Transactions on Computer Science and Intelligent Systems Research
ISSN: 2960-1800, eISSN: 2960-2238 | Volume 5
AIDML 2024

Analyzing the Research and Application of the RSA Cryptosystem in the Context of Digital Signatures

Mashuo Ding

School of Cyber Science and Engineering, Huazhong University of Science and Technology, Wuhan, China

U202112080@hust.edu.cn

Abstract. The article commences by laying the foundational knowledge of public key cryptographic systems, with a focus on RSA encryption and digital signature technology. It then utilizes RSA encryption as a paradigm to elucidate the application of this encryption in digital signature schemes, offering an algorithmic perspective. A critical link is established between the security of digital signatures and the robustness of RSA public key cryptographic systems. Subsequently, the article presents a concise overview of the potential threats facing RSA encryption and proposes pertinent strategies to mitigate these security challenges. It culminates in a synthesis of three distinct approaches, encompassing both algorithmic and hardware aspects, to augment the efficiency of modular exponentiation – a key operation in RSA encryption. Since the inception of the RSA public key cryptosystem, it has been subject to continuous refinement, standardization, and practical deployment. Enhancing the computational efficiency and security of RSA encryption remains a pivotal area of research and discussion within the cryptographic community. Furthermore, with the progressive advancements in quantum theory, traditional public key systems, including RSA and elliptic curve cryptography, confront emerging security vulnerabilities. Thus, the pursuit of post-quantum cryptography, capable of withstanding quantum attacks, represents a frontier scientific endeavor in the field of cryptography.

Keywords: Public Key Cryptosystem; RSA Encryption; Digital Signature.

1. Introduction

The digital signature system, a subset of public key cryptography, serves as a pivotal cryptographic technology in validating the authenticity, integrity, and non-repudiation of information sources. In the age of digital communication, particularly over the Internet, digital signatures are instrumental in asserting identity and safeguarding information security. Rooted in public key cryptography, digital signatures are designed to confirm the authenticity, integrity, and non-repudiation of digital data. They amalgamate hash functions, asymmetric encryption, and digital certificates to prevent data tampering during transmission and storage, and to authenticate data sources. This technology is critical in an era where data breaches and cyber threats are rampant, offering a robust layer of protection for digital interactions. This article delves into the application and execution of the RSA encryption algorithm within digital signatures. It undertakes a comprehensive analysis of the algorithm's security and computational efficacy. Moreover, it provides an extensive overview of the operational mechanisms of digital signatures, emphasizing the RSA algorithm's role in ensuring data security in various digital transactions and communications. Furthermore, the article explores innovative methods to enhance the computational efficiency of the RSA algorithm. This includes examining alternative approaches to key generation and encryption processes, exploring potential optimizations in the algorithm's core operations, and considering the implications of emerging technologies on the future of RSA-based digital signatures..

2. Background Knowledge

2.1. Public Key Cryptosystem

Public key cryptography is a cryptosystem that utilizes asymmetric key algorithms, where the encryption and decryption keys are different. In this system, the cryptographic algorithm is public, and it generates different encryption and decryption keys. The encryption key is openly available, while the decryption key is kept confidential, and it is challenging and very difficult to derive the decryption key based on the encryption key.

Most public key cryptographic algorithms are based on specific mathematical problems. For instance, the security of the RSA public key cryptographic algorithm relies on the difficulty of the factorization problem of large integer. Elliptic curve cryptography, on the other hand, is based on the complexity of the discrete logarithm problem. Additionally, the NTRU public key cryptographic system relies on the computational complexity of lattice-based shortest vector problems and closest vector problems.

These public key cryptographic algorithms rely on a particular mathematical challenge known as a one-way trapdoor function, as illustrated in Fig. 1 [1]. In the context of this function $y = f(x)$, if x is known, it is easy to calculate y, but if y is known, calculating $x = f^{-1}(y)$ is extremely difficult. Additionally, there exists a trapdoor z so that knowing z allows for the easy calculation of $x = f^{-1}(y)$, while not knowing z makes it impossible to calculate $x = f^{-1}(y)$.

