

Modular reduction with step-by-step using of several bits of the reducible number

Sakhybay Tynymbayev¹, Yevgeniya Aitkhozhayeva², Dana Tananova³, Sairan Adilbekkyzy²

¹Department of Information Security Systems, Almaty University of Power Engineering and Telecommunications named after Gumarbek Daukeev, Almaty, Kazakhstan

²Department of Cybersecurity, Information Processing and Storage, Satbayev University, Almaty, Kazakhstan

³Department of Information Security Systems, Kazakh National University named after Al-Farabi, Almaty, Kazakhstan

Article Info

Article history:

Received Feb 23, 2021

Revised Dec 6, 2021

Accepted Dec 15, 2021

Keywords:

Field-programmable gate array

Hardware encryption

Modular reduction

Public-key cryptography

ABSTRACT

Although public key cryptography is known to solve the problem of physically secure key exchange, the main drawback of this system is its low performance during encrypting and decrypting data. One of the ways to solve this issue is to increase the speed of the modular reduction operation, one of the basic operations of asymmetric cryptoalgorithms. A new method of step-by-step reduction by the N -bit module P using several bits of the $2N$ -bit reducible number A in one step is proposed in this paper. The method is based on using multiples of the P and reducing modulo at each step not the entire initial number, but its parts ($A_1, A_2 \dots A_i$), which allows to reduce the bit capacity of A . A structural diagram of the hardware implementation of this method are developed. The main unit of the modular reduction device is a block of partial remainder formers, in which the partial remainder is computed using multiples of the P . The circuits are modeled in the Vivado Design Suite computer aided design (CAD) on base Artix-7 Field-programmable gate array (FPGA) device from Xilinx. Optimization of hardware costs is achieved by applying the same comparison circuits to compare different multiples of P with A_i .

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Sairan Adilbekkyzy

Department of Cybersecurity, Information Processing and Storage, Satbayev University

22a Satpaev street, Almaty 050013, Kazakhstan

Email: sairan.02.95@mail.ru

1. INTRODUCTION

In the past decade the challenge of protecting information has recently become even more urgent. Organizations and enterprises, regardless of their form of ownership and type of activity, were forced to switch to online work due to the COVID-19 pandemic. Global situation has clearly demonstrated the importance of digital technologies for maintaining the functioning, competitiveness and sustainability of individual companies and entire industries. The digital transformation of the entire society is proceeding at an accelerated pace due to the need for a sudden and forced transfer to the online environment of professional and personal life. Therefore, the risks and threats to information security have also multiplied.

Cryptographic encryption methods are considered one of the most reliable ways to protect information. Public key cryptography is becoming popular in modern information systems, since it does not require physically secure data transmission channels. This is an important advantage in the context of widespread digitalization of all sectors of society.

However, the application of asymmetric cryptoalgorithms is constrained by their low speed in comparison with symmetric cryptoalgorithms. It is worth-noticing that the hardware implementation allows

obtaining asymmetric cryptosystems with higher performance. In addition, it is easier to physically protect the equipment from outside penetration. The hardware implementation of encryption systems is simple to install. Nevertheless, the hardware implementation of asymmetric cryptoalgorithms performs encryption and decryption operations about 1000 times slower than symmetric cryptoalgorithms. Thus, many researchers around the world are engaged in improving the performance of asymmetric cryptosystems [1]-[7].

One of the approaches to improve the performance of public key cryptosystems is acceleration of basic modular arithmetic operations: modular multiplication, modular exponentiation, modular reduction [8]-[16]. The modular reduction is the most complex of them. Various number-theoretic methods for dividing and calculating the remainder when dividing by the module P and devices for their hardware implementation are proposed in research-scientific articles and patents [17]-[30].

As for this date different methods of obtaining the remainder modulo are implemented. Though, performance of most of them is achieved by increasing hardware costs, which are directly proportional to the bit capacity of the given numbers. Consequently, their application when reducing large numbers is not feasible. Whereas public-key cryptoalgorithms rest upon cumbersome modular arithmetic with multi-bit numbers. The increase in hardware costs leads to an increase in power consumption (deterioration of thermal conditions) and a decrease in reliability. The issue of accelerated determination of the remainder by an arbitrary modulus of the number (modular reduction) with optimization of effective costs is urgent. This paper considers numerous approaches and circuit solutions on the field-programmable gate array (FPGA) implementation for modular reduction operating units for asymmetric cryptosystems.

The proposed work is organized in the following sections: Section 2 describes a modular reduction method with step-by-step using of several bits of the reducible number and provides a detailed description of proposed hardware. In section 3 analytical and implementation results are explained with insights for further improvement. Finally, section 4 includes the concluding remark.

