 1 and $C_2 = 1$ allow the difference $S_i - P$ from Add2 output to be transmitted to AND9 outputs. Furthermore, the Sign2 = 0 signal blocks the transmission of the S_i value to the outputs of AND10. For $S_i <$ 2P и $S_i < P$, the Sign2 = 1 signal S_i is transmitted to the PRF output via the AND10 block of circuits.

Since the hardware cost of an N-bit binary adder is about 3 times that of an N-bit comparison circuit, it is advisable to construct a partial residue collector by replacing one binary adder with two comparison schemes. For this reason, Fig. 3 shows the second variant of the PRF partial residue shaper functional scheme, which is based on one binary adder and two comparison schemes COMP-1 and COMP-2.

In this scheme, the value $S_i = 2R_{i-1} + a_i * A$ is given by the left inputs of the COMP-1 scheme, where it is compared to the value of 2P, and in the COMP-2 scheme the value of S_i is compared to the value of P. If it is $S_i \geq$ 2P, then at the output of 2 COMP-1 the "1" signal is generated, which allows the 2$\overline{\text{P}}$ value to pass to the right inputs of the Add2 adder. Since the value of S_i is given to the left inputs, $R_i = S_i + 2\bar{P} + 1$ is executed.

If $2P > S_i \geq P$ conditions are met, then a signal "1" is formed at the output of the AND9 circuit, which allows passing the value $\overline{\text{P}}$. $R_i = S_i + \bar{P} + 1$ to the right inputs of Add2.

When $S_i <$ P a single signal is generated at output 1 of COMP-2, which allows the passage of the value S_i to the outputs AND8 and OR3. In this case, R = S_i.

Let's consider an example. A = 43_{10} = $\begin{cases} a_5\, a_4 a_3 a_2 a_1 a_0 \\ 1\;\; 0\;\, 1\;\, 0\;\, 1\;\, 1_2 \end{cases}$

$P = 54_{10}$: $2P = 108_{10}$

Calculation order $R = 43^2 mod54$ is shown in Table 1.

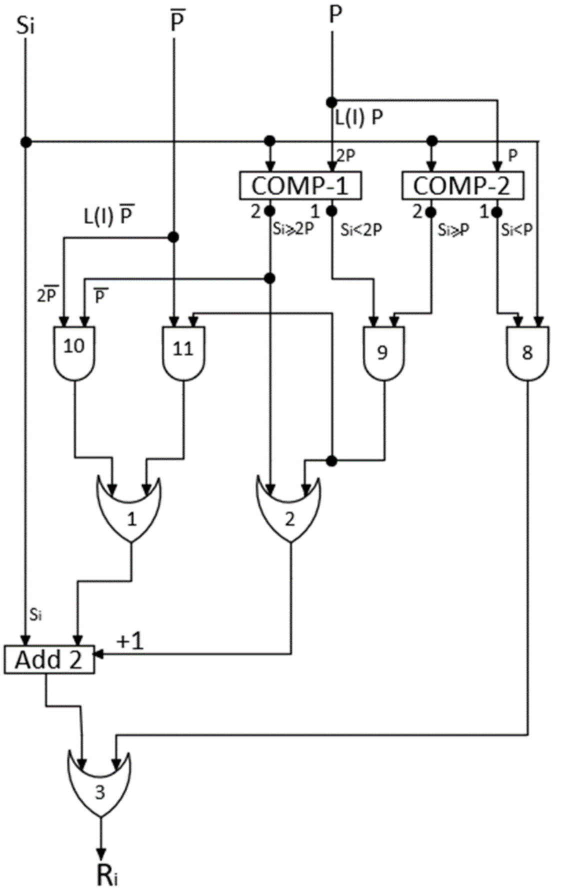

Fig. 3. Functional circuit of the partial residue shaper (2—variant)

