IMPROVING THE RSA PUBLIC-KEY ENCRYPTION ALGORITHM BASED ON PARAMETRIC ALGEBRA

Khudoykulov Zarifjon
PhD, Tashkent University of Information Technologies named after Muhammad al Khorazmi,Tashkent, Uzbekistan
zarif.khudoykulov@tuit.uz
Khudoynazarov Umidjon
PhD student, Tashkent University of Information Technologies named after Muhammad al Khorazmi,Tashkent, Uzbekistan
umidjonxudoynazarov@gmail.com

Abstract. This article presents a modification of the RSA public-key cryptographic algorithm based on parametric algebraic operations. The application of parametric multiplication, inversion, and exponentiation operations enhances the algorithm's exponentiation process through an additional parameter, thereby increasing its cryptographic resilience. The proposed closed-form expression for parametric exponentiation optimizes the computation process and aligns the overall complexity of the algorithm with that of standard RSA. The practical results confirm the functional correctness and efficiency of the improved approach.

Keywords. RSA algorithm, public-key cryptography, parametric algebra, modular arithmetic, parametric exponentiation, cryptographic resistance.

Introduction. In information and communication systems, the protection of large volumes of data has become one of the urgent issues of today, and the requirements for the security and efficiency of public-key cryptosystems are steadily increasing. Although the standard RSA algorithm is based on the computational hardness of factorization, increasing key lengths to ensure a high level of cryptographic robustness leads to a significant slowdown in the encryption and decryption processes [8]. The emergence of modern high-performance computing machines makes it an important scientific task to enhance security while maintaining speed by optimizing the mathematical foundations of the RSA algorithm.

Parametric algebra is a complex algebraic structure, and the introduction of an additional parameter into arithmetic operations forms a new mathematical complexity [4]. Applying multiplication, inversion, and exponentiation operations performed by means of this algebra increases the mathematical complexity of cryptographic functions and creates additional opportunities for improving existing encryption systems. Although in some studies results have been obtained on modifying symmetric and asymmetric algorithms on the basis of parametric algebra, the issue of reconstructing the modular exponentiation step in the RSA algorithm on the basis of parametric algebra has not been sufficiently covered.

The purpose of this study is to modify the exponentiation mechanism of the RSA algorithm by using parametric algebra operations, to introduce additional mathematical complexity through parametric exponentiation, and also to evaluate the computational efficiency and functional correctness of the proposed approach. In addition, it is intended to bring the overall computational complexity of the algorithm closer to the level of the standard RSA algorithm by replacing the iterative structure of parametric exponentiation operations with a closed-form expression.

The practical experiments carried out demonstrate that the RSA algorithm enhanced using parametric algebra operates correctly and achieves computational efficiency comparable to that of the standard RSA algorithm. These results confirm that the parametric-algebra–based approach is feasible for use in practical cryptographic systems.

Literature Review. The analysis of the literature related to the topic shows that there exists a wide range of research in national and foreign sources on the mathematical foundations of information security and cryptographic systems. In the works of G‘aniyev and Tashev [9], as well as Akbarov D.E. [1], the theoretical foundations of cryptographic protection methods and information security are described, while Xasanov X.P. [4] has thoroughly studied the algebraic foundations of parametric algebra operations and their application to cryptosystems. Fundamental concepts regarding the RSA algorithm, modular exponentiation, and cryptographic complexity theory are presented in detail by Stallings [6] and by Menezes, Van Oorschot, and Vanstone [5]. Modern modular operations and exponentiation mechanisms have been analyzed by Koç and collaborators [6], and in the mathematical justification of parametric operations, the book ‘Concrete Mathematics’ by Graham, Knuth, and Patashnik [7] plays an important role. Foreign research by Boneh [3] indicates the possibilities of improving RSA and the potential security threats that may arise.

Problem Statement. Based on the requirements for data protection in modern information systems, increasing the computational efficiency and stability of public-key cryptosystems is one of the important scientific tasks. The security of the standard RSA algorithm relies on increasing the key length, which causes the modular exponentiation operations to slow down and limits efficiency in practical systems [2]. Parametric algebra, by offering an extended form of classical arithmetic operations, serves as a promising basis for improving the mathematical structure of the RSA algorithm and introducing an additional complexity component [5]. Therefore, optimizing the RSA algorithm by relying on parametric algebra operations is considered relevant from the perspective of enhancing its security and practical efficiency.