2. RESEARCH METHOD

Several recent studies have suggested that the method of step-by-step modular reduction using several bits of the reducible number in one step can improve the performance of calculating the modular reduction [31]-[35]. The proposed by authors method relies on the use of multiples of the module and modular reduction at each step not of the entire initial number, but of its parts, which makes it possible to reduce the bit capacity of the reducible numbers. The modular reduction is divided to the sequential obtaining of partial remainders, and each previous remainder after shifting to the left by several bits is used to obtain the next remainder. At the first step, to obtain a zero remainder, the N most significant bits of number A are used. When obtaining the remaining remainders, the least significant bits of the number A are used (several bits at each step). The implementation of this method makes it possible to design a high-speed device with the optimization of costs.

The developed device consists of a unit for generating multiples of an N -bit module P ($P, 2P, 3P$), a register RgA for storing $2N$ -bit reducible number A , $N/2$ formers of a partial remainder (PRF), a delay element DE . The method of obtaining the remainder modulo implemented in the device relies on the following principles: initially, the remainder R_0 is determined by the N -most significant bits of the number A (half of the bits of the number A). Subsequently, the remainder R_0 is shifted by two bits to the left - $4R_0$. The next two least significant bits after R_0 in register RgA are appended to $4R_0$, forming $A_1 = (4R_0 + a_{n-1}a_{n-2})$. Meanwhile, $PRF1$ determines the remainder R_1 modulo P from the number A_1 . The remainder R_1 is shifted by two bits to the left (i.e., $4R_1$) from the output of the $PRF1$ and the next two bits of the number A ($a_{n-3}a_{n-4}$) derive A_2 . Afterwards, A_2 is fed to the input of the $PRF2$, where remainder R_2 is generated. Ultimately, the formation of the number A_i ($i=1 \div N/2$), which is fed to the PRF_i to calculate the remainder R_i modulo P , is carried out in the same way. The formation of the remainder R_i in each PRF_i is realized in parallel due to the simultaneous comparison of A_i with the values of multiples of the module $P, \bar{P}, 2P, 2\bar{P}, 3P, 3\bar{P}$.

Figure 1 shows the structural diagram of a device for reducing a number modulo, which implements described method. The device contains a *block 4* for forming multiples of the P module $\bar{P}, P, 2\bar{P}, 2P, 3\bar{P}$ and $3P$), a *register 5* for storing a $2N$ -bit reducible number A , $N/2$ of PRF formers $8.1 \div 8.N/2$, delay element *6* (blocks are indicated by the numbers in figure). The information outputs of *block 4* are connected to the information inputs of the PRF $8.1 \div 8.N/2$ for transmitting the values $\bar{P}, 2\bar{P}, 3\bar{P}, P, 2P$ and $3P$. The information outputs of *register 5* of the number A are connected with the inputs of the PRF $8.1 \div 8.N/2$ for transmitting the corresponding two bits of the number A . Each PRF has its own two bits of the number A .

Firstly, the remainder of R_0 is determined by the N most significant bits of the $2N$ -bit number A in *register 5* then $4R_0$ is created by shifting two bits to the left. $4R_0$ together with the two bits attached to it from *register 5*, represents the number $A_1 = (4R_0 + a_{n-1}a_{n-2})$, which in PRF 8.1 . PRF 8.1 determines the remainder R_1 . Similarly, the generating of the number A_i ($i = 1 \div N/2$) is achieved, then it is fed to the PRF $8.i$ to compute

the remainder R_i modulo P . The remainder R_{i-1} from the output of the PRF $8.i-1$ is shifted to the left by two bits towards the most significant bits, the next two least significant bits of the number A are attached to it $A_i = L(2) R_{i-1} + a_{n-2i+1} a_{n-2i} = 4R_{i-1} + a_{n-2i+1} a_{n-2i}$.

