The Oscar Publishing Services

American Journal of Applied Science and Technology

VOLUME Vol.05 Issue 12 2025
PAGE NO. 31-36
DOI 10.37547/ajast/Volume05Issue12-06

Symmetric Encryption Algorithm Based On A One-Way Function Extended By Parametric Algebra

Khudoykulov Zarifjon

Tashkent University of Information Technologies named after Muhammad al-Khorazmi,100084, Amir Temur Avenue 108, Tashkent, Uzbekistan

Khudoynazarov Umidjon

Tashkent University of Information Technologies named after Muhammad al-Khorazmi,100084, Amir Temur Avenue 108, Tashkent, Uzbekistan

Received: 12 October 2025; **Accepted:** 04 November 2025; **Published:** 09 December 2025

Abstract: A key concern in information security within communication systems is the necessity of employing robust symmetric encryption algorithms to safeguard large volumes of data. Designing stable and secure symmetric encryption algorithms remains one of the most challenging problems in cryptology. This paper introduces a methodology to enhance symmetric cryptographic systems by utilizing a one-way function-based encryption algorithm, offering a level of security comparable to the discrete logarithm problem in finite fields.

Keywords: Cryptographic strength, discrete logarithm, factorization, inversion, one-way function, parametric algebra, parametric multiplication, symmetric cryptography.

INTRODUCTION:

One of the most effective ways to protect information is through the use of symmetric cryptographic algorithms. These encryption algorithms safeguard data from unauthorized access, ensuring that only individuals with the appropriate key can decrypt and read the message. The effectiveness of an encryption algorithm depends on factors such as key length, encryption process, and the number of encryption rounds. Selecting a strong key and implementing robust key management practices are crucial for ensuring the overall security of systems utilizing symmetric encryption [1].

Encryption plays a vital role in protecting the privacy and confidentiality of information, whether it is stored or transmitted. If unauthorized access occurs, the encrypted data appears as an unintelligible jumble of bytes, rendering it unreadable. Decryption is the reverse process, converting ciphertext back into its original readable form. Both encryption and decryption rely on a unique, randomized string of bits known as an encryption key. The security of the encryption increases with the length of the key, as longer keys are much harder to crack. Figure 1 illustrates the encryption and decryption process for plaintext in a cryptographic system [2].

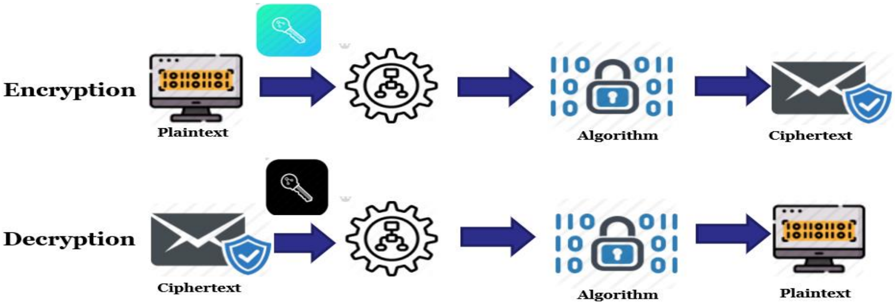


FIGURE 1. Cryptographic system

Symmetric key cryptosystems. Symmetric key cryptosystems use a single, shared key for both encryption and decryption. To preserve confidentiality, the key must be kept secret and accessible only to authorized parties. While symmetric encryption is highly efficient, implementing it on a large scale poses challenges, particularly in securely exchanging and maintaining the secrecy of shared keys [2]. Figure 2 illustrates the encryption and decryption process for plaintext in a symmetric cryptographic system.

