Postprint: Research on an Improved RSA Algorithm Based on Multi-Prime and Parameter Substitution

Authors: Zhou Jinzhi, Gao Lei

Date: 2018-05-20T00:00:00+00:00

Abstract

To enhance the security of the RSA public-key algorithm in message encryption, this paper conducts improvement research based on an in-depth analysis of the traditional RSA algorithm, proposing a more effective method to optimize its security. Building upon the modification of traditional RSA to four-prime RSA, mathematical transformations are applied for parameter substitution, eliminating the need to transmit the product n of two random prime numbers in the public key and introducing a new parameter x to replace the original parameter n. To address the computational efficiency shortcomings of the improved algorithm, the Chinese Remainder Theorem (CRT) is employed to optimize large-number modular exponentiation operations. Experimental results confirm the feasibility of the improved algorithm, providing a more secure approach for message transmission and reception through public-key encryption; simultaneously, a comparative analysis of decryption (signature) time is conducted between the improved algorithm and both the traditional RSA and four-prime RSA algorithms. The experimental results demonstrate that the improved algorithm also achieves a certain degree of optimization in signature efficiency between message senders and receivers.

Full Text

Preamble

Title: Research on Improved RSA Algorithm Based on Multi-Prime Number and Parameter Substitution

Authors: Zhou Jinzhia†, Gao Leib
Affiliation: Southwest University of Science & Technology, Mianyang Sichuan 621010, China

a. School of Information Engineering; b. Robot Technology for Special Environment Key Laboratory of Sichuan Province

Abstract: In order to improve the security of RSA public-key algorithm in message encryption, this paper conducts improvement research based on in-depth analysis of traditional RSA algorithm and proposes a more effective method to optimize its security. On the basis of improving traditional RSA to four-prime RSA, mathematical transformation is applied for parameter substitution, eliminating the need to transmit the product n of two random primes in the public key. A new parameter x is introduced to replace the original parameter n. To address the efficiency shortcomings of the improved algorithm, the Chinese Remainder Theorem (CRT) is employed to optimize large-number modular exponentiation. Experimental results confirm the feasibility of the improved algorithm, providing a more secure path for message transmission and reception through public-key encryption. Meanwhile, comparative analysis of decryption (signature) time is performed among the improved algorithm, traditional RSA, and four-prime RSA algorithm. Results demonstrate that the improved algorithm also optimizes signature efficiency between message sender and receiver to a certain degree.

Keywords: RSA algorithm; data encryption; parameter substitution; Chinese remainder theorem; public key; signature efficiency

0 Introduction

With the continuous development of information communication technology in the Internet era, information security issues have attracted widespread attention, and information security technology based on cryptography has also developed rapidly. Data encryption technology has evolved into two different cryptographic systems according to key types: symmetric-key cryptosystem and asymmetric-key cryptosystem. Among them, asymmetric-key cryptosystem involves using two different but related keys, namely public key and private key. Plaintext is converted into ciphertext through the public key. This process is called encryption and is performed by the sender. On the other hand, decoding the ciphertext using the receiver' s private key is called decryption and is performed by the receiver. To maintain the confidentiality of the private key, the public key is made available to the public. The public key is used for authentication to ensure messages come from the intended sender. Only the receiver' s private key can decrypt ciphertext from the sender. Since knowledge of the public key is insufficient to decrypt ciphertext, message communication can be conducted securely.

Traditional RSA algorithm is vulnerable to chosen ciphertext attacks, fault attacks, continued fraction attacks, and factorization attacks on modulus n. Regarding improvements to RSA algorithm to enhance its security, references [3,4] propose new algorithm concepts and improved forms for accelerating RSA algo-

rithm implementation during network data exchange, increasing factorization complexity of parameter n by using an additional third prime in public and private key composition, but this is vulnerable to chosen ciphertext attacks. Reference [5] proposes an improved RSA cryptosystem that eliminates parameter n, but only provides a special case illustration lacking experimental scientific demonstration, making the conclusion unconvincing. Reference [6] proposes an improved RSA algorithm based on parameter n, optimizing existing RSA algorithm using four primes, but this method can still attack messages through factorization methods.