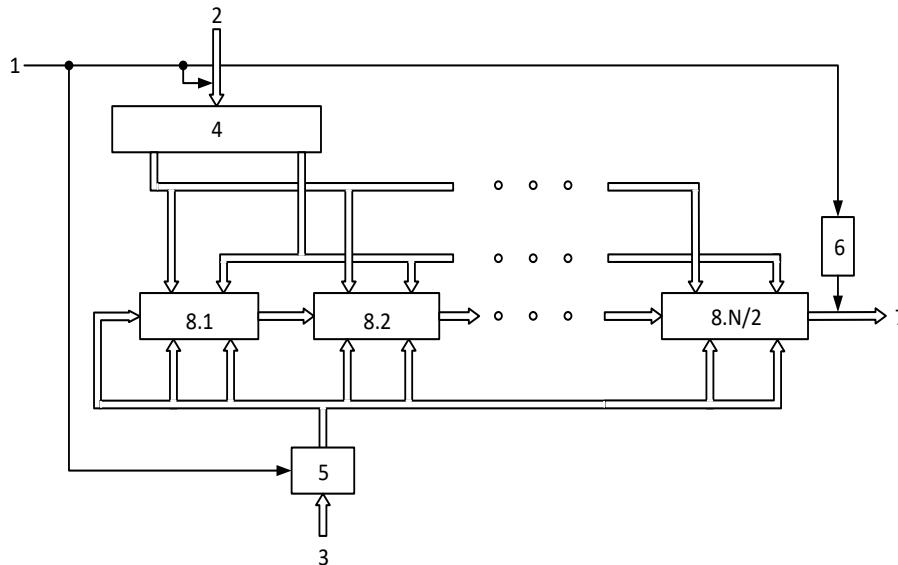


Figure 1. Structural diagram of a device for reducing a number modulo

Eventually, bits a_1 and a_0 of the number A from register 5 and a partial remainder $4R_{N/2-1}$ from the outputs of the PRF $8.N/2-1$ are fed to the inputs of the PRF $8.N/2$. At the outputs of the PRF $8.N/2$, the final result $R_{N/2} = N$ is formed. Consider the operation of the device for modulo reducing a $2N$ -bit number A by N -bit module P in more detail:

First of all, according to the "Start" signal (input 1), the value of P is received from input 2 into block 4, the value of the number A from input 3 is taken into register 5. In block 4, multiples of the module P , \bar{P} , $2P$, $2\bar{P}$, $3P$, $3\bar{P}$ are generated, which are fed to the inputs of all PRF $8.1 \div 8.N/2$. Simultaneously, N high-order bits (i.e. R_0) from the outputs of register 5 with the shift by two bits towards the high-order bits are fed to the inputs of the PRF 8.1 . In this case, the bits $a_{n-1} a_{n-2}$ of the number A is attached to the least significant bits of the shifted R_0 from register 5, forming the number A_1 . In turn PRF 8.1 calculates the remainder R_1 . Further, the value of R_1 with the shift two bits towards the higher bits, as well as the bits $a_{n-3} a_{n-4}$ of the number A form A_2 and are fed to the inputs of the PRF 8.2 , and where the partial remainder R_2 is created. At the final stage, the partial remainder $R_{N/2-1}$ from the outputs of the previous PRF $8.N/2-1$ with two-bit shift towards the higher bits and bits $a_1 a_0$ of the number A from register 5 are applied to the inputs of the PRF $8.N/2$ and at its outputs the remainder $R_{N/2}$ is formed. By the signal "End of operations" that is formed at the output of the delay element 6, the remainder of $R_{N/2}$ is issued to the output of the device 7. The time of formation of the result T_{fr} is determined by the total time of the signal passing through the PRF $8.1 \div 8.N/2$, i.e. $T_{fr} = N/2 * T_{PRF}$.

Figure 2 illustrates a functional diagram of the partial remainder former block PRF_i , which consists of a binary adder Add ; two comparator circuits $CC-1$ and $CC-2$; blocks of logic gates $AND1 \div AND5$, $AND8$; gates $AND6$, $AND7$, $OR3$; blocks of logic gates $OR1$, $OR2$; NOT gate.

The circuit works as follows: from the block of former of the multiples of module 4, the bits of the $2P$ module are fed to the right inputs of the $CC-1$ circuit. On the right inputs of the $CC-2$ circuit comes from block 4 the value of the module P through $AND1$ and $OR1$ or the value of $3P$ through $AND2$ and $OR1$. The value of the previous remainder R_{i-1} , shifted to the left by two bits towards the most significant bits, with the next two bits of the reducible number attached to it, determines the value $A_i = L(2) R_{i-1} + a_{n-2i+1} a_{n-2i}$.

The A_i value is fed to the right inputs of the Add adder through the $AND8$ and to the left inputs of $CC-2$ and $CC-1$. The circuit $CC-1$ compares the A_i with the $2P$. If, in this case, $A_i \geq 2P$, then at the output 2 of the $CC-1$ circuit, a unit impulse 1 is generated, which is fed to the control input of the $AND2$, allowing the passage of the bits of multiples of the $3P$ module to the right inputs of the $CC-2$. At the same time, at the output 1 of the $CC-1$ - signal 0, which disables the passage of the bits of the module P through the $AND1$ to the inputs of the $CC-2$ and through the $AND6$, NOT gates lead to the formation of 1 impulse at the right input of the $AND7$.