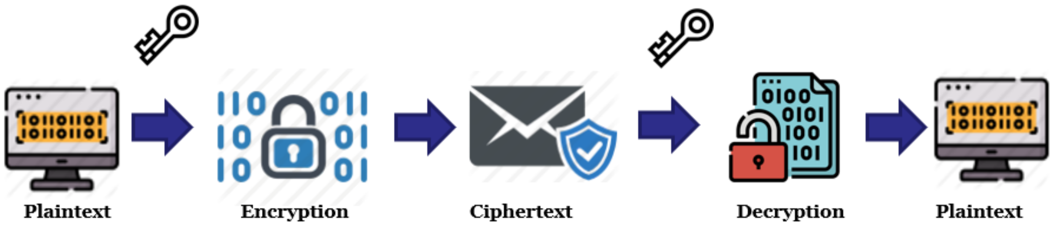


FIGURE 2. Symmetric key cryptographic system

Modern symmetric encryption algorithms fall into two main categories:

Block Encryption*:* Block encryption processes data in fixed-size blocks. Popular block cipher algorithms are designed using architectures like the Feistel network, Substitution-Permutation Network (SPN), and Lai-Massey architecture [3].

Stream Encryption*:* Stream cipher algorithms encrypt data as a continuous stream of bits or bytes. These algorithms typically utilize random number generators to produce a keystream for encryption [3].

There is a wide range of symmetric encryption algorithms, including: AES, DES, 3DES (Triple DES), Blowfish, Twofish, Threefish, RC variations, A5, etc. [4].

The foundation of modern block and stream encryption algorithms lies in random number generators and XOR operations, which do not introduce significant mathematical complexity compared to symmetric encryption algorithms [1]. In contrast, public key cryptographic systems rely on one-way functions, which add various levels of mathematical complexity. These one-way functions involve challenges such as discrete logarithms, factoring, and finding rational points on elliptic curves within a finite field [5]. Additionally, there are one-way functions based on integer algebraic structures that raise the degree parameter problem [6].

Researchers have enhanced existing public-key and symmetric encryption algorithms by applying parametric algebra. However, no symmetric encryption algorithms currently incorporate one-way functions, which would introduce a different form of mathematical complexity.

This article introduces a symmetric encryption algorithm that is based on a one-way function and has been improved using parametric algebra.

LITERATURE REVIEW

This article examines existing literature on current encryption techniques and algorithms. Scholars such as Shriram P., Navghare N., Bhalerao A. [1], Subhi R. and M. Zeebaree [2], Achuthshankar A. [3], N. A. Advani and A. M. Gonsai [4], Khasanov X.P. [6], Mulder V. [7], D. Akbarov [8], M. Turdimatov and F. Mukhtarov [9], O.P. Akhmedova [10], Schneier B. [11], and others have made significant contributions to this field.

METHOD

A major challenge in information security is ensuring the confidentiality and integrity of large volumes of data during storage, transmission, and processing. Developing stronger symmetric encryption algorithms is a key approach to addressing the current issues in cryptography. By applying the required mathematical representations of one-way functions, it is possible to build sufficiently secure symmetric cryptosystems.

One-way function is defined as such: $y = f(x)$, where $f(x) = y$ can be calculated with ease for any x within the field of its definition. However, it is difficult to calculate the values of $f(y) = x$ that are suitable to all y within the field of values [8].

Algorithms based on one-way functions are defined as mathematical functions of considerable complexity. At present, these cryptosystems are capable of addressing various information security challenges [11].

The mathematical foundations of encryption algorithms based on one-way functions are rooted in number theory, linear algebra, and algebraic structures, including groups, finite fields, rings, subgroups, and one-way functions that ensure privacy. The security of these algorithms relies on the fact that inverting one-way functions is an exceptionally difficult problem [10].

One-way functions based on algebraic structures with integer parameters add another layer of complexity to cryptography, particularly through degree parameters. This complexity opens up opportunities to improve existing algorithms and develop new crypto-resistant ones [12].

In many instances, current cryptographic algorithms can be viewed as special cases of algorithms based on parameter algebra.