This article focuses on introducing the RSA encryption algorithm within the framework of public key cryptography and explains the usage of this public key cryptographic technique in digital signatures.

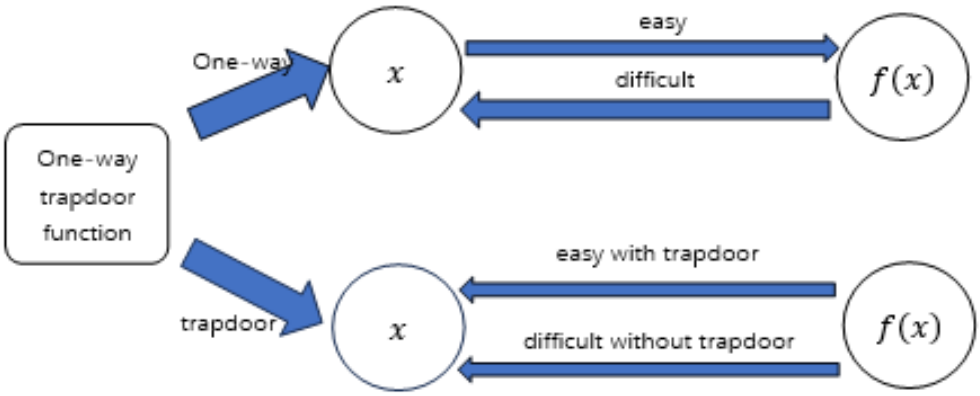

Fig. 1 One-way trapdoor function (Photo/Picture credit: Original).

2.2. RSA Cryptosystem

One of the earliest public-key cryptosystems, RSA (Rivest-Shamir-Adleman) is frequently used for safe data transfer. In RSA, the public key is used for encryption, and the corresponding private key is used for decryption. Suppose that m is plaintext, c is ciphertext. The RSA algorithm involves the following steps. As shown in Fig. 1:

Determine the key

Select two large prime numbers p and q. And compute $n = p \times q$.

Compute $\varphi(n) = (p-1)(q-1)$.

Choose an integer e s. t. $1 < e < \varphi(n)$ and $gcd(e, \varphi(n)) = 1$,then choose {e, n} as the public key

Compute d s.t. $e \cdot d = 1 \bmod (\varphi(n))$and choose {d, n} as the private key.

Encryption.

The sender uses the recipient's public key {e, n} to compute the encryption of the plaintext m by calculating $c = m^e (mod\ n)$.

Decryption.

The recipient decrypts the ciphertext C with their private key {d, n} to recover the original plaintext by computing $m = c^d \ (mod\ n)$.

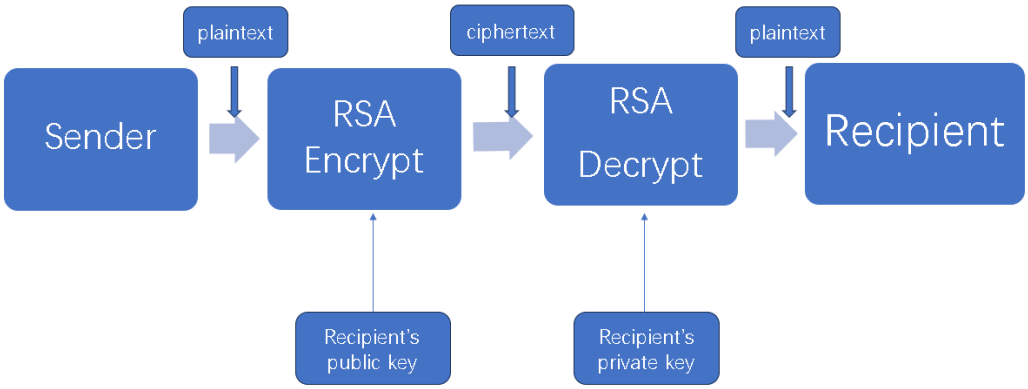

Fig. 2 RSA encryption/decryption process (Photo/Picture credit: Original).