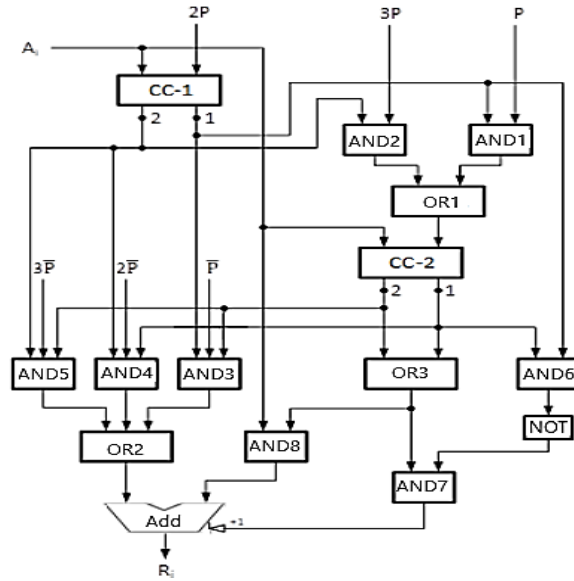


Figure 2. Functional diagram of PRFi

Besides, when comparing the A_i with the $2P$, if the inequality $A_i < 2P$ holds, *signal 1* is generated at the *output 1* of the *CC-1*, allowing the passage of the bits of the P module through the *AND1* circuit block to the right inputs of the *CC-2*. At the same time, at the *output 2* of the *CC-1* there is a *signal 0* that stops the passage of $3P$ through the *AND2* to the inputs of the *CC-2*.

Whereas the condition $A_i \geq 2P$ is fulfilled, the *CC-2* compares the A_i with the $3P$. If at the same time $A_i < 3P$, then *signal 1* will be set at the *output 1* of the *CC-2*, it is fed to the input of the *OR3* gate and to the control inputs of the *AND4*, allowing the passage of the one's complement $2\bar{P}$, supplied to the information inputs *AND4*, to the input of the adder *Add* through the *OR2*. At the output of the *OR3*, *signal 1* is generated, allowing the passage of A_i through the *AND8* to the adder *Add*, and the passage of which through the circuit *AND7* to the *input +1* of the adder *Add* is allowed by a unit impulse from the output of the *NOT* gate. In this case, the *Add* adder performs the operation $R_i = A_i + 2\bar{P} + 1$.

If the condition $A_i \geq 2P$ is met, when comparing the A_i with the $3P$ on the *CC-2*, it turns out that $A_i \geq 3P$, then at the *output 2* of the *CC-2*, *signal 1* will be set, which is fed to the input of the *OR3* and to the control inputs of the *AND5*, allowing the passage of the $2\bar{P}$ supplied to the information inputs *AND5*, to the input of the *Add* adder through the *OR2*. At the output of the *OR3*, *signal 1* is generated, allowing the passage of A_i through the *AND8* to the adder *Add*, and the passage of which through the circuit *AND7* to the *input +1* of the adder *Add* is allowed by a single signal from the output of the *NOT*. In this case, the *Add* adder performs the operation $R_i = A_i + 3\bar{P} + 1$.

As well as, if the condition $A_i < 2P$ is set, when comparing the A_i with the P on the *CC-2*, it turns out that $A_i < P$, then *signal 1* will be set at the *output 1* of the *CC-2*, which is fed to the input of the *OR3* and *AND6*. At the output of the *OR3*, *signal 1* is generated, allowing the passage of A_i through the *AND8* to the adder *Add*, and the passage of which through the circuit *AND7* to the *input +1* of the adder *Add* is prohibited by a zero signal from the output of the *NOT*. In this case, the *Add* adder performs the operation $R_i = A_i + 0 = A_i$, since multiples of the module \bar{P} , $2\bar{P}$, $3\bar{P}$ are not fed to the second input of the adder *Add*, as they are disabled by zero control signals on the *AND3*, *AND4*, *AND5*. Consider the operation of the circuit using an example of reducing the $2N$ -bit number A to the N -bit module P .

$$A = 1437_{10} = \begin{Bmatrix} a_{11} & a_{10} & a_9 & a_8 & a_7 & a_6 & a_5 & a_4 & a_3 & a_2 & a_1 & a_0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1_2 \end{Bmatrix}$$

$$2N=12, N=6, N/2=3. P=35_{10}=100011_2, 2P=70_{10} \text{ and } 3P=105_{10}$$

The most significant six bits of the binary representation of the number A determine the value $R_0=010110_2=22_{10}$. Attaching the remainder R_0 shifted to the left by two bits with the next two bits a_5a_4 of the number A , gives the number $A_i=L(2)R_0+(a_5a_4) = 4R_0+(a_5a_4) = (88+1)_{10}=89_{10}$. For clarity, all calculations to solve $R=A \bmod P$ are given in Table 1 in decimal notation.

