International Journal of Computer Science and Business Informatics

IJCSBI.ORG

A New Design to Improve the Security Aspects of RSA Cryptosystem

Sushma Pradhan and Birendra Kumar Sharma

School of Studies in Mathematics,
Pt. Ravi Shankar Shukla University,
Raipur, Chhattisgarh, India

ABSTRACT

This paper introduce a security improvement on the RSA cryptosystem, it suggests the use of randomized parameters in the encryption process to make RSA many attacks described in literature, this improvement will make the RSA semantically secure i.e. , an attacker cannot distinguish two encryptions from each other even if the attacker knows (or has chosen) the corresponding plaintexts. This paper also briefly discuss some other attacks on the RSA and the suitable choice of RSA parameter to avoid attacks, also an important issue for the RSA implementation is how to speed up the RSA encryption and decryption process.

Keywords

RSA cryptosystem, RSA signature, RSA Problem, Public Key Cryptosystems, Private Key Cryptography.

1. INTRODUCTION

A powerful tool for protection is the use of Cryptography. Cryptography underlies many of the security mechanisms and builds the science of data encryption and decryption. Cryptography [1] enables us to securely store sensitive data or transmit across insecure networks such that it cannot be read by anyone except the intended recipient. By using a powerful tool such as encryption we gain privacy, authenticity, integrity, and limited access to data. In Cryptography we differentiate between private key cryptographic systems (also known as conventional cryptography systems) and public key cryptographic systems.

Private Key Cryptography, also known as secret-key or symmetric-key encryption is based on using one shared secret key for encryption and decryption. The development of fast computers and communication technologies did allow us to define many modern private key cryptographic systems, e.g. in 1960's Feistel cipher [2], Data Encryption Standard (DES), Triple Data Encryption standards (3DES), Advanced Encryption Standard

(AES), The International Data Encryption Algorithm (IDEA), Blowfish, RC5, CAST, etc. The problem with private key cryptography was the key management, a system of n communicating parties would require to manage ((n-1)*n)/2, this means that to allow 1000 users to communicate securely, the system must manage 499500 different shared secret key, thus it is not scalable for a large set of users.

A new concept in cryptography was introduced in 1976 by Diffie and Hellman [2] called as public-key cryptography and is based on using two keys (Public and Private Key). The use of public key cryptography solved many weaknesses and problems in private key cryptography, many public key cryptographic systems were specified (e.g. RSA [3], ElGamal [4], Diffie-Hellman key exchange [2], elliptic curves [5], etc.). The security of such Public key cryptosystems is often based on apparent difficulties of some mathematical number theory problems (also called "one way functions") like the discrete logarithm problem over finite fields, the discrete logarithm problem on elliptic curves, the integer factorization problem or the Diffie-Hellman Problem, etc. [1].

One of the firstly defined and often used public key cryptosystems is the RSA. The RSA cryptosystem is known as the de-facto standard for Public-key encryption and signature worldwide and it has been patented in the U.S. and Canada. Several standards organizations have written standards that use of the RSA cryptosystem for encryption, and digital signatures [6]. The RSA cryptosystem was named after his inventors R. Rivest, A. Shamir, and L. Adleman and is one of the mostly used public-key cryptosystem.

Due to the wide use of the RSA cryptosystem, it is critical to ensure a high level of security for the RSA. In this paper, we introduce a new design to improve the security of the RSA cryptosystem; this is achieved by using randomized parameter that make the encrypted message more difficult for an adversary to break; thus making the RSA more secure.

This paper is organized as follows. In the next section the mathematical basics preliminaries on RSA are briefly described. In Section 3, we describe our new scheme with security improvement that can protect us against the given attacks. In Section 4, we give the comparison between basic RSA cryptosystem and our new scheme. Finally, we give a short conclusion in Section 5.

2. BASIC PRELIMINARIES

The security of the RSA cryptosystem is based on the intractability of the RSA problem. This means that if in the future the RSA problem generally solved then the RSA cryptosystem will no longer be secure.

Definition2.1. The RSA problem (RSAP) is the following: given a positive integer n that is a product of two distinct odd primes p and q, a positive integer e such that gcd(e, (p − 1)(q − 1)) = 1, and an integer c, find an integer m such that $m^e = c(\mathrm{mod}\, n)$.

This means that the RSA problem is based on finding the e-th roots modulo a composite integer n.

Definition2.2. For n>1, let φ (n) denote the number of integers in the interval [1, n] which are relatively prime to n. The function φ is called the Euler phi function (or the Euler totient function).