2.3. Digital Signature Technology

The digital signature technology is based on the foundation of public key cryptosystem. For a specific public key cryptographic algorithm, digital signature technology involves the following steps, as shown in Fig. 2. Create message digest: The sender uses a secure hash function to compute the message, creating a message digest [2]. Encrypt message digest: The digital signature is then created by encrypting the message digest with the sender's private key. Send the signature and original message: The digital signature is then attached to the original message. The sender sends the signed message to the recipient.

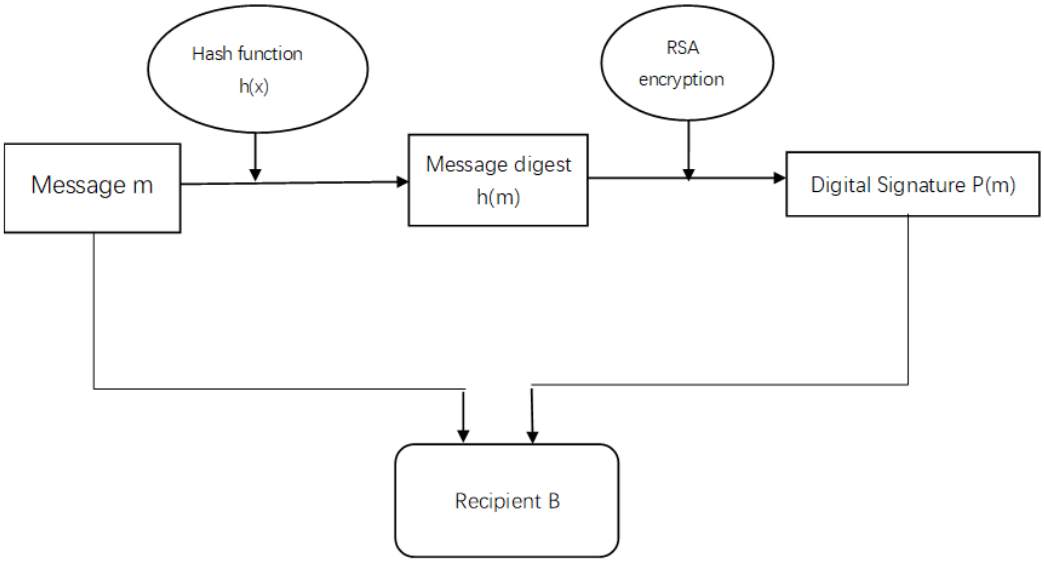

Fig. 3 Digital Signature process (Photo/Picture credit: Original).

Verification of digital signature involves the following steps, as shown in Fig. 4.

Decrypt the digital signature: The original hash value can be obtained by the recipient by decrypting the digital signature using the sender's public key.

Hashing: In order to generate the message digest, the recipient uses the same secure hash algorithm to determine the hash value of the received message.

Comparing: The recipient compares the original hash value with the message digest. If they match, the signature is verified, and the integrity and authenticity of the message are confirmed

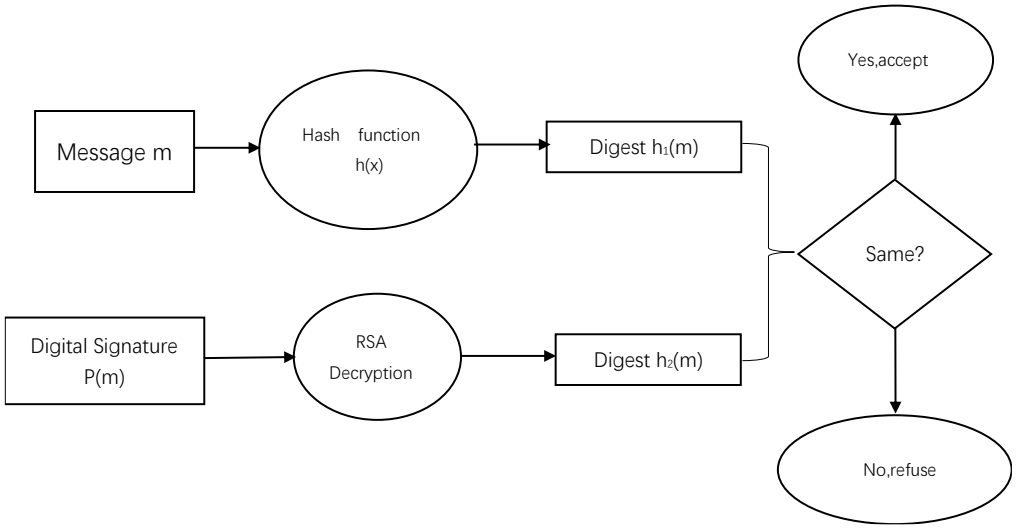

Fig. 4 The process of verifying the digital signature (Photo/Picture credit: Original).