RSA encryption algorithm. The RSA algorithm is a public-key encryption algorithm and consists of the processes of generating keys, encrypting data, and decrypting data [2]

Key generation algorithm

1. Prime numbers p and $q$of sufficiently large size are generated
2. The modulus is computed as $N = p * q$
3. Next, the Euler totient of n is calculated as $\varphi(n) = (n - 1) * (q - 1)$
4. An integer e is then selected such that it is coprime with $\varphi(n)$ and satisfies the condition (1< e <φ(n))

5. Then the number d is computed, which is the multiplicative inverse of e modulo $\varphi(n)$

6. Finally, the pair $n.e$ is defined as the public key, while d form the private key [2]

Encryption

1. The open message M is divided into blocks that are smaller than N

2. The ciphertext is computed through the $C_1 = M^e m\,od\, N$

Decryption

1. Plaintext M is recovered through the $M = (C_1)^d \, mod\, N$

Today, supercomputers that perform a very large number of operations per second have extremely large resources, and to ensure the above-mentioned resistance, it becomes necessary to sharply increase the key length. This, in turn, slows down the processes of key generation, encryption, and decryption of the cryptosystem.

Taking the above conclusion into account, it is intended to enhance the RSA public-key cryptographic algorithm by employing parametric multiplication, inversion, and exponentiation operations on integers.

The following section examines the fundamental parametric operations of parametric algebra—namely, parametric multiplication, parametric inversion, and parametric exponentiation.

Parametric algebra. It is known that algebraic operations based on parametric algebra have been developed by the scientists of our country. This parametric algebra leads to a mathematical problem of finding a new degree parameter in a sufficiently finite field.

Several modern symmetric and asymmetric encryption algorithms have been improved using parametric algebra operations, and new encryption algorithms have been developed [4].

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 $^{\backslash -1}$ 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 [4].

Solution of the problem. Through the given parametric algebra, computing one-way functions is very convenient, and this makes it possible to create new cryptographic algorithms or to improve existing cryptographic algorithms by means of these operations [5].

One-way cryptographic functions based on integer parametric algebraic structures bring about, in addition to the above-mentioned complexities, a complexity based on an additional degree parameter in the field of cryptography. This, in turn, makes it possible to further improve existing algorithms and to develop new crypto resistant algorithms [5].

The difference between the RSA algorithm improved by means of parametric algebra and the standard RSA algorithm is that the ordinary exponentiation operation in the algorithm is replaced with the R-parametric exponentiation operation. The encryption and decryption operations in the algorithm are carried out by parametric exponentiation.

The RSA public-key encryption algorithm based on parametric algebra also includes three processes.

Key generation algorithm

1. Prime numbers p and $q$of sufficiently large size are generated
2. The modulus is computed as $N = p * q$
3. Next, the Euler totient of n is calculated as $\varphi(n) = (n - 1) * (q - 1)$
4. The R parameter is generated.
5. An integer e is then selected such that it is coprime with $\varphi(n)$ and satisfies the condition ($1< e <\varphi(n)$)
6. Then the number d is computed, which is the multiplicative inverse of e modulo $\varphi(n)$
7. Finally, the pair n, e is defined as the public key, while d, R form the private key [2]

Encryption

1. The open message M is divided into blocks that are smaller than N
2. The ciphertext is computed through the $C_1 = M^{\backslash e} m\,od\,N$