Table 1
Calculation order $R = 43^2 mod54$

Clock	a_i	$S_i = 2R_i + a_i * A$	$R_i = S_i modP$
Start	$a_5 = 1$	$S_0 = 0 + A = 43_{10}$	$R_0 = S_0 modP = 43mod54 = 43_{10}$
Clock 1	$a_4 = 0$	$S_1 = 2R_{i-1} + a_4 * A = 86_{10}$	$R_1 = S_1 modP = 86mod5 = 32_{10}$
Clock 2	$a_3 = 1$	$S_2 = 2R_{i-1} + a_3 * A = 64 + 43 = 107_{10}$	$R_2 = S_2 modP = 107mod = 53_{10}$
Clock 3	$a_2 = 0$	$S_3 = 2R_{i-1} + a_2 * A = 106_{10}$	$R_3 = S_3 modP = 106mod = 52_{10}$
Clock 4	$a_1 = 1$	$S_4 = 2R_{i-1} + a_1 * = 104 + 43 = 47_{10}$	$R_4 = S_4 modP = 147 - 2P = 147 - 108 = 39_{10}$
Clock 5	$a_0 = 1$	$S_5 = 2R_{i-1} + a_0$ *A=78+43= 121_{10}	$R_5 = S_5 modP = 121 - 2P = 121 - 108 = 13_{10}$

Checking: $R = 43^2 modP = 1849mod54 = 1849 - 1836 = 13_{10}$

In Fig. 4 a functional circuit of the device of modular number-squaring based on two Add1, Add2, and two PRF residue shapers (2—variant) is shown, where on each clock number A is multiplied by two higher bits of the RgA2 register, which reduces by half the number of shifts of RgA2 [25].

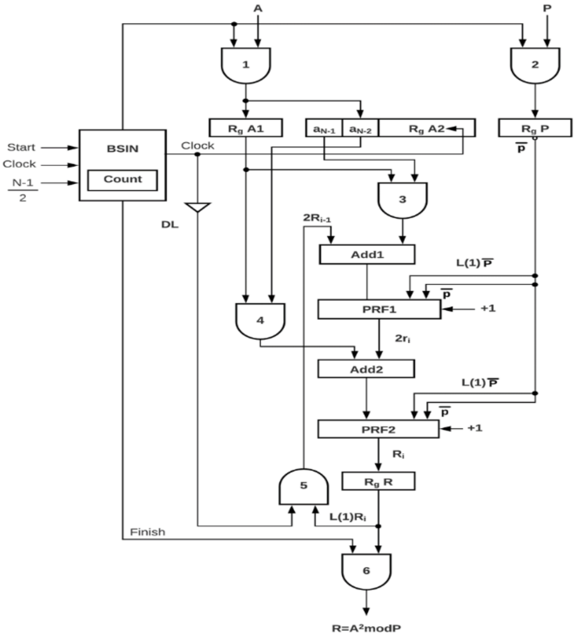

Figure 4: Functional circuit of the partial residue shaper (2—variant)

Table 2 shows the calculation progress $R = a^2 modP$ for the numbers $A = 59_{10}$ и $P = 65_{10}$ using the above-mentioned square-setting device [26].

$$A = 59_{10} = \begin{cases} a5a4a3a2a1a0 \\ \ 1 \ \ 1 \ \ 1 \ \ 0 \ \ 1 \ \ 1_2 \end{cases}$$