---

1 RSA Algorithm and Related Problem Analysis

1.1 Algorithm Principle

The most famous asymmetric encryption algorithm is the RSA algorithm published by Rivest et al. from MIT in 1978, which is currently the most widely used public-key encryption algorithm and recommended by the International Organization for Standardization (ISO) as a public-key data encryption standard. It is an asymmetric-key cryptosystem involving assignment of public and private keys to sender and receiver for message encryption and decryption. Different keys are used during encryption/decryption, namely encryption key and decryption key. Its security is based on number theory and computational complexity theory: calculating the product of two large primes is very easy, but quickly factorizing the product of two large primes to find its factors is computationally difficult. No method has yet been found to effectively crack it. After message encryption using RSA public-key algorithm, message security mainly depends on the complexity of large integer factorization.

RSA public-key encryption algorithm includes three steps: key generation, message encryption, and message decryption. The algorithm description is given below.

1.1.1 RSA Key Generation Randomly select two unrelated large primes p and q and store them confidentially. Calculate n = pq and Euler' s totient function $\phi(n) = (p-1)(q-1)$. Randomly select a positive integer e such that $1 < e < \phi(n)$ and require that $\gcd(e, \phi(n)) = 1$. Then the public encryption key, i.e., public key, is (e, n). Calculate decryption key d such that $0 < d < \phi(n)$ and $ed \equiv 1 \pmod{\phi(n)}$. The private key is (d, n).

1.1.2 Message Encryption The message sender encrypts plaintext M using: $C \equiv M^e \pmod{n}$, where M represents the plaintext to be encrypted, C represents the ciphertext after encryption, e represents the encryption key, d represents the decryption key, and n represents the modulus.

The message receiver decrypts ciphertext C using: $M \equiv C^d \pmod{n}$.

1.2 Optimizability Analysis

RSA has undergone nearly forty years since its publication, experiencing various attack tests. It remains one of the most typical and mature algorithms and is still one of the best algorithms in public-key cryptography. Although RSA algorithm security is based on large integer factorization, there is currently no theoretical proof that cracking RSA is equivalent in difficulty to large integer factorization. Theoretically mastering RSA encryption performance remains unachievable. Moreover, for large-scale data encryption/decryption, RSA algorithm efficiency bottlenecks must be addressed.

RSA algorithm optimizability can be divided into two aspects: algorithm security and operational efficiency, analyzed below.

1.2.1 Security Analysis This paper focuses on algorithm security analysis regarding attacks on parameter n.

(1) Factorization of modulus n
In public keys, cryptanalysts can use forged signatures to factorize modulus n. If successful, they may calculate $\phi(n)$. Thus, the decryption key can be obtained, and the entire RSA public-key algorithm is cracked. For example, attackers can factorize parameter n through the following method:

(a) According to $n = pq$ and $\phi(n) = (p-1)(q-1) = pq - p - q + 1$, we can obtain $p + q = n - \phi(n) + 1$.

(b) According to $(p+q)^2 = p^2 + 2pq + q^2 = (p-q)^2 + 4pq = (p-q)^2 + 4n$, we can find $(p-q) = \sqrt{(p+q)^2 - 4n}$.

(c) According to $p = \frac{(p+q)+(p-q)}{2}$, we can find p.

(d) Then from $q = n/p$, we can find q.

Thus, the attacker completes factorization of parameter n and may crack the RSA cryptosystem.

(2) Chosen ciphertext attack on RSA
Since n is transmitted through public key, attackers can use the public key to encrypt plaintext, then find its influencing factors through trial and error. If any ciphertext matches, attackers can learn secret messages, reducing RSA algorithm security. Chosen ciphertext attack means cryptanalysts pre-select different ciphertexts, have the attacked algorithm decrypt them to obtain corresponding plaintext using unknown keys, then deduce private keys or modulus to obtain desired plaintext. This is the most common attack method against public-key algorithms like RSA. For example, if an attacker wants to crack message x to obtain its signature, they can fabricate two legitimate messages x_1 and x_2 and trick the user into signing them. By calculating $S_1 \equiv x_1^d \pmod{n}$ and $S_2 \equiv x_2^d \pmod{n}$, they can compute the signature of x: $S \equiv x^d \equiv (S_1 \cdot S_2) \pmod{n}$.