Decryption

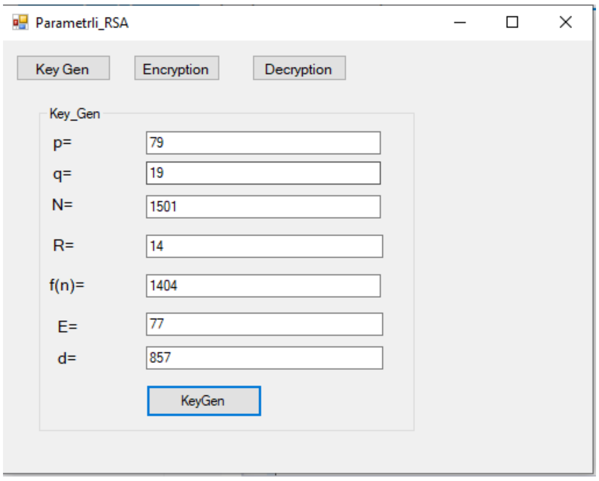
1. Plaintext M is recovered through the $M = (C_1)^{\backslash d}\, mod\, N$

In this approach, the parameter R serves as an additional layer of security or as a mechanism for adaptive encryption. Consequently, the method becomes more complex than standard RSA, as it introduces an extra mathematical challenge: alongside the factorization problem, an adversary must also solve the problem of determining the exponent parameter. This added complexity strengthens the overall security of the system.

To validate the encryption transformations of the proposed algorithm in practice, a prototype implementation was developed (Figure 1). The program was executed multiple times using input values provided in an electronic spreadsheet to ensure correctness and consistency.

The program consists of three components: key generation, message encryption, and message decryption.

Key generation is performed using random number generators in accordance with the specified conditions.

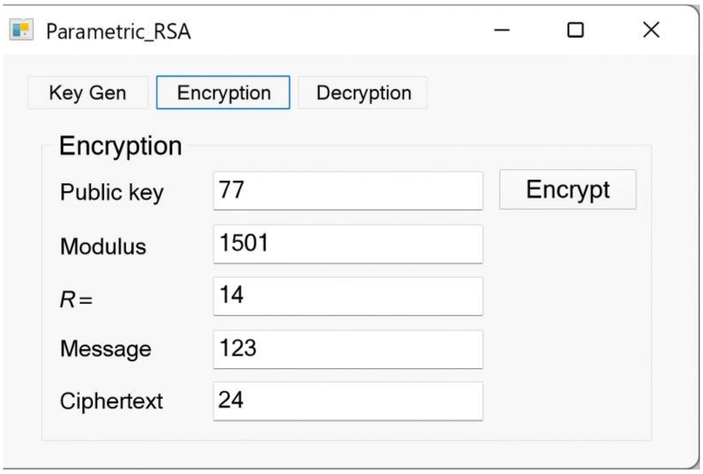


p= 79
q= 19
N= 1501
R= 14
f(n)= 1404
E= 77
d= 857

Figure 1. The key generation process in the RSA algorithm based on parametric algebra.

Unlike the standard RSA algorithm, in the RSA algorithm based on parametric algebra the secret keys *d* and the parameter *R* are not disclosed. The key *d* is taken as the private key, while the parameter *R* is known only between the message exchangers.

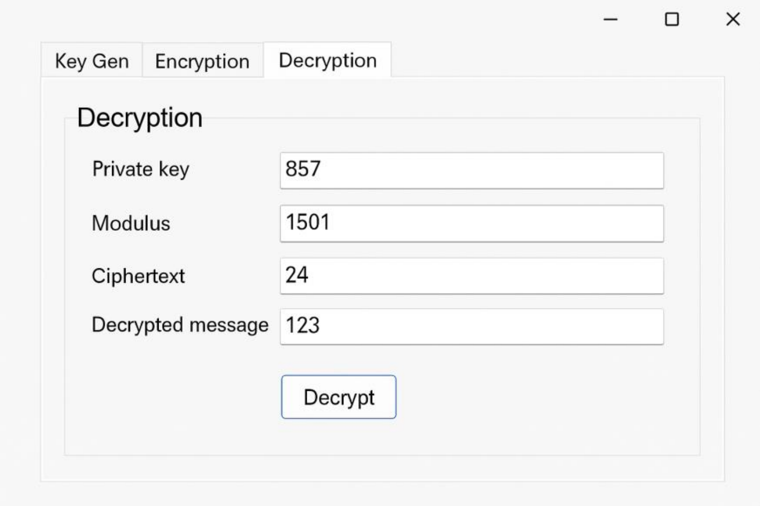
The process of encrypting the message was also carried out on the basis of parametric algebra. In this case, *C* and the ciphertexts were obtained (Figure 2).



