

Extended RSA Cryptosystem Based on the Study of Number Theory and Public Key Cryptosystems

Vijaysinh K. Jadeja¹ Chirag R. Patel² Jignesh N. Solanki³ Sohil A. Gadhiya⁴
^{1,3}Department of Information Technology ^{2,4}Department of Computer Engineering
^{1,2,3,4}C. U. Shah University, Wadhwan, Gujarat, India

Abstract— Security is required to transmit top secret information over the network. Security is also demanding in wide range of applications. Cryptographic algorithms play a vital role in providing the data security against malicious attacks. RSA algorithm is extensively used in the popular implementations of Public Key Infrastructures. In asymmetric key cryptography, also called Public Key cryptography, two different keys (which form a key pair) are used. One key is used for encryption and only the other corresponding key must be used for decryption. No other key can decrypt the message – not even the original (i.e. the first) key used for encryption. In this paper, we have proposed an improved approach of RSA algorithm using two public key pairs and using some mathematical logic rather than sending one public key directly. Because if an attacker has an opportunity of getting the public key component they can find private key value by brute force search.

General Terms: Cryptography, network security

Keywords: ERS, RSA, security, cryptography

I. INTRODUCTION

Number theory may be one of the “purest” branches of mathematics, but it has turned out to be one of the most useful when it comes to computer security. Sensitive data exchanged between a user and a Web site needs to be encrypted to prevent it from being disclosed to or modified by unauthorized parties. The encryption must be done in such a way that decryption is only possible with knowledge of a secret decryption key. The decryption key should only be known by authorized parties.

In traditional cryptography, such as was available prior to the 1970s, the encryption and decryption operations are performed with the same key. This means that the party encrypting the data and the party decrypting it need to share the same decryption key. Establishing a shared key between the parties is an interesting challenge. If two parties already share a secret key, they could easily distribute new keys to each other by encrypting them with prior keys. But if they don't already share a secret key, how do they establish the first one? This challenge is relevant to the protection of sensitive data on the Web and many other applications like it.

II. RSA CRYPTOSYSTEM

The Rivest-Shamir-Adleman (RSA) cryptosystem is one of the best known public key cryptosystems for key exchange or digital signatures or encryption of blocks of data. RSA uses a variable size encryption block and a variable size key. The key-pair is derived from a very large number, n , that is the product of two prime numbers chosen according to special rules; these primes may be 100 or more digits in

length each, yielding an n with roughly twice as many digits as the prime factors.

The public key information includes n and a derivative of one of the factors of n ; an attacker cannot determine the prime factors of n (and, therefore, the private key) from this information alone and that is what makes the RSA algorithm so secure.

A. RSA Key Generation Algorithm:

- 1) Generate two large random primes, p and q , of approximately equal size such that their product $n = pq$ is of the required bit length, e.g. 1024 bits.
- 2) Compute $n = pq$ and $(\phi) \phi = (p-1)(q-1)$.
- 3) Choose an integer e , $1 < e < \phi$, such that $\text{gcd}(e, \phi) = 1$.
- 4) Compute the secret exponent d , $1 < d < \phi$, such that $ed \equiv 1 \pmod{\phi}$.
- 5) The public key is (n, e) and the private key is (n, d) . Keep all the values d, p, q and ϕ secret.
 - n is known as the modulus.
 - e is known as the public exponent or encryption exponent or just the exponent.
 - d is known as the secret exponent or decryption exponent.

B. Encryption Algorithm

Sender A does the following:-

- 1) Obtains the recipient B's public key (n, e) .
- 2) Represents the plaintext message as a positive integer m .
- 3) Computes the cipher text $c = m^e \pmod{n}$.
- 4) Sends the cipher text c to B.

C. Decryption Algorithm

Recipient B does the following:-

- 1) Uses his private key (n, d) to compute $m = c^d \pmod{n}$.
- 2) Extracts the plaintext from the message representative m .

III. NUMBER THEORY BEHIND RSA

A. Prime Generation and Integer Factorization

Two basic facts and one conjecture in number theory prepare the way for today's RSA public-key cryptosystem.