The RSAP has been studied for many years but still an efficient solution was not found thus it is considered as being difficult if the parameters are carefully chosen, but if the factors of n are known then the RSA problem can easily be solved, an adversary can then compute Euler φ (n) = (p−1) (q−1) function, and the private key d, once d is obtained the adversary can decrypt any encrypted text. It is also widely believed that the RSA and the integer factorization problems are computationally equivalent, although no proof of this is known.

Remark: The problem of computing the RSA decryption exponent d from the public key (n, e) and the problem of factoring n are computationally equivalent [6]. This imply that when generating RSA keys, it is important that the primes p and q be selected in sufficient size such that factoring n = p*q should be computationally infeasible.

The RSA public key and signature scheme is often used in modern communications technologies; it is one of the firstly defined public key cryptosystem that enable secure communicating over public unsecure communication channels.

The following algorithms describe the RSA key generation, and the RSA cryptosystem (basic version)

Algorithm2.1 Key generation for the RSA public-key encryption

Each user A creates an RSA public key and the corresponding private key.

User A should do the following:

1. Generate two large random (and distinct) prime's p and q, each roughly the same size.
2. Compute n = p *q and φ (n) = (p − 1) (q − 1).
3. Select a random integer e, 1 < e < φ (n), such that gcd (e, φ (n)) =1.
4. Use the Euclidean algorithm to compute the unique integer d, 1 < d < φ (n), such that e *d ≡ 1 (mod φ (n)).
5. User A public key is (n, e) and A's private key is d.

Definition2.3. The integer's e and d in RSA key generation are called the encryption exponent and the decryption exponent, respectively, while n is called the modulus.

Algorithm2.2. The RSA public-key encryption and decryption (Basic version)

User B encrypts a message m for user A, which A decrypts.

1. Encryption. User B should do the following:

1. Obtain user A authentic public key (n, e).
2. (Represent the message as an integer m in the interval [0, n − 1].
3. Compute $c = m^e \pmod{n}$.
4. Send the encrypted text message c to user A.

2. Decryption. To recover plaintext m from c, user A should do the following:

1. Use the private key d to recover $m = (m^e)^d \pmod{n}$.

The original RSA encryption, decryption does not contain any randomized parameter making the RSA cryptosystem deterministic, which means that an attacker can distinguish between two encryptions, based on this many of the attacks listed below can be performed on the RSA basic version.

3. NEW SCHEME

The key generation remains unchanged as in the original RSA, see above. The following algorithms describe as follows.

Algorithm3.1. The RSA public-key encryption and decryption (Modified version)

User B encrypts a message m for user A, which A decrypts.

1. Encryption. User B should do the following:

1. Obtain user A authentic public keys (n, e).
2. Represent the message as an integer m in the interval [0, n − 1].
3. Select a random integer k, 1 < k < n, such that gcd (k, n)) = 1.
4. Compute $c_1 = k^e \pmod{n}$
5. Compute $c_2 = m^e k \pmod{n}$
6. Send the encrypted text message (c_1, c_2) to user A.

2. Decryption. To recover plaintext m from c_2, user A should do the following:

1. Use own private key d and compute: $c_1 = k \pmod{n}$
2. Use the Euclidean algorithm and calculate the unique integer s, 1 < s < n, such that s *k ≡1 (mod n).
3. Compute $c_2 s = (m^e k)s = (m^e)ks = m^e \pmod{n}$.
4. Recover m, use the private key d and compute: $(m^e)^d = m \bmod n$

The following example illustrates the use of modified RSA cryptosystem.

Example: (RSA Encryption/Decryption)

Key Generation: Assume p = 2350, q = 2551, n = p* q = 6012707

1. Encryption. User B should do the following:

1. User A authentic public key e = 3674911
2. Message m = 31
3. Random k = 525
4. Compute: $525^{3674911}$= 20639mod6012707
5. Compute: $31^{367491125}$ = 2314247mod6012707
6. Send (20639, 2314247) to user A

2. Decryption. To recover plaintext m from c, user A should do the following:

1. User A private key d= 422191, compute: 20639^{422191} =

525mod6012707

2. Extended GCD (525, 6012707) → s = 3516002
3. Compute: 2314247*3516002 = 2913413mod6012707
4. Recover m: 2913413^{422191} = 31mod6012707.

The RSA encryption/decryption is much slower than commonly used symmetric key encryption algorithms such as the well know algorithm DES and this is the reason why in practice RSA encryption is commonly used to encrypt symmetrical keys or to encrypt small amount of data, there are many software solutions or hardware implementations to speeding up the RSA