Digital signatures are widely used in electronic identity authentication, file integrity verification, electronic payments, and various other fields due to the following characteristics: Authentication: It can verify the integrity of the data. Due to the avalanche effect of the hash function, even a minor change in the input data can cause a significant change in the hash output. Even if only 1 bit of data in a file is tampered with, the recipient, by applying the hash function to the message, will obtain a drastically different result compared to the decrypted digest of the received digital signature. Non-repudiation: The sender cannot deny having sent a message because only the sender possesses the private key, preventing others from using the sender's private key to encrypt the digest (unless the private key leaks or is stolen).

3. The application of RSA in digital signature

3.1. The implementation of RSA based on traditional digital signature

The implementation of RSA based on traditional digital signature involves the following steps:

Generate signature and verification parameters [3].

Step1 Signer A selects two large prime numbers p and q confidentially, then compute $n = p \times q$ and $\varphi(n) = (p-1)(q-1)$.

Step2 Signer A selects e, d∈Zn, s. t. the following principles.

$$gcd\ (e, \varphi(n)) = 1 \tag{1}$$

$$ed = 1\ (mod\ \varphi(n)\ ) \tag{2}$$

Step3 Signer A publishes parameter pair (e, n) as public key and saves the pair {p, q, d, φ(n)} as private key.

Step4 Signer A select a series of secure hash functions to create hash values [4].

Signature Algorithm

Step1The signer A will encode the relevant information of the document m and generate the message digest h(m) through the hash function h(x).

Step2Signer A uses the private key d to encrypt the message digest h(m) with RSA, generating the digital signature P(m)

Step3Signer A sends the digital signature and the message {m, P(m)} to recipient B.

Verification Algorithm.

Step1 Recipient B uses the public key e, which is published by signer A, to decrypt P(m) using the RSA algorithm to obtain h1(m).

Step2 Then, using the public hash function, recipient B encrypts the message m to obtain h2(m).

Step3 Recipient B verifies whether h1(m) is equal to h2(m). If they are equal, the file is immediately accepted; otherwise, the file and signature are rejected.

3.2. Implementation of RSA Based on Group Signature

The concept of group signatures was first introduced by Chaum and Heyst in 1991 [5]. In a group signature, any member of a group can sign a message on behalf of the entire group in an anonymous manner. Similar to other digital signatures, group signatures can be verified by the group's public key.

The group collectively selects large prime numbers p and q, and computes$n = pq$. Then, a value for e is chosen and d is computed such that $ed = 1(mod\ \varphi(n))$, where φ(n) is the Euler's totient function. The values of n and e are made public, while p, q and d are kept confidential. Here, n and e serve as the public key of the group center, and d serves as the private key of the group center. The group center then selects a hash function h, chooses an integer g from the group G, and publishes h, g, the group public key n, and e as system-wide public parameters. Each member of the group has their own keys (x, y) such that $g^{xy} = g \quad (mod(n))$.

Member A uses the key (x,g^y)to sign the message m as follows:

Firstly, a random positive integer l is chosen such that l<p-1.

Compute the positive integers r, s and y, where $r = g^l\ mod\ p$, $y' = (g^y)^{\,l} \quad mod\ p$, $s = l^{-1}h(m)x_i\ mod(\ p\ -\ 1)$. This results in the signature σ = (r, y, s).

Member A sends the signature σ to recipient B.

The group signature verification process is as follows:

$$\text{Letl-1=k mod (p)} \tag{3}$$

When member B receives the signature, they verify the equation $(y')^s\ (mod\ p) = g^{kxy\ h(m)l}(mod\ p) = g^{h(m)}(mod\ p)$.