Table 1. Procedure for calculating $R=A \bmod P$

1-step <i>PRF1</i>	2-step <i>PRF2</i>	3-step <i>PRF3</i>
$A_1=L(2)R_0+a_5a_4=88_{10}+1_{10}=89_{10}$.	$A_1=L(2)R_0+a_5a_4=88_{10}+1_{10}=89_{10}$.	$A_3=L(2)R_2+a_1a_0=36_{10}+1_{10}=37_{10}$.
Since $70 < 89 < 105$, the ratio	Since $70 < 89 < 105$, the ratio	Since $35 < 37 < 70$, the inequality
$2P \leq A_1 < 3P$ takes place.	$2P \leq A_1 < 3P$ takes place.	$P \leq A_3 < 2P$ takes place.
Therefore, the operation will be performed: $R_1 = A_1 - 2P = 89 - 70 = 19_{10}$.	Therefore, the operation will be performed: $R_1 = A_1 - 2P = 89 - 70 = 19_{10}$.	This will execute the operation: $R_3 = A_3 - P = 37_{10} - 35_{10} = 2_{10}$.

*Check: $R=A \bmod P=1437 \bmod 35=2_{10}$

3. RESULTS AND DISCUSSIONS

The verification of the correct functioning of the developed circuit was performed by modeling in Vivado design suite CAD with a focus on the implementation of devices on FPGA of the Artix-7 series from Xilinx. Moreover, the Verilog hardware description language was chosen to describe the devices [36]. In the simulation, the numbers $A = 1437_{10}$ and $P = 35_{10}$ were used as illustrated in example. In this case, the result of reducing the initial number A will be equal to $R=A \bmod P=1437 \bmod 35=2_{10}$. Figure 3 shows waveforms of the operation of the circuits for reducing the number A modulo P with step-by-step use of two bits of the reducible number A .

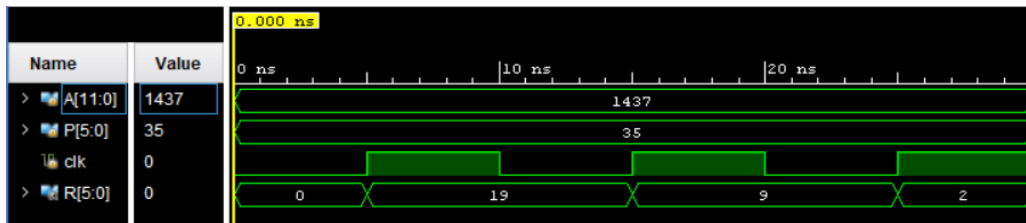


Figure 3. Timing diagram of the operation of the device for reducing the number modulo using two bits of the reducible number

The simulation accomplished for the selected numbers demonstrates that three clock signals were required to the obtain remainder with the stepwise use of two bits of the reducible number A . The timing diagram presents the value of the reducible number $A=1437_{10}$, the value of the module $P=35_{10}$, the number of clock signals used, the remainders R_i (19, 9, 2) obtained at each step. It is noted that the complication of the design of the partial remainder formers that are basic part of the entire device, makes it possible to accelerate the process of obtaining the remainder by using three bits of the A number added to the partial remainder. In this case, the number of *PRF* decreases and, consequently, the number of obtained partial remainders required for modular reduction and the operation time is reduced.

The structural diagram of the modular reduction device using three bits of the reducible number will be similar to the structural diagram of the device shown in Figure 1. But the number of forms *PRF* is $N/3$ and multiples of the modules $P, \bar{P}, 2P, 2\bar{P}, \dots, 7P, 7\bar{P}$ should be formed in *block 4*. However, important to point out that the *PRF* circuit will be more complex than that shown in Figure 2, as it is necessary to use five comparator circuits instead of two as part of the *PRF*. Nonetheless, since they work in parallel, the response time of the *PRF* does not increase. Binary representations $P, \bar{P}, 2P, 2\bar{P}, \dots, 6P, 6\bar{P}$ and $7P, 7\bar{P}$ should be fed to the inputs of *PRF_i* from the outputs of the unit for forming multiples of the module. The information outputs of the *register 5* of the number A are connected with the inputs of all *PRF_s* $8.1 \div 8.N/3$ to transmit the corresponding three bits of the number A . For each *PRF*, its own three bits are supplied.