The fundamental operations in parameter algebra are defined as follows:

1. Multiplication with parameter R: $a®b \equiv a + b + a * R * b \ (mod \ \ n)$ R can be called a coefficient or parameter in parametric algebra. When $R = 0$, this expression represents the addition operation in classical algebra.
2. Parametric inversion operator modulo n: $a^{\backslash -1} \equiv a * (1 + R * a)^{-1} \ (mod \ n)$, where -1 is the inverse modulo n and \-1 is the inverse modulo n with respect to the parameter R. If it is multiplied by a parameter, the result is zero $a® \ a^{\backslash -1} \equiv 0(mod \ n)$.

   In parametric algebra, 0 is considered the unit element and has the property $a®0 \equiv a(mod \ n)$.
3. The operation of raising to a level with the R parameter:

$a^{\backslash x+1} \equiv a * \sum_{i=0}^{i=x} F^{\ i} (mod \ n)$, bunda $F = 1 + R * a$.

To increase the degree of a parameter R more quickly, it is useful to use the following property of the algebra of parameters:

$$a^{\backslash x} \equiv ((1 + R * a)^x - 1) * R^{-1} \ (mod \ n)$$

It is very convenient to calculate one-way functions using the above operations, which allows you to create new cryptographic algorithms or improve existing ones [6].

SOLUTION OF THE PROBLEM

This paper presents a symmetric encryption algorithm based on a one-way function, offering security equivalent to the discrete logarithm problem in a finite space. As with all encryption algorithms, the proposed method involves key generation, encryption, and decryption processes.

Key generation. The sender is responsible for generating the modulus components (r, R), the multiplication factor R', and the base for the exponentiation g_1 using the key source, and transmitting them to the receiver via a secure channel. The key source subsequently generates the module $n_0 — R^r$, and the organization stage concludes with the transmission of the numbers n_0, R', g_1 to the cryptographic tool.

The process of key generation is outlined as follows:

1. An arbitrary 128-bit long odd modulus base R number is generated.
2. $n_0 \leftarrow R^r$ is calculated.
3. The generation of a private key, designated as k, is undertaken with a length of 256 bits.
4. It is accepted that bits 1 to 192 of key k are accepted into g_1, as: $g_1 \leftarrow k$.
5. If $GCD(g_1, R) = 1$, then the bit of k from $i = 64$ to $i = 192$ is accepted as x: $x \leftarrow k$:

   Else, return step 3.
6. If $GCD \ (g_1, k) \neq 1$ then, return to step 3.

Message encryption

1. To encrypt a message, it is first necessary to enter the parameter R, the module $n_0 \leftarrow R^r$, and the private keys g_1 and x.
2. The integer representation of the message M is entered.
3. $R_1 = R * x \ modn_0$.is calculated.
4. $M_1 = g_1{}^{\backslash M} mod \ n_0$ is calculated, where the diamultiplication parameter is equal to R_1.
5. $C = g_1{}^{\backslash M_1} mod \ n_0$ is calculated, where the diamultiplication parameter is equal to R_1.
6. The C ciphertext is printed.

Decryption of ciphertext

1. Initially, the input values are the number R, the private key g_1, the numbers x, the diamultiplication coefficient $R_1 = R * x \ modn_0$ and the numerical value of the ciphertext C.
2. $s(2)_2 = C * g_2{}^{-1} modR$ is calculated.
3. $s(2)_1 = R - s(2)_2 mod \ R$ is calculated.
4. $g_2{}^{\backslash s(2)_1} = g_2^{\backslash s(2)_1} modR^r$ is calculated.
5. $M_2 = \left(C®g_2{}^{\backslash s(2)_1}\right) * g_2{}^{-1} - s(2)_1 mod \ R^r$ is calculated.
6. $s(1)_2 = M_2 * g_1^{-1} mod \ R$ is calculated.
7. $s(1)_1 = R - s(1)_2$ is calculated.
8. $g_1^{\backslash s(1)_1} = g_1^{\backslash s(1)_1} mod \ R^r$ is calculated.
9. $M_1 = \left(M_2®g_1^{\backslash s(1)_1}\right) * g_1^{-1} - s(1)_1 mod \ R^r$ is calculated.
10. $s_2 = M_1 * g_1^{-1} mod \ R$ is calculated.
11. $s_1 = R - s_2$ is calculated.
12. $g_1{}^{\backslash s_1} = g_1^{\backslash s_1} mod \ R^r$ is calculated.
13. $M = \left(M_1®g_1^{\backslash s_1}\right) * g_1^{-1} - s_1 mod \ R^r$ plaintext is calculated.