If this equation holds true, then σ is a valid signature, and recipient B accepts the message m; otherwise, recipient B rejects the message m.

3.3. Implementation of RSA in practical threshold signature

Combining threshold secret sharing technology with digital signatures, threshold signatures represent a significant subset of digital signatures. In 1991, Desmedt-Frankel first proposed the (t, n) threshold signature scheme. This threshold signature scheme refers to a group of n members with a pair of public and private keys. Within this group, any combination of t or more legitimate and honest members can collectively sign on behalf of the group using the group's private key, and anyone can verify the signature using the group's public key. Here, t represents the threshold value, indicating that only t or more legitimate members can collectively sign on behalf of the group.

The first implementation of threshold signatures based on the RSA algorithm was proposed by IBM Laboratories [6].

The process of this algorithm is shown in Fig. 5.

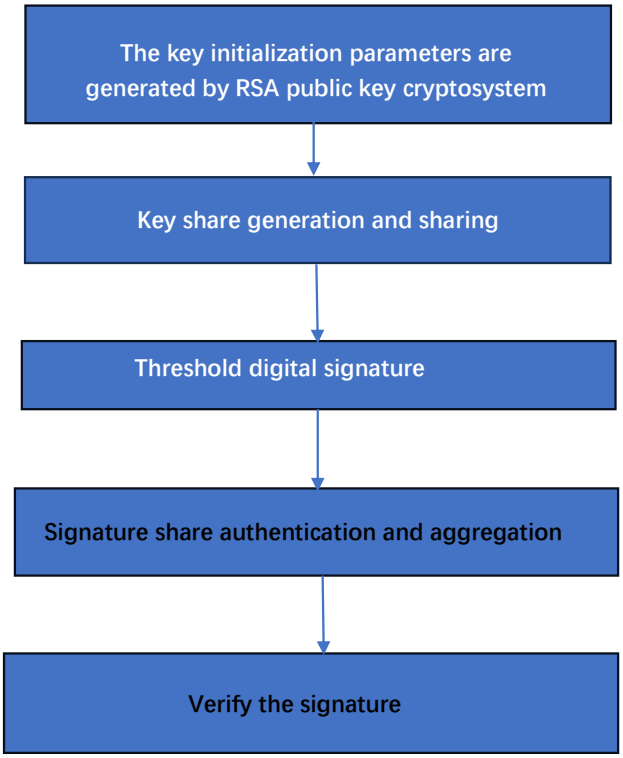

Fig. 5 The implementation of RSA in threshold signature (Photo/Picture credit: Original).

The implementation of RSA in threshold signature involves the following steps:

System initialization: Utilize the RSA public key cryptosystem to generate p, q, and e. m is a parameter generated from p and q, setting the stage for subsequent key sharing.

Key sharing: Participant i's private key, si, is generated through a polynomial f(i) mod m. Additionally, the corresponding verification key v is a randomly chosen value, where $v_i = v^{s_i}(mod\ n)$.

Generation of threshold signature shares: Each participant generates the individual signature shares xi based on a hash function H. c is data generated through a hash function, and z = si + r, where r is a randomly chosen number.

$$v' = v^r, x' = \tilde{x}^r, c = H'(v, \tilde{x}, v_i, x_i^2, v', x'), z = s_i + r \tag{4}$$

To verify the correctness of the signature share, equation 5 is checked for validity.

$$c = H'(v, \tilde{x}, v_i, x_i^2, v^z v_i^{-c}, \tilde{x}^z x_i^{-2c}) \tag{5}$$

Combining signature shares involves first verifying each individual signature share, ensuring that the valid signature shares are not less than the threshold value. Then, a combined signature y is generated through certain mathematical derivations.

Signature verification: Calculate $x = y^e\ (mod\ n)$ to determine its validity.

4. The security and efficiency of RSA in digital signature

Because the security of RSA-based digital signatures depends on the security of the RSA cryptosystem, enhancing the security of RSA public-key cryptography is an important direction for the development of digital signature technology. Similarly, improving the efficiency of RSA modular exponentiation is also an important research area for the application of RSA in digital signatures. This section will analyze the security and efficiency of RSA cryptography and summarize some methods for enhancing security and efficiency.