The value of the partial remainder from the outputs of the *PRF_{i-1}* with a three-bit shift towards the higher bits with the addition of the next three bits of the reducible number is fed to the left inputs of the adder and comparators of the *PRF_i*, where they are compared with the value of the modules $P, 2P, \dots, 7P$. As a result, control signals are generated that commute the value of one of the modules $\bar{P}, 2\bar{P}, \dots, 7\bar{P}$ to the right inputs of the binary adder, forming the remainder R_i at the outputs of the adder. The time of forming the result T_{fr} is determined by the total time of the signal passing through the *PRF*, i.e., $T_{fr} = N/3 T_{PRF}$. In the general case, the reduction of a number modulo with step-by-step use of three bits of the reducible number allows you to speed up the operation by one and a half times compared to using two bits of the reducible number.

4. CONCLUSION

The increase in terms of the speed of the device is achieved by the simultaneous comparison of several multiples of the P module in the scheme of partial remainder former with the next reducible number A_i . The use of several bits of the reducible number in one step also makes it possible to accelerate the receipt of the remainder. Step-by-step modular reduction decreases the bit capacity of the reducible numbers, which leads to a decline in hardware costs. Optimization of hardware costs is also achieved by using the same comparators in the partial remainder formers PRF to compare different multiples of the module P with A_i .

Decreasing hardware costs makes possible to increase device reliability and improve thermal operation. The gathered simulation results confirm the correctness of the developed circuits. As well as the depletion in time costs can be achieved with the increase in the number of bits of the reducible number A from two or more, used at each step of the reduction modular. The developed circuit efficiently can be applied both in cryptographic applications and in digital devices to form elements of finite fields.