Example. Let's look at the encryption and decryption of M=1194 plaintext using the algorithm presented.

Key generation:

Diamultiplication parameter $R = 11$.

Module index $r = 3$.

The number of bases is $l = 1$.

A random number $x = 1$ as the secret key.

Encryption and decryption module $n_0 = R^3 = 11^3 = 1331$.

The basis of the encryption $g_1 = 4$.

Message encryption:

Initial values: M=1194, $n_0 = 1331$, $g_1 = 4$, $R = 11$.

$M_1 = g_1{}^{\backslash M} mod\ n_0 = 4^{\backslash 1194} mod\ 1331 = 1245$ is calculated.

$C = g_1^{\backslash M_1} mod\ n_0 = 4^{\backslash 1245} mod\ 1331 = 558$ is calculated.

$C = 558$ ciphertext is printed.

Decryption of ciphertext:

Initial values: C=558, $n_0 = 1331$, $g_1 = 4$, $R = 11$

$$s(1)_1 = 11 - 558 * g_1^{-1}\ mod\ 11$$

$$s(1)_2 = \left(558®4^{\backslash s(1)_1}\right) * g_1^{-1} - s(1)_1\ mod\ 11^2 = 35$$

$$s(1)_3 = 11^2 - s(1)_2 mod\ 11^3 = 86$$

$$M_1 = D(558{,}4{,}11{,}3) mod\ 11^3 = \left(558®4^{\backslash s(1)_3}\right) * g_1^{-1} - s(1)_3 mod\ 11^3 = 1245$$

$$s_1 = 11 - 1245 - g_1^{-1}\ mod\ 11 = 5$$

$$s_2 = (1245®g_1^{-1}) - s_1\ mod\ 11^2 = 105$$

$$s_3 = 11^2 - s_2 = 16M = D(M_1, 4{,}11{,}3) mod\ 11^3 = \left(1245®4^{\backslash s_3}\right) * g_1^{-1} - s_3\ mod 11^3 = 1194$$

$M = 1194$ plaintext is printed.

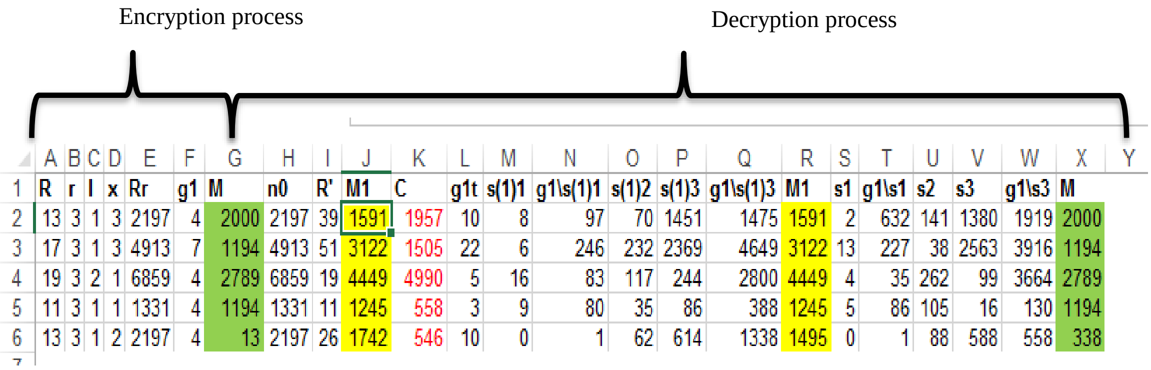
RESULTS AND DISCUSSION

The primary functions utilized in developing the algorithm include modular computation, parametric inversion, four-argument exponentiation, and parametric multiplication.