4.1. Security analysis of RSA cryptography

Based on background knowledge, we can understand that the security of the RSA public-key cryptosystem is based on the difficult mathematical problem of "integer factorization of large numbers." The trapdoor of breaking the RSA public-key cryptosystem lies in the factorization $N = pq$.

Currently, RSA cryptography is susceptible to the following types of attacks on its security:

The attack in computation $\varphi(n)$: In, a method was proposed. Assuming that a password attacker can calculate $\varphi(n)$ in some way, the attacker can easily factorize the factor of n by constructing a quadratic equation with n and $\varphi(n)$ [7].

Common modulus attack: Suppose we transmit a plaintext m to two different recipients, and the attacker intercepts two sets of public keys and ciphertexts (e1, c1) and (e2, c2) under the same modulus n, where e1 and e2 are coprime. Through mathematical deduction and calculation, the attacker can obtain the plaintext message m.

Low decryption exponent attack: Due to the small value of the private key d, an attack can use the LLL algorithm and knowledge of lattice theory to calculate the private key d. In previous research on the RSA cryptosystem, the attack methods and techniques used by Wiener and Ernst have established an upper bound for d at $N^{0.25}$, while in, this upper bound has been increased to $N^{0.292}$ [8,9].

Factorization-based attack: When the large prime numbers p and q are not significantly different or their values are very close, it can lead to security issues in RSA [10]. Fermat's factorization method or Pollard's p-1 factorization method can also be utilized to factorize p and q when they are not significantly different.

Attack based on quantum computing: Quantum computers can exploit special principles of quantum mechanics to solve certain mathematical problems faster than traditional computers. Currently, the two most threatening types of quantum algorithms to cryptography are the Shor's algorithm and Grover's search algorithm. Grover's search algorithm is a general quantum search algorithm that provides a square root speedup for a large class of search problems, effectively halving the key length and thereby posing a threat to existing cryptographic systems [11].

So, if we can enhance the security of RSA parameter selection and ensure the confidentiality of the private key, the security of RSA in its implementation process can be improved

4.2. RSA efficiency analysis and improvement method

The core operation of RSA algorithm for encryption and decryption is modular exponentiation, which essentially involves the CPU performing division with remainder at the hardware level. However, the division operation in computers is relatively slower and less efficient than other arithmetic operations. Therefore, a large amount of multiplication and division operations with large integers in RSA cryptography is the reason for the low efficiency of RSA operations.

In the process of implementing RSA, several methods are commonly used to improve the computational efficiency of RSA:

Modular Exponentiation by Repeated Squaring.

This is an algorithm used to efficiently compute the power of large numbers. The basic idea is to take advantage of the binary representation of the exponent to gradually obtain the result through iterative calculation. The modular exponentiation algorithm reduces the number of modular multiplications in the ordinary algorithm, making the time complexity reduce to the level of log(n) [12].

This algorithm is frequently used in cryptography and cryptanalysis because it can perform modular exponentiation on large numbers without requiring a large amount of memory.

Chinese Remainder Theorem (CRT).

This is a theorem in number theory concerning a system of simultaneous linear congruences. It provides criteria for the existence of a solution to the system of linear congruences and methods for finding the solution. The fundamental idea in applying the Chinese Remainder Theorem is to transform the equation $m^d = c \bmod (n)$ to a system of linear congruences

$$m^d = c_1 \, mod \ (p) \tag{6}$$

$$m^d = c_2 \, mod \ (q) \tag{7}$$

Then by using the CRT to solve for $m^d = c \bmod (n)$, the bit length of the modulus is reduced, thereby increasing the efficiency of modular exponentiation calculations [13].

Montgomery Algorithm.

This is the Montgomery multiplication algorithm proposed by P.L. Montgomery. When computing modulo N, it utilizes modular division. However, division operations require numerous multiplications, resulting in high computational complexity. The idea behind the Montgomery algorithm is to simplify the division operation using base representation, transforming it into bitwise operations.