- 1) Step 1. Prime generation is easy: It's easy to find a random prime number of a given size.

This is a result of two other points: Prime numbers of any size are very common, and it's easy to test whether a number is a prime – even a large prime.

According to the Prime Number Theorem, the expected number of candidates to test will be on the order of $\ln x$ (the natural logarithm of x) where x is a typical number of the intended size.

“Large” in the cryptographic context typically means 512 bits (155 decimal digits) or more.

2) Step 2. Multiplication is easy: Given p and q, it's easy to find their product, $n = pq$.

There are many efficient ways to multiply two large numbers.

3) Step 3. Factoring is hard: Given such an n, it appears to be quite hard to recover the prime factors p and q.

Despite hundreds of years of study of the problem, finding the factors of a large number still takes a long time in general. The fastest current methods are much faster than the simple approach of trying all possible factors one at a time. (Such a method would take on the order of n steps.) However, they are still expensive. For instance, it has been estimated recently that recovering the prime factors of a 1024-bit number would take a year on a machine costing US \$10 million. A 2048-bit number would require several billion times more work. [11]

B. Modular Exponentiation and Roots

Given this background, n will hereafter denote the product of two large, randomly generated primes. Let m and c be integers between 0 and n-1, and let e be an odd integer between 3 and n-1 that is relatively prime to p-1 and q-1, meaning the following equation has to be satisfied:

$$\text{gcd}(e, (p-1) \cdot (q-1)) = 1$$

The encryption and decryption operations in the RSA public-key cryptosystem are based on two more facts and one more conjecture:

4) Step 4. Modular exponentiation is easy: Given n, m, and e, it's easy to compute $c = m^e \text{ mod } n$.

The value $m^e \text{ mod } n$ is formally the result of multiplying e copies of m, dividing by n, and keeping the remainder. This may seem to be an expensive computation, involving e-1 multiplications by m with increasingly large intermediate results, followed by a division by n. However, two optimizations make the operation easy:

- 1) Multiplying by an appropriate sequence of previous intermediate values, rather than only by m, can reduce the number of multiplications to no more than twice the size of e in binary.
- 2) Dividing and taking the remainder after each multiplication keeps the intermediate results the same size as n.
- 5) Step 5. Modular root extraction – the reverse of modular exponentiation – is easy given the prime factors: Given n, e, c, and the prime factors p and q, it's easy to recover the value m such that $c = m^e \text{ mod } n$.

The value m can be recovered from c by a modular exponentiation operation with another odd integer d between 3 and n-1. In particular, for this d, the following holds for all m:

$$m = (m^e)^d \text{ mod } n$$

This integer d is easy to compute given e, p, and q.

6) Step 6. Modular root extraction is otherwise hard: Given only n, e, and c, but not the prime factors, it appears to be quite hard to recover the value m.

We will want to compute d from e, p and q, where d is the multiplicative inverse of e. It means $e \cdot d = 1 \text{ (mod } \phi(n))$

IV. EXTENDED ERSA(EXTENDED RSA)ALGORITHM

RSA is a block cipher in which the plaintext and cipher text are integers between 0 and n- 1 for some n.

Encryption and decryption are of the following form, for some plaintext block M and cipher text block C:

$$C = M^{y/x} \text{ mod } n$$

$$M = C^d \text{ mod } n = (M^{y/x})^d \text{ mod } n$$

Both sender and receiver must know the values of n, y and x only the receiver knows the value of d.

A. Process of Encryption and Decryption:

This is a public key encryption algorithm with a public key of $KU = \{y, n\}, \{x\}$ and a private key of $KR = \{d, n\}$.