Two distinct methods were employed to practically address the implementation and verification of the algorithm.

The first method involves implementing the encryption algorithms within MS Excel office software, following a sequence of steps as part of an experiment (Figure 3). This approach is useful for verifying the calculation results and ensuring the algorithm executes correctly.

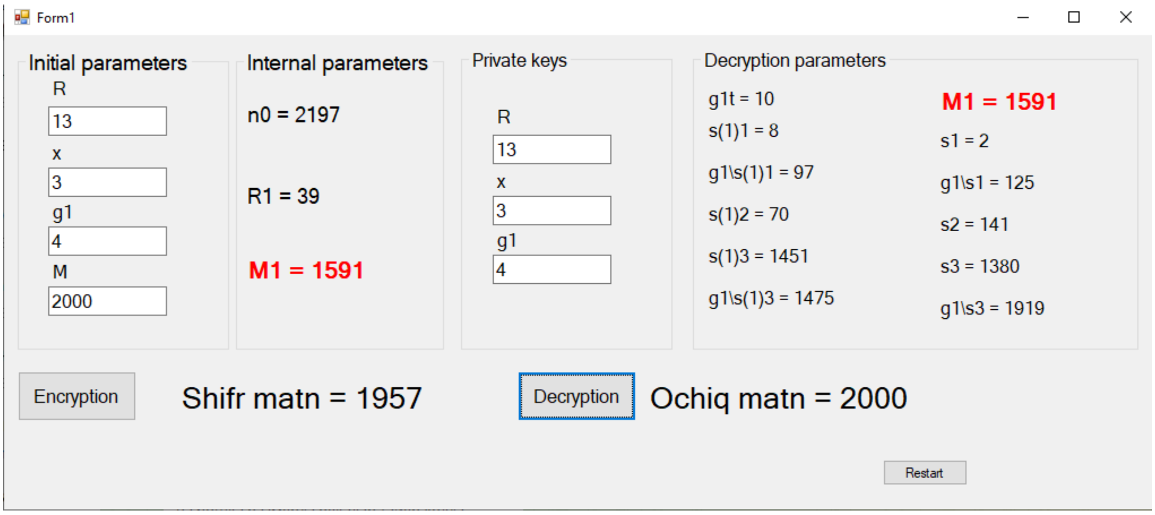
Most symmetric encryption algorithms feature an equal number of encryption and decryption steps. However, in a symmetric encryption algorithm based on a one-way function, the number of functional substitutions used for decryption is significantly greater than those used for encryption. This disparity complicates the process of recovering the plaintext without knowledge of the secret keys. Moreover, if the number of encryption base-level indicators and pairs in the general form of the algorithm is sufficiently increased, the difference between the encryption and decryption steps becomes even more pronounced.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
1	R	r	l	x	Rr	g1	M	n0	R'	M1	C	g1t	s(1)1	g1\s(1)1	s(1)2	s(1)3	g1\s(1)3	M1	s1	g1\s1	s2	s3	g1\s3	M	
2	13	3	1	3	2197	4	2000	2197	39	1591	1957	10	8	97	70	1451	1475	1591	2	632	141	1380	1919	2000	
3	17	3	1	3	4913	7	1194	4913	51	3122	1505	22	6	246	232	2369	4649	3122	13	227	38	2563	3916	1194	
4	19	3	2	1	6859	4	2789	6859	19	4449	4990	5	16	83	117	244	2800	4449	4	35	262	99	3664	2789	
5	11	3	1	1	1331	4	1194	1331	11	1245	558	3	9	80	35	86	388	1245	5	86	105	16	130	1194	
6	13	3	1	2	2197	4	13	2197	26	1742	546	10	0	1	62	614	1338	1495	0	1	88	588	558	338	

FIGURE 3. Formation of the algorithm in the Microsoft Excel program

The second method involves implementing the algorithm as a software product. This approach allows the properties and capabilities of the algorithm to be demonstrated more clearly. To verify the implementation of the encryption and decryption functions of the symmetric encryption algorithm based on a one-way function, a program was developed for a specific case of the algorithm with $l = 1$ and $r = 3$ (Figure 4).