REFERENCES

- [1] T. Eisenbarth, S. Kumar, C. Paar, A. Poschmann, and L. Uhsadel, "A Survey of Lightweight-Cryptography Implementations," in *IEEE Design & Test of Computers*, vol. 24, no. 6, pp. 522-533, Nov.-Dec. 2007, doi: 10.1109/MDT.2007.178.
- [2] Y. Wang and R. Li, "FPGA based unified architecture for public key and private key cryptosystems," *Frontiers of Computer Science*, vol. 7, no. 3, pp. 307-316, 2013, doi: 10.1007/s11704-013-2187-2.
- [3] S. S. Chawla and N. Goel, "FPGA implementation of an 8-bit AES architecture: A pipelined and masked approach," *2015 Annual IEEE India Conference (INDICON)*, 2015, pp. 1-6, doi: 10.1109/INDICON.2015.7443849.
- [4] Y. Li, "Improved RSA Algorithm in Hardware Encryption," *Applied Mechanics and Materials*, vol. 543-547, pp. 3610-3613, 2014, doi: 10.4028/www.scientific.net/AMM.543-547.3610.
- [5] N. S. S. Srinivas and M. Akramuddin, "FPGA based hardware implementation of AES Rijndael algorithm for Encryption and Decryption," *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, 2016, pp. 1769-1776, doi: 10.1109/ICEEOT.2016.7754990.
- [6] V. Narasimha Nayak, M. Ravi Kumar, K. Anusha, and C. Kranthi Kiran, "FPGA based asymmetric crypto system design," *International Journal of Engineering & Technology*, vol. 7, no. 11, p. 612, 2017, doi: 10.14419/ijet.v7i11.1.10788.
- [7] V. Shende and M. Kulkarni, "FPGA based hardware implementation of hybrid cryptographic algorithm for encryption and decryption," *2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT)*, 2017, pp. 416-419, doi: 10.1109/ICEECCOT.2017.8284540.
- [8] Y. Zh. Aitkhozhayeva and S. T. Tynymbayev, "Aspects of hardware reduction modulo in asymmetric cryptography," *Bulletin of National Academy of Sciences of the Republic of Kazakhstan*, vol. 5, no. 375, pp. 88-93, 2014.
- [9] A. P. Renardy, N. Ahmadi, A. A. Fadila, N. Shidqi, and T. Adiono, "Hardware implementation of montgomery modular multiplication algorithm using iterative architecture," *2015 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, 2015, pp. 99-102, doi: 10.1109/ISITIA.2015.7219961.
- [10] B. Hanindhito, N. Ahmadi, H. Hogantara, A. I. Arrahmah, and T. Adiono, "FPGA implementation of modified serial montgomery modular multiplication for 2048-bit RSA cryptosystems," *2015 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, 2015, pp. 113-118, doi: 10.1109/ISITIA.2015.7219964.
- [11] W. Dai, D. Chen, R. C. C. Cheung, and Ç. K. Koç, "FFT-Based McLaughlin's Montgomery Exponentiation without Conditional Selections," in *IEEE Transactions on Computers*, vol. 67, no. 9, pp. 1301-1314, 1 Sept. 2018, doi: 10.1109/TC.2018.2811466.
- [12] B. Li, B. Lei, Y. Zhang, and S. Lei, "A Novel and High-Performance Modular Square Scheme for Elliptic Curve Cryptography Over $GF(p)$," in *IEEE Tran. Circuits and Sys. II: Exp. Bri.*, vol. 66, no. 4, pp. 647-651, 2019, doi: 10.1109/TCSII.2018.2867618.
- [13] E. Ozcan and S. S. Erdem, "A High Performance Full-Word Barrett Multiplier Designed for FPGAs with DSP Resources," *2019 15th Conference on Ph.D Research in Microelectronics and Electronics*, 2019, pp. 73-76, doi: 10.1109/PRIME.2019.8787740.
- [14] B. Liu, "Research and Implementation of RSA IP Core Based on FPGA," *Data Processing Techniques and Applications for Cyber-Physical Systems (DPTA 2019)*, pp. 1311-1319, 2020, doi: 10.1007/978-981-15-1468-5_154.
- [15] L. Gnanasekaran, A. Eddin, H. El Naga, and M. El-Hadedy, "Efficient RSA Crypto Processor Using Montgomery Multiplier in FPGA," *Advances in Intelligent Systems and Computing*, pp. 379-389, 2019. doi: 10.1007/978-3-030-32523-7_26
- [16] E. Öztürk, "Design and Implementation of a Low-Latency Modular Multiplication Algorithm," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 6, pp. 1902-1911, June 2020, doi: 10.1109/TCSI.2020.2966755.
- [17] Pankratova I. A. *Number-theoretical methods of cryptography*. Tomsk State University, 2009.
- [18] V. M. Zakharov, E. L. Stolov, and S. V. Shalagin, "Apparatus for generating remainder for given modulo, (in Russian)," R.U. Patent 2 421 781 C1, Oct. 19, 2009.
- [19] V.V. Kopytov, V.I. Petrenko, and A.V. Sidorchuk, "Device for generating remainder from arbitrary modulus of number," Russian Federation Patent 2445730, Mar. 20, 2012.
- [20] E. Pisek and T. M. Henige, "Method and apparatus for efficient modulo multiplication," U.S. Patent 8417756, Apr. 9, 2013.
- [21] Z. Yin, W. Yang, and J. Xiong, "An adaptive modular reduction based error-detection algorithm," *2012 4th International High Speed Intelligent Communication Forum*, 2012, pp. 1-4, doi: 10.1109/HSIC.2012.6212967.
- [22] M. Huang and D. Andrews, "Modular Design of Fully Pipelined Reduction Circuits on FPGAs," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 9, pp. 1818-1826, Sept. 2013, doi: 10.1109/TPDS.2012.267.
- [23] R. J. Lambert, "Method and apparatus for modulus reduction," United States Patent 8862651, 2014.
- [24] Y. Aitkhozhayeva and S. Tynymbaevich, "Method for forming element of finite fields in digital computing device, involves Performing subtraction process on Raman adder to form preset residue, and providing combinational circuits to compare residue values," Patent of the Republic of Kazakhstan 30983-A4, Nov. 15, 2014. (In Russian).
- [25] M. Bockes and J. Pulkus, "Method for arbitrary-precision division or modular reduction," U.S. Patent 9042543, May 26, 2015.
- [26] H. Yu, G. Bai, and H. Hao, "Efficient Modular Reduction Algorithm without Correction Phase," *Frontiers in Algorithmics*, pp. 304-313, 2015, doi: 10.1007/978-3-319-19647-3_28.