$$P = 65_{10} = 2P = 130_{10}$$

Table 2

Execution order R = 59^2мод65

Clock	$\boldsymbol{a_i}$ $\boldsymbol{a_{i-1}}$	***Add 1*** ***PRF 1***	***Add 2*** ***PRF 2***
Start	$a_5 = 1$ $a_4 = 1$	$S_0 = 0 + a_i \cdot A = 59_{10}$ $S_0 < P; 59 < 65$ $r_0 = S_0 = 59_{10}$	$S_1 = 2_r + a_4 \cdot A = 2 \cdot 59 + 59 = 117_{10}$ $R_1 = S_1 - 2P = 177 - 130 = 47_{10}$ $R_gR := \ 47_{10}$
Clock1	$a_3 = 1$ $a_2 = 0$	$S_2 = 2R_1 + a_3 \cdot A = 47 * 2 + 59 = 153_{10}$ $2P < S_2 > 0$ $r_1 = S_2 - 2P = 156 - 130 = 23_{10}$	$S_3 < 2R_1 + 0 \cdot A = 23 \cdot 2 = 46_{10}$ $S_3 < P$ $R_2 = S_3 = 46_{10}$ $R_gR_{\underline{+}} = 46_{10}$
Clock2	$a_1 = 1$ $a_0 = 1$	$S_4 = 2R_4 + A = 151_{10}$ $2P < S_4 > P$ $r_2 = S_4 - 2P = 151 - 136 = 21_{10}$	$S_4 = 2R_2 + A = 106_{10}$ $2P < S_4 > P$ $R_3 = S_4 - P = 104 - 68 = 36_{10}$ $R_gR = R = 36_{10}$

Checking: R=$59^2 mod65 = 3481 mod65 = 36_{10}$

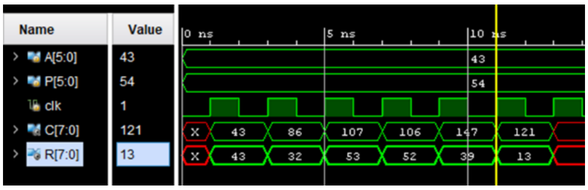

Figure 5: A time diagram of the operation of the modular number-squaring device with one-bit analysis of the RgA2 multiplier

Figs. 5 and 6 present the time diagrams of the above-considered devices for modular square construction with one-bit and two-bit analysis of the RgA2 multiplier per clock.

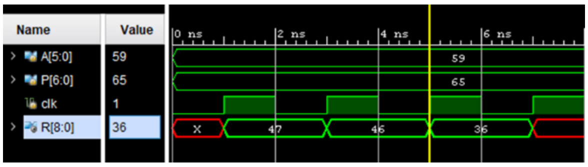

Figure 6: A time diagram of the modular number-squaring device with a two-bit analysis of the RgA2 multiplier

The experimental part of the work was implemented on FPLD Nexys 4 Artix-7 FPGA by Xilinx in Verilog equipment description language.

4. Conclusions

The experimental research showed the correct functioning of the developed devices for modularizing numbers into a square. A single-stage synthesis of the modular squaring device is more efficient than a two-stage synthesis, which makes it possible to handle the required number of operand bits without complicating the structure and composition of the operating blocks.

Future research study is related to the practical implementation of the proposed devices in real cryptosystems to provide security and privacy at high speed.

References

[1] Recommendations on the Importance of Critical Energy Infrastructure (CEI) Stakeholder Engagement, Coordination and Understanding of Responsibilities in Order to Improve Security, NATO Energy Security Centre of Excellence (2018).

[2] Cybersecurity in the Energy Sector, European Commission, Energy Security, Online Access Mode. URL: https://ec.europa.eu/energy/topics/energy-security/critical-infrastructure-and-cybersecurity_en?redir=1

[3] S. Mohammed, D. Taha, Performance Evaluation of RSA, ElGamal, and Paillier Partial Homomorphic Encryption Algorithms, International Conference on Computer Science and Software Engineering (2022) 89–94. doi: 10.1109/CSASE51777.2022.9759825.

[4] A. Bessalov, et al., CSIKE-ENC Combined Encryption Scheme with Optimized Degrees of Isogeny Distribution, in: Workshop on Cybersecurity Providing in Information and Telecommunication Systems, vol. 3421 (2023) 36–45.

[5] A. Bessalov, et al., Implementation of the CSIDH Algorithm Model on Supersingular Twisted and Quadratic Edwards Curves, in: Workshop on Cybersecurity Providing in Information and Telecommunication Systems, vol. 3187, no. 1 (2022) 302–309.

[6] A. Bessalov, et al., Modeling CSIKE Algorithm on Non-Cyclic Edwards Curves, in: Workshop on Cybersecurity Providing in Information and Telecommunication Systems, vol. 3288 (2022) 1–10.