Initial parameters
R
13
x
3
g1
4
M
2000

Internal parameters
n0 = 2197
R1 = 39
M1 = 1591

Private keys
R
13
x
3
g1
4

Decryption parameters
g1t = 10
s(1)1 = 8
g1\s(1)1 = 97
s(1)2 = 70
s(1)3 = 1451
g1\s(1)3 = 1475
M1 = 1591
s1 = 2
g1\s1 = 125
s2 = 141
s3 = 1380
g1\s3 = 1919

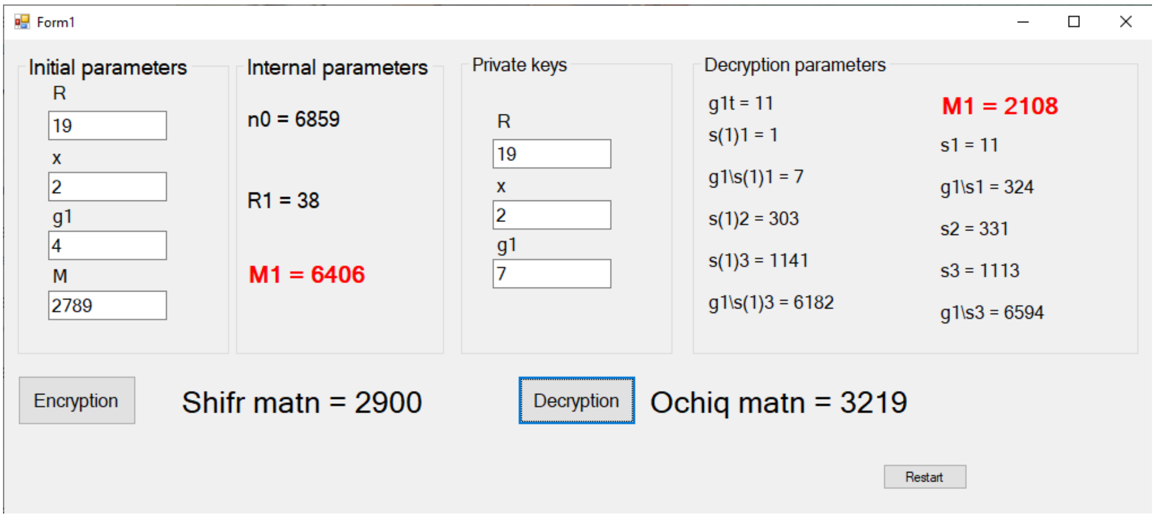
Shifr matn = 1957

Ochiq matn = 2000

FIGURE 4. Software interface

To test the program, the input parameter values from the spreadsheet shown in Figure 4 are entered into the program. These values are then compared with the functional substitutions in the program and the algorithm. The input parameters used for the test are R=13, x=3, g=4, and the plaintext M=2000.

By comparing the values in the table with those in the program, it is evident that they match. This confirms that the program functions, as shown in Figure 4, are working correctly.



Initial parameters
R
19
x
2
g1
4
M
2789

Internal parameters
n0 = 6859
R1 = 38
M1 = 6406

Private keys
R
19
x
2
g1
7

Decryption parameters
g1t = 11
s(1)1 = 1
g1\s(1)1 = 7
s(1)2 = 303
s(1)3 = 1141
g1\s(1)3 = 6182
M1 = 2108
s1 = 11
g1\s1 = 324
s2 = 331
s3 = 1113
g1\s3 = 6594

Shifr matn = 2900

Ochiq matn = 3219

FIGURE 5. Programm values on key change

If the key is changed while the program is running, the values of the reflections will adjust accordingly. Since the encryption and decryption secret parameters are interdependent, modifying any one of them will result in nearly all encryption parameters changing, which in turn alters the plaintext.