- [27] M. Kovtun and V. Kovtun, "Review and classification of algorithms for dividing and modulating large integers for cryptographic applications". Accessed: January 20, 2018. [Online]. Available: <https://docplayer.ru/30671408-Obzor-i-klassifikaciya-algoritmov-deleniya-i-privedeniya-po-modulyu-bolshih-celyh-chisel-dlya-kriptograficheskikh-prilozheniy.html>.
- [28] P. Choi, M. Lee, J. Kim, and D. K. Kim, "Low-Complexity Elliptic Curve Cryptography Processor Based on Configurable Partial Modular Reduction Over NIST Prime Fields," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 65, no. 11, pp. 1703-1707, Nov. 2018, doi: 10.1109/TCSII.2017.2756680.
- [29] S. Tynymbayev, E. Aitkhozhayeva, R. Berdibayev, S. Gnatyuk, T. Okhrimenko, and T. Namazbayev, "Development of Modular Reduction Based on the Divider by Blocking Negative Remainders for Critical Cryptographic Applications," *2019 IEEE 2nd Ukraine Conference on Electrical and Computer Engineering (UKRCON)*, 2019, pp. 809-812, doi: 10.1109/UKRCON.2019.8879846.
- [30] R. Dorrance, A. Belogolov, H. Wang, and X. Zhang, "A Digital Root Based Modular Reduction Technique for Power Efficient, Fault Tolerance in FPGAs," *2020 30th International Conference on Field-Programmable Logic and Applications (FPL)*, 2020, pp. 341-346, doi: 10.1109/FPL50879.2020.00063.
- [31] Y. Aitkhozhayeva, S. Tynymbayev, S. Adilbekkyzy, A. Skabylov, and M. Ibraimov, "Design and research of the behavioral model for the modular reduction device," *Eurasian Physical Technical Journal*, vol. 17, no. 1, pp. 151-156, 2020, doi: 10.31489/2020no1/151-156.
- [32] S. Tynymbayev, Y. Aitkhozhayeva, and S. Adilbekkyzy, "High speed device for modular reduction," *Bulletin of National Academy of Sciences of the Republic of Kazakhstan*, vol. 6, no. 376, pp. 147-152, 2018, doi: 10.32014/2018.2518-1467.38.
- [33] S. Tynymbayev, Aitkhozhayeva, O. Mamyrbayev, and S. Adilbekkyzy, "Device for reduction mod of numbers," Patent of the Republic of Kazakhstan 33812, Jul. 29, 2019.
- [34] S. Tynymbayev, R. Berdibayev, T. Omar, Y. Aitkhozhayeva, A. Shaikulova, and S. Adilbekkyzy, "High-speed devices for modular reduction with minimal hardware costs," *Cogent Engineering*, vol. 6, no. 1, 2019, doi: 10.1080/23311916.2019.1697555.
- [35] S. Tynymbayev, Aitkhozhayeva, O. Mamyrbayev, and S. Adilbekkyzy, "Device for reduction of numbers by module with the analysis of three digits of the given number in steps," Patent of the Republic of Kazakhstan 34255, Mar. 30, 2020.
- [36] N. Botros, *HDL with Digital Design*. Bloomfield: Mercury Learning & Information, 2015.

BIOGRAPHIES OF AUTHORS






Sakhybay Tynymbayev    is Professor in the Department of Information Security Systems, Almaty University of Power Engineering and Telecommunications named after Gumarbek Daukeev, Candidate of technical sciences. Professor S. Tynymbayev got an academic degree at Bauman Moscow State Technical University, Russia. He has more than 50 years of scientific and pedagogical experience. He is interested in computer science, cryptographic hardware and embedded systems and physical and numerical modeling of operating units for digital devices. He has over 200 scientific and educational works. He can be contacted at email: s.tynym@mail.ru.






Yevgeniya Aitkhozhayeva    is associate professor in Department of Cybersecurity, Information Processing and Storage Satbayev University, Candidate of Technical Sciences. One of the leading scientists of the Republic of Kazakhstan in the field of computing and information security, a teacher with over 40 years of experience. She received an academic degree at Department of Computer Engineering St. Petersburg State Electro Technical Institute (Technical University "LETI"), Russia. Professor Y. Aitkhozhayeva is interested in Information Security, Databases Systems, Hardware of Cryptography and has over 200 scientific and educational works. She can be contacted at email: ait_evg@mail.ru.



Dana Tananova    received a bachelor's degree in Information Systems from Al-Farabi Kazakh National University in Almaty. From 2012 to 2014, she studied for a master's degree and received an academic master's degree in Computer Engineering and Software at KazNU. From 2017 to 2020, she studied for a doctoral degree in Information Security Systems. Dana is senior lecturer at the Department of Telecommunications and Innovative Technologies, Almaty University of Power Engineering and Telecommunications named after Gumarbek Daukeev She is interested in information security in different platform environments. She can be contacted at email: tananova.dana@gmail.com.



Sairan Adilbekkyzy    received the B.Sc. and M.Sc. degrees in military affairs and security studied at Kazakh National Research Technical University named after K. I. Satbayev, specialty "Information Security Systems". Sairan is doctoral degree student at Satbayev University in Management of Information systems since September 2020. Author and co-author of more than 15 scientific works. Her main field of interests are Information Security and Hardware of Cryptography. She can be contacted at email: sairan.02.95@mail.ru.