1.2.2 Efficiency Analysis The biggest bottleneck in RSA public-key algorithm development is large integer modular exponentiation. Currently, RSA algorithm efficiency is inevitably affected by such huge modulus. Regarding accelerating RSA algorithm implementation during data transmission, many different algorithms have been proposed, such as Sliding Window, modular repeat squaring algorithm, and optimization algorithms based on multiplicative congruence symmetry and exponent 2^k combinations. References [4] and [11] propose methods for offline storage of key parameters and offline key generation in RSA algorithm, optimizing algorithm operational efficiency to some extent but with low improvement. Reference [12] uses four prime factors to calculate two natural numbers in a public-private key pair, increasing cryptosystem security, but lacks experimental proof of feasibility and correctness. Reference [13] comprehensively uses symmetric cipher algorithm AES-128 and asymmetric cipher algorithm RSA. AES-128 maintains data message confidentiality and integrity during message transmission, while RSA handles message authentication and non-repudiation, also demonstrating excellent efficiency performance.

This paper approaches from algorithm operational efficiency perspective, integrating CRT into four-prime RSA algorithm and applying it to digital signatures.

---

2 Improved RSA Algorithm

For the same key bit length, four-prime RSA algorithm requires smaller random primes than traditional RSA algorithm and can reduce key generation computation. However, smaller prime factors make large integer factorization easier. But using more primes reduces RSA algorithm security strength. This paper compares with traditional RSA algorithm by selecting four random large primes, described as follows.

2.1 Key Generation

Randomly generate four different large primes p, q, r, and s, calculate $n = pqrs$ and $\phi(n) = (p-1)(q-1)(r-1)(s-1)$. Randomly select a positive integer e such that $\gcd(e, \phi(n)) = 1$ and $\sqrt{n} < e < \phi(n)$. Calculate decryption key d such that $ed \equiv 1 \pmod{\phi(n)}$.

2.2 Chinese Remainder Theorem for Efficiency Optimization

Chinese Remainder Theorem (CRT) is an ancient Chinese method for solving systems of linear congruences and an important fundamental theorem in elementary number theory. Its content is: let $m_1, m_2, \ldots, m_k$ be k pairwise coprime positive integers, $M = m_1 m_2 \cdots m_k$. For any integers $a_1, a_2, \ldots, a_k$, the system of congruences $x \equiv a_i \pmod{m_i}$ (for $i = 1, 2, \ldots, k$) has a unique solution expression: $x \equiv \sum_{i=1}^{k} a_i b_i M_i \pmod{M}$, where $M_i = M/m_i$ and $b_i \equiv M_i^{-1} \pmod{m_i}$.

Assume plaintext is M and ciphertext is C. The specific process of applying CRT in RSA algorithm is as follows:

Calculate residues of C: $C_p \equiv C \pmod{p}$, $C_q \equiv C \pmod{q}$, $C_r \equiv C \pmod{r}$, $C_s \equiv C \pmod{s}$.

Calculate decryption exponents: $d_p \equiv d \pmod{p-1}$, $d_q \equiv d \pmod{q-1}$, $d_r \equiv d \pmod{r-1}$, $d_s \equiv d \pmod{s-1}$.

Calculate plaintext residues: $M_p \equiv C_p^{d_p} \pmod{p}$, $M_q \equiv C_q^{d_q} \pmod{q}$, $M_r \equiv C_r^{d_r} \pmod{r}$, $M_s \equiv C_s^{d_s} \pmod{s}$.

According to CRT, we can obtain: $M \equiv M_p qrsM_p^{-1} + M_q prsM_q^{-1} + M_r pqsM_r^{-1} + M_s pqrM_s^{-1} \pmod{n}$.

From the above process, the advantage of this theorem is that it can convert large integer modular exponentiation into relatively smaller number modular exponentiation. Applying CRT in RSA algorithm can reduce computation amount and complexity during message decryption.

2.3 Parameter n Replacement for Security Optimization

In traditional RSA algorithm, both keys contain parameter n, which can be factorized into two large primes. If cryptanalysts guess n' s factors, they can easily obtain the private key. To overcome this weakness against parameter n attacks analyzed in Section 1.2.1, this paper attempts to eliminate n distribution in both keys on the basis of four-prime CRT-RSA algorithm by changing the original modulus (product of two primes), i.e., introducing a new parameter x to replace original parameter n, making it difficult for attackers to attack through n. The changed modulus value will be publicly declared, which may be a false value. Even if attackers factorize this new modulus value, they cannot find the original decryption key. Without finding the original decryption key, factorization is worthless.