Figure 5 illustrates that when the encryption key $g1$ is modified, both the decryption parameters and the resulting plaintext differ from the original text.

CONCLUSION

Modern symmetric stream and block encryption algorithms are built on the mathematical principles of XOR operations, random number generation, and polynomial functions. These algorithms typically rely on Feistel networks, SPN (Substitution-Permutation Network) architectures, and Lai-Massey architectures, with random number generators driving the encryption processes.

This paper proposes a function-based symmetric encryption algorithm and presents the following results:

1. Development of a symmetric encryption algorithm featuring a degree parameter and discrete logarithm complexity.
2. Exploration of the application of parametric algebra operations to symmetric encryption algorithms.
3. Introduction of methods based on one-way functions, in addition to Feistel networks, SPN architectures, and Lai-Massey architectures, for generating symmetric encryption algorithms.

One-way functions, which are commonly used in public key cryptographic systems, are implemented in the proposed encryption algorithm for symmetric cryptography. This approach distinguishes it from traditional symmetric algorithms.

In the proposed symmetric encryption algorithm, the number of decryption steps is significantly greater than the number of encryption steps. These characteristics contribute to a high level of complexity in recovering the plaintext when the secret keys are unknown, ensuring robust cryptographic resistance.

Incorporating parametric algebra operations into symmetric encryption algorithms based on one-way functions introduces a new direction in cryptography. This advancement facilitates the development of new symmetric encryption algorithms using one-way functions and parametric algebra operations.

REFERENCES

1. Shriram P., Navghare N., Bhalerao A. Y. File Encryption Using AES and XOR Algorithm for Data Security //2024 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT). – IEEE, 2024. – C. 407-413.
2. Subhi R. M. Zeebaree, “FPGA Implementations for Data Encryption and Decryption via Concurrent and Parallel Computation: A Review”,ResearchGate, March 2021.
3. Achuthshankar A., Achuthshankar A. A novel symmetric cryptography algorithm for fast and secure encryption //2015 IEEE 9th International Conference on Intelligent Systems and Control (ISCO). – IEEE, 2015. – C. 1-6.
4. N. A. Advani and A. M. Gonsai, "Performance Analysis of Symmetric Encryption Algorithms for their Encryption and Decryption Time," *2019 6th International Conference on Computing for Sustainable Global Development (INDIACom)*, New Delhi, India, 2019, pp. 359-362
5. Kenekayoro Patrick T. One-way functions and public key cryptography //African Journal of Mathematics and Computer Science Research. – 2011. – T. 4. – №. 6. – C. 213-216.
6. Khasanov Kh. P. Methods and algorithms for creating cryptosystems based on improved diamatrices algebras and parametric algebras. – 2008.
7. Mulder V. et al. Trends in Data Protection and Encryption Technologies. – Springer Nature, 2023. – C. 262. https://doi.org/10.1007/978-3-031-33386-6
8. Akbarov D. E. Cryptographic methods of ensuring information security and their application // Uzbekistan Stamp. – 2009. – T. 432. 4
9. Mamirjon Turdimatov, Farrukh Mukhtarov, Sultonali Abdurakhmonov, Umidjon Khudoynazarov and Mastura Muminova. E3S Web of Conferences **389**, 07012 2023) https://doi.org/10.1051/e3sconf/202338907012 [CrossRef] [EDP Sciences] [Google Scholar
10. Akbarov D., Khasanov P., Khasanov H., Akhmedova O. Mathematical basics of cryptography. Study guide. - Tashkent, 2010 - 210 p.
11. Schneier B. Applied cryptography: protocols, algorithms, and source code in C. – John WileyJ & sons, 2007.
12. Khudoynazarov U.U. "Comparative Analysis of Modern Encryption Algorithms". Republican seminar: "Actual problems using electronic digital signature". Collection of theses and documents. Tashkent, May 20, 2016