Public key 77
Modulus 1501
R= 14
Message 123
Ciphertext 24

Figure 2. The encryption process in the RSA algorithm enhanced with parametric algebra.

In the decryption process as well, the correspondence between the plaintext and the decrypted text is shown (Figure 3).



Decryption

Private key 857

Modulus 1501

Ciphertext 24

Decrypted message 123

Figure 3. The data decryption process in the RSA algorithm based on parametric algebra.

The results obtained for small values in the program indicate that the functional transformations and the algorithm’s execution sequence are being performed correctly.

The iterative parametric exponentiation operation used in the algorithm has a computational complexity of $O(k)$, where k denotes the exponent value. In the RSA algorithm, however, the exponents—particularly the private key exponent d—are typically very large (for example, in the range of 512 to 2048 bits). For this reason, using an iterative parametric exponentiation function is not computationally efficient in practice, since the number of repeated operations increases proportionally with the exponent itself [7].

The parametric exponentiation operation was transformed using special algebraic substitutions, which significantly increased the speed of both encryption and decryption. As a result, the algorithm achieved a performance level nearly equivalent to the encryption and decryption speeds of the standard RSA algorithm.

The operations of the accelerated parametric exponentiation algorithm are as follows:

Fast parametric exponentiation

1. $A = 1 + R \cdot a \; mod \; n$ s computed.
2. $P = A^k \; mod \; n$ s computed.
3. The result is obtained as $Diadaraja = (P - 1) \cdot R^{-1} mod \; n$

The accelerated parametric exponentiation algorithm converts the iterative formulation into a closed-form expression and evaluates the result using the following equation:

$$Diadaraja(R, n, a, k) = \frac{(1 + Rk)^k - 1}{R} \; modn$$

The advantage of this expression lies in the fact that the parametric exponentiation operation is no longer dependent on recursive or iterative steps. All computations are reduced to performing a modular exponentiation of the form $(1 + Ra)^k \; modn$. As a result, the standard RSA algorithm enhanced with fast parametric exponentiation achieves a computational complexity nearly equivalent to that of the standard RSA algorithm, namely $O(logk)$ [8].

Results and Analysis. The RSA algorithm enhanced with fast parametric exponentiation was compared with the standard RSA algorithm in terms of mathematical complexity and execution speed(table.1).

Table 1. Complexity analysis of the algorithms

Algorithm	**Mathematical operation**	**Computational complexity**	Note
Standart RSA	$m^e \ mod \ n$, $c^d mod \ n$	$O(log \ k)$ modular multiplication	Efficient with binary exponentiation
Parametric RSA	$(1 + Rm)^k$ and subsequent operations	$(log \ k)$ modular multiplication	Same complexity as modular exponentiation; only 1 additional multiplication and modular inversion are used.

Table 2. Comparison of the characteristics of the algorithms

Characteristic	**Standard RSA**	**Parametric RSA**
Main operation	Modular exponentiation	Modular exponentiation
Mathematical basis	$m^k \ mod \ n$	$(1 + Rm)^k - 1 \ \ modn$
Complexity	$O(\log k)$	$O(\log k)$
Algorithm speed	Fast	Nearly the same as standard RSA

As shown in **Table 2**, the parametric RSA algorithm preserves the computational complexity of standard RSA while introducing an alternative mathematical structure for exponentiation.

The experimental results indicate that introducing parametric exponentiation does not significantly affect overall performance, as both algorithms exhibit nearly identical timing behavior across different key lengths (Figure.4)