[7] A. Bessalov, et al., Multifunctional CRS Encryption Scheme on Isogenies of Non-Supersingular Edwards Curves, in: Workshop on Classic, Quantum, and Post-Quantum Cryptography, vol. 3504 (2023) 12–25.

[8] V. Sokolov, P. Skladannyi, H. Hulak, Stability Verification of Self-Organized Wireless Networks with Block Encryption, in: 5th International Workshop on Computer Modeling and Intelligent Systems, vol. 3137 (2022) 227–237.

[9] S. Gnatyuk, T. Zhmurko, P. Falat, Efficiency Increasing Method for Quantum Secure Direct Communication Protocols, IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications. 1 (2015) 468–472.

[10] A. Mittal, F. Sidney, Secure Data Communication Using Padding Key Encryption Cryptography Algorithm, IEEE International Conference on Integrated Circuits and Communication Systems (2023) 1–5. doi: 10.1109/ICICACS57338.2023.10099570.

[11] Z. Hu, et al., Method of Searching Birationally Equivalent Edwards Curves Over Binary Fields, Adv. Intel. Syst. Comput. 754 (2019) 309–319.

[12] A. Aitkhotayeva, S. Tynymbayev, Aspects of Modular Hardware in Asymmetric Cryptography, Gazette of the National Academy of Sciences 5 (2014) 88–93.

[13] K. Sethi, R. Panda, An Improved Squaring Circuits for Binary Nowbens, Int. J. Adv. Comput. Sci. Appl. 3(2) (2012) 111–116.

[14] A. Kumar, D. Kumar, Hardware Implementation of 16*16 Multiplayer and Square Using Vedie Mathematics, International Conference on Signal Image and Video Processing (2012) 309–314.

[15] S. Kvardakov, O. Khromov, Probing Devices in the Square Copyright Certificate USSR 1417007, 13 (1992).

[16] V. Organs, I. Cornienko, L. Akulova, Copyright Certificate 699521, 48 (1992).

[17] A. Drozd, et al., Square Probing Device, Copyright Certificate Soviet Union 1451686, 1 (1990).

[18] V. Laqin, V. Shabadash, A. Shapito, Squaring Device, Copyright Certificate USSR 656056, 11 (1990).

[19] R. Román, et al., A Quantum-Resistant Face Template Protection Scheme using Kyber and Saber Public Key Encryption Algorithms, International Conference of the Biometrics Special Interest Group (2022) 1–5. doi: 10.1109/BIOSIG55365.2022.9897052.

[20] A. Ohromenko, V. Sorcerer, M. Sorcerer, Method of Bringing Integers After the Module, Patent on Corinth model, UA 118066, 14 (2017).

[21] S. Tynymbayev, et al., Schematic Solutions of the Establishment of Numbers Modulo for Public Key Measurement, AUES Gazettem, (2018) 33–41.

[22] S. Tynymbayev, Hig Speed Devices for Modular Reduction with Hardware Costs, Cogent Eng. 6(1) (2019) 1697555. doi: 10.1080/23311916.2019.1697555.

[23] S. Tynymbayev, et al., Devices for Bringing Numbers Modulo, Patent KZ 133812 (2019).

[24] S. Gnatyuk, et al., Method of Algorithm Building for Modular Reducing by Irreducible Polynomial, 16th International Conference on Control, Automation and Systems (2016) 1476–1479.

[25] O. Oksiiuk, V. Chaikovska, A. Fesenko, Security Technique for Authentication Process in the Cloud Environment, IEEE International Scientific-Practical Conference: Problems of Infocommunications Science and Technology (2019) 379–382.

[26] S. Srivastava, A. Tiwari, P. Srivastava, Review on Quantum Safe Algorithms Based on Symmetric Key and Asymmetric Key Encryption methods, 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (2022) 905–908, doi: 10.1109/ICACITE53722.2022.9823437.