As bitwise operations are simpler and faster to implement in hardware than division operations, using the Montgomery algorithm can enhance the efficiency of RSA computations. Typically, the implementation of the Montgomery algorithm is combined with the methods (1) and (2) mentioned above, improving efficiency in RSA modular exponentiation calculations in both software and hardware.

5. Conclusion

The article commences by laying the foundational knowledge of public key cryptographic systems, with a focus on RSA encryption and digital signature technology. It then utilizes RSA encryption as a paradigm to elucidate the application of this encryption in digital signature schemes, offering an algorithmic perspective. A critical link is established between the security of digital signatures and the robustness of RSA public key cryptographic systems. Subsequently, the article presents a concise overview of the potential threats facing RSA encryption and proposes pertinent strategies to mitigate these security challenges. It culminates in a synthesis of three distinct approaches, encompassing both algorithmic and hardware aspects, to augment the efficiency of modular exponentiation – a key operation in RSA encryption. Since the inception of the RSA public key cryptosystem, it has been subject to continuous refinement, standardization, and practical deployment. Enhancing the computational efficiency and security of RSA encryption remains a pivotal area of research and discussion within the cryptographic community. Furthermore, with the progressive advancements in quantum theory, traditional public key systems, including RSA and elliptic curve cryptography, confront emerging security vulnerabilities. Thus, the pursuit of post-quantum cryptography, capable of withstanding quantum attacks, represents a frontier scientific endeavor in the field of cryptography.

References

[1] Zheng, D., Zhao, Q., & Zhang, Y. (2013). Cryptography Review. Journal of Xi'an University of Posts and Telecommunications, 18(6), 1-10.

[2] Liu, C., & Fan, J. (2009). Research on the application of RSA asymmetric encryption algorithm in digital signatures. Communication Technology, (3), 192-193.

[3] He, J., Jiang, L., Liao, Q., & Wang, X. (2018). Research on the security of the JSON-based RSA-PKCS# 1 encryption algorithm. Information Technology and Cybersecurity, 37(1), 25-29.

[4] Chen, Q. (1999). Comparison of digital signature algorithms and their applications. Application Research in Computers, 16(10), 10-11.

[5] Zhu, Y., & Cai, G. (2009). Security analysis and improvement of an RSA group signature scheme. Journal of Hubei University of Technology, (1), 68-70.

[6] Shoup, V. (2000). Practical Threshold Signatures. In B. Preneel (Ed.), Advances in Cryptology — EUROCRYPT 2000 (Lecture Notes in Computer Science, vol 1807). Springer, Berlin, Heidelberg.

[7] Liang, X., Sheng, H., & Zhou, X. (2006). Research on RSA Algorithm Mechanism. Computer Security, (12), 26-28.

[8] Sun, Y. (2017). Research on lattice-based RSA small decryption index attack [D]. Beijing Jiaotong University.

[9] Boneh, D., & Durfee, G. (1999). Cryptanalysis of RSA with Private Key d Less than N 0.292. In J. Stern (Ed.), Advances in Cryptology — EUROCRYPT '99 (Lecture Notes in Computer Science, vol 1592). Springer, Berlin, Heidelberg.

[10] Antonov, P., & Georgieva, N. (2024). Cipher Attack Against the Assymetric Cipher RSA. In T. Tagarev & N. Stoianov (Eds.), Digital Transformation, Cyber Security and Resilience (DIGILIENCE 2020, Communications in Computer and Information Science, vol 1790). Springer.

[11] Wang, Y. (2020). Research on quantum attack method of RSA cryptography [D]. Wuhan University.

[12] El Makkaoui, K., Lamriji, Y., Ouahbi, I., Nabil, O., Bouzahra, A., & Beni-Hssane, A. (2022). Fast Modular Exponentiation Methods for Public-Key Cryptography. In 2022 5th International Conference on Advanced Communication Technologies and Networking (CommNet) (pp. 1-6). Marrakech, Morocco.

[13] Guo, S., & Han, W. (2009). An optimization approach to the RSA cryptographic algorithm. Microcomputer Information, 25(3), 63-64..