For this algorithm to be satisfactory for public-key encryption, the following requirements must be met:

- 1) It is possible to find values of y, x, d, n such that $(M^{y/x})^d = M \text{ mod } n$ for all $M < n$.
- 2) It is relatively easy to calculate $M^{y/x}$ and C^d for all values of $M < n$.
- 3) y is a multiple of x and e (which the public key in the normal RSA algorithm)

B. Computational Steps:

- Begin by selecting two prime numbers, p and q, and calculating their product n, which is the modulus for encryption and decryption.
- Next, we need the quantity $\phi(n)$ referred to as the Euler totient of n, which is the number of positive integers less than n and relatively prime to n.
- Then select an integer e that is relatively prime to $\phi(n)$ (i.e., the greatest common divisor of e and $\phi(n)$ is 1).
- Select two numbers x and y such that $y = xe$
- Using this numbers formulate two public key $\{y, n\}, \{x\}$
- Finally, calculate d as the multiplicative inverse of e (which is public key in normal RSA), modulo $\phi(n)$.
- Suppose that user A has published its public key and that user B wishes to send the message M to A.
- Then B calculates $C = M^{y/x} \text{ (mod } n)$ and transmits C.

V. EXPERIMENTAL RESULTS

In order to justify the performance we used different modulus length(256 bits, 512 bits,1024 bits) and the block sizes (128 bits,256 bits,512 bits,1024 bits). The two following table shows the experimental results of RSA and IRSA respectively.

bitlength h(n)	blocks ize in bits	key generatio n(x) MS	Encrypt ion time(y) MS	Decrypt ion time(z) MS	Total time t=x+y +z MS
128	64	42	86	230	358
256	128	98	102	520	720
1024	512	212	112	1250	1574

Table 1: Table of experimental result on RSA

bit length (n)	blocks ize in bits	key generatio n(x) MS	Encrypt ion time(y) MS	Decrypt ion time(z) MS	Total time t=x+y +z MS
128	64	42	86	230	358
256	128	98	102	520	720
1024	512	212	112	1250	1574

128	64	44	87	245	386
256	128	100	112	720	932
1024	512	425	129	1890	2444

Table 2: Table of experimental result on ERSA

RSA	ERSA
Use only one public key	Use two public key
Less communication overload	High communication overload
More vulnerable to brute force attack	Less vulnerable to brute force attack
Less secure	More secure
The Public key is sent once	The Public key is sent separately twice

Table 3: Table of comparison between RSA and ERSA

VI. CONCLUSION

This research is a study of number theory and public key cryptosystems and based on this improving the RSA cryptosystem that is more reliable to brute force attack. RSA cryptosystem produces one public key to encrypt the message. Though it is hard to find out the factors of n and get p and q , two large prime numbers, therefore brute force attack is more difficult in our proposed algorithm as the encryption keys are sent separately, not at once. The proposed RSA is used for system that needs high security but with less speed.

REFERENCES

- [1] B.A. Forouzan, *Cryptography & Network Security*, Tata Mcgraw Hill publishing company limited, 2010.
- [2] A.S. Tanenbaum, "Computer Networks", 4th edition, Prentice Hall of India, 2007-2008.
- [3] S. Sharma, J.S. Yadav, P. Sharma. Modified RSA Public Key Cryptosystem Using Short Range Natural Number Algorithm. *International Journal of Advanced Research in Computer Science and Software Engineering*, volume 2, Issue 8, August 2012
- [4] P.S.Yadav, P. Sharma, Dr. K. P. Yadav, Implementation of RSA algorithm using Elliptic Curve algorithm for security and performance enhancement, *International Journal of Scientific & Technology Research*, Volume 1, Issue 4, May 2012.
- [5] D. Pugila, H. Chitrana, S. Lunawat, P.M. Durai Raj Vincent, An Efficient Encryption Algorithm Based On Public Key Cryptography. *International Journal of Engineering and Technology*, Volume 5 No 3, Jun-2013.
- [6] Burt Kaliski, "The Mathematics of the RSA Public-key Cryptosystem RSA Laboratory.
- [7] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, "Introduction to Algorithm", 2nd edition, MIT press, 2002.
- [8] Shilpa M Pund, Chitra G Desai. Implementation of RSA using Mersenne Prime. *International Journal of Networking & Parallel Computing*. Volume 1, Issue 3, Dec 2012-Jan 2013.