Combining the optimization methods described in Sections 2.1 and 2.2, the improved RSA algorithm also includes three steps: key generation with replaced parameter n, message encryption, and message decryption.

2.3.1 Key Generation Select four arbitrary large primes p, q, r, and s, calculate $n = pqrs$ and $\phi(n) = (p-1)(q-1)(r-1)(s-1)$. Randomly select a positive integer e such that $1 < e < \phi(n)$ and $\gcd(e, \phi(n)) = 1$.

Calculate x (used to replace n) as follows: if $p > q$, define x such that $(n-q) < x < n$ and $\gcd(x, n) = 1$; if $p < q$, define x such that $(n-p) < x < n$ and $\gcd(x, n) = 1$.

Calculate decryption key d such that $ed \equiv 1 \pmod{\phi(n)}$. The public key is (e, x) and private key is (d, x).

2.3.2 Message Encryption The message sender encrypts plaintext M using: $C \equiv M^e \pmod{x}$.

2.3.3 Message Decryption The message receiver decrypts ciphertext C using: $M \equiv C^d \pmod{x}$.

The improved RSA algorithm flow is shown in [Figure 1: see original paper].

---

3 Experimental Results and Analysis

Experiments were conducted on an i5-2450M CPU with 6 GB memory and Windows 64-bit system using MATLAB R2016a. For key algorithm implementation steps, primality testing adopted the mature Robin-Miller test, CRT algorithm was used to implement large integer modular exponentiation, and Fermat' s Little Theorem was used for modular inverse operations: if a is an integer, p is prime, and $\gcd(a, p) = 1$, then $a^{p-1} \equiv 1 \pmod{p}$, so $a^{-1} \equiv a^{p-2} \pmod{p}$. Modular inverse operations were replaced with polynomial operations to further optimize efficiency. By comparing hash functions MD5, SHA-160, SHA-256, and SHA-512 from reference [14], this experiment adopted SHA-512.

3.1 Key Generation Time Comparison

Algorithm key generation times are shown in . Comparing time consumption for key generation among the three algorithms shows: at 256-bit, the improved algorithm' s key generation speed is 1.68 times that of traditional RSA; at 512-bit, 1.73 times; at 1024-bit, reaching 1.81 times. Although the improved algorithm consumes more time than traditional four-prime RSA algorithm, this is negligible compared to its security contributions, analyzed in Section 3.3. For more intuitive observation of algorithm efficiency, time consumption of both algorithms is compared in [Figure 3: see original paper].

3.2 Decryption Time Comparison

In this experiment, traditional RSA algorithm was improved to four-prime RSA algorithm integrated with CRT, and a new parameter x replaced the originally public n for confidential processing. By generating two groups of random primes for comparison, encrypted ciphertext cannot be recognized, effectively protecting message security and confirming that this improved RSA algorithm achieves good encryption/decryption effects. The improved RSA algorithm effect is shown in [Figure 2: see original paper].

This paper uses Robin-Miller algorithm to generate 256-bit, 512-bit, and 1024-bit primes respectively, then tests traditional RSA, traditional four-prime algorithm, and improved algorithm under varying bit sizes.

Decryption times for the three algorithms are shown in . Comparing time consumption shows: at 256-bit, the improved algorithm' s decryption speed reaches 5.88 times that of traditional RSA; at 512-bit, 9.85 times, close to the theoretical value of 10.86 times; at 1024-bit, reaching 10 times. Compared with four-prime RSA algorithm, the improved algorithm also shows significant speed improvement. Algorithm decryption time comparison is shown in [Figure 4: see original paper].

3.3 Brute-Force Attack Time Comparison

To factorize large integer n, a combination of Elliptic Curve Method (ECM) and General Number Field Sieve (GNFS) is adopted. ECM is typically used for small integer factorization, while GNFS can factor numbers larger than 100 bits. Experiments apply brute-force attacks to the three algorithms, with time consumption shown in . With 16-bit prime generation, traditional algorithm requires 78.676s to brute-force factor parameter n (replaced by x in improved algorithm), four-prime RSA requires 126.498s, while improved algorithm requires 243.95s. The improved algorithm' s brute-force factorization time is 3.1 times that of traditional RSA and 1.93 times that of four-prime RSA. It can be inferred that at higher bit levels, the time gap between traditional RSA and improved algorithm will be larger. This demonstrates that the improved algorithm helps overcome the weakness of parameter n being vulnerable to factorization attacks in RSA.