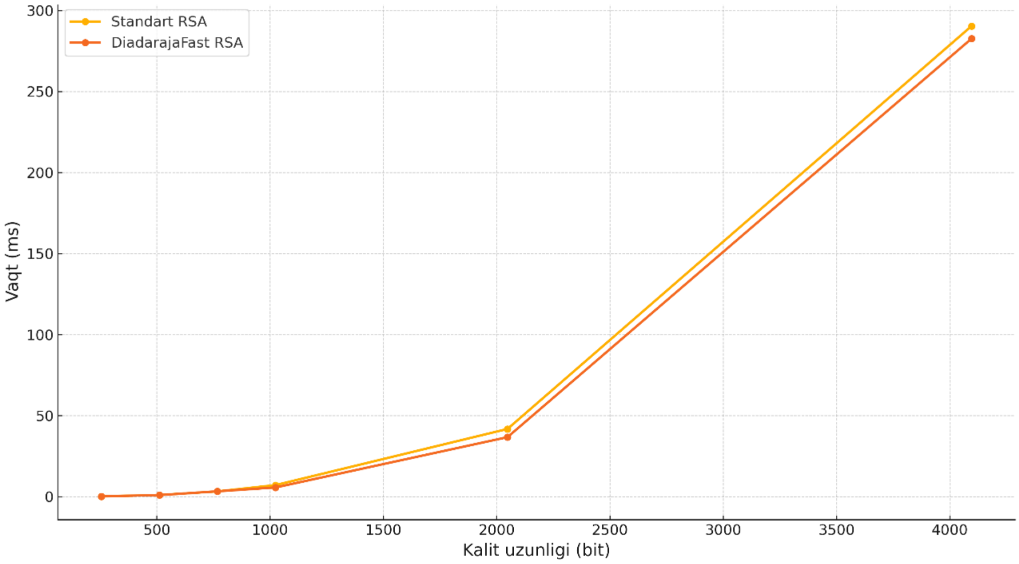


Figure 4. Dependence of the execution speed of the Standard RSA and Parametric RSA algorithms on key length.

Conclusion. In this study, the RSA public-key cryptographic algorithm was modified through the incorporation of parametric algebraic operations, and the potential for introducing additional mathematical complexity into its exponentiation stage was examined. The algebraic properties of parametric multiplication, inversion, and parametric exponentiation were analyzed, demonstrating that these operations provide a sound mathematical basis for enhancing the modular exponentiation process of RSA through the introduction of an additional parameter.

The proposed algorithm based on fast parametric exponentiation brings the computational complexity of the modified scheme close to that of standard RSA, while the introduction of an additional parameter enhances its cryptographic resistance. Practical experiments confirmed that the modified RSA algorithm operates correctly from a functional perspective and that the parametric algebra–based modifications do not negatively impact its efficiency. The obtained results indicate that the improved RSA algorithm can be applied in practical cryptographic systems and may serve as a scientific foundation for future research on developing other cryptosystems based on parametric algebra and evaluating them within formal security models. Furthermore, subsequent studies may explore the homomorphic properties of the RSA algorithm enhanced through parametric algebra, which holds significant importance for advanced cryptographic applications.”

REFERENCES:

1. *Акбаров Давлатали Егиталиевич, Хасанов Пўлат Фаттохович, Хасанов Хислат Пўлатович, Ахмедова Ойдин Пўлатовна. (т.ф.д., профессор П.Ф. Хасанов таҳрири остида)* "Криптографиянинг математик асослари" – ТОШКЕНТ 2010. 210 б

2. Rivest, R. L., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, *21*(2), 120-126.

3. Boneh D. *Twenty Years of Attacks on the RSA Cryptosystem.* Stanford University, 1999.

4. *Xasanov X.P.* Takomillashgan diamatristalar algebralari va parametrli algebra asosida kriptotizimlar yaratish usullari va algoritmlari.– Toshkent, 2008. -208 b.

5. *Хасанов Хислат Пулатович*. Мавжуд криптоалгоритмларни параметрлар алгебраси асосида такомиллаштиришнинг умумий усули. http://ru.infocom.uz/wp-content/download/information_security_24112005_17.html

6. *William Stallings*. Cryptography and Network Security: Principles and Practice, Sixth edition. Prentice Hall, 2014.

7. Graham, R. L., & Knuth, D. E. (1994). O: Patashnik, Concrete Mathematics, A Foundation for Computer Science, AddisonWesley. *Reading, MA.*

8. Menezes, A. J., Van Oorschot, P. C., & Vanstone, S. A. (2018). *Handbook of applied cryptography*. CRC press.

9. S.K. Ganiyev, M.M. Karimov, K.A. Tashev. Axborot xavfsizligi. -T.: «Fan va texnologiya», 2017,372 bet.