Brute-force attack time comparison is shown in [Figure 5: see original paper].

---

4 Conclusion

The most direct method to attack RSA public-key algorithm is prime factorization of modulus n. Based on discussion of algorithm optimizability, this paper proposes an improved RSA public-key algorithm that integrates Chinese Remainder Theorem into four-prime algorithm and eliminates parameter n from original RSA algorithm, adding a new parameter x to replace n in private and public keys. Experimental verification demonstrates the effectiveness of the proposed RSA algorithm. Meanwhile, experiments were conducted using standard RSA, four-prime RSA algorithm, and improved RSA algorithm with generated 256-bit, 512-bit, and 1024-bit primes, recording key generation time and decryption time. Results show that improved RSA algorithm significantly reduces signature time and improves algorithm signature efficiency, offering certain advantages over traditional RSA algorithm. Using brute-force attacks on parameter n at 8-bit, 10-bit, and 16-bit levels, the time consumed to crack the improved algorithm is obviously longer than the other two algorithms. Results demonstrate that the improved algorithm enhances security while improving signature speed, proving the superiority and practicality of the improved RSA algorithm based on multi-prime and parameter substitution. Future experiments and ap-

plications will use this improved algorithm to enhance RSA system security coefficients and operational speed, further developing RSA algorithm.

References

[1] Chhabra A, Mathur S. Modified RSA algorithm: a secure approach[C]//Proc of International Conference on Computational Intelligence and Communication Networks. Gwalior, India: IEEE Press, 2011: 545-548.

[2] Xiao Zhenjiu, Hu Chi, Chen Hong. Research and implementation of four-prime RSA digital signature algorithm[J]. Computer Applications, 2013, 33(5): 1376-1377.

[3] Arora S, Pooja. Enhancing cryptographic security using novel approach based on enhanced-RSA and elamal: analysis and comparison[J]. International Journal of Computer Applications, 2015, 112(13): 35-39.

[4] Al-Hamami A H, Aldariseh I A. Enhanced method for RSA cryptosystem algorithm[C]//Advanced Computer Science Applications and Technologies. Kuala Lumpur, Malaysia: IEEE Press, 2013: 402-408.

[5] Sahu J, Singh V, Sahu V, et al. An enhanced version of RSA to increase the security[J]. Journal of Network Communications and Emerging Technologies, 2017, 7(4): 1-4.

[6] Thangavel M, Varalakshmi P, Murrali M, et al. Comment on an enhanced and secured RSA key generation scheme (ESRKGS)[J]. Journal of Information Security & Applications, 2015, 20(C): 3-10.

[7] Rivest R, Shamir A, Aldeman L. A method for obtaining digital signatures and public-key cryptosystems[J]. Communications of the ACM, 1978, 21(2): 120-126.

[8] Feng Dengguo, Zhao Xianfeng. Introduction to Information Security Technology[M]. Beijing: Electronic Industry Press, 2012: 30-32.

[9] Li Jialan. Research on attack methods of RSA algorithm[J]. Science & Technology Vision, 2015(2): 45-47.

[10] He Junjie, Jiao Shuyun, Qi Chuanda. Analysis and improvement of an identity-based signcryption scheme[J]. Computer Application Research, 2013, 30(3): 913-916, 920.

[11] Nagar S A, Alshamma S. High speed implementation of RSA algorithm with modified keys exchange[C]//Proc of the 6th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications. Sousse, Tunisia: IEEE Press, 2012: 639-642.

[12] Mahaveerakannan R, Dhas C S G. Customized RSA public key cryptosystem using digital signature of secure data transfer natural 22 number algo-

rithm[J]. International Journal of Computer Technology & Applications, 2016, 9(5): 2627-2632.

[13] Siregar H, Junaeti E, Hayatno T. Implementation of digital signature using aes and rsa algorithms as a security in disposition digital signature based on MD5 and SHA hash functions using system af letter[J]. Materials Science and Engineering, 2017, 180(1): 12-55.

[14] Mansour A H. Encryption and decryption analysis of the RSA and strong prime[J]. Journal of Soft Computing and Decision Support Systems, 2017, 4(1): 7-15.

Note: Figure translations are in progress. See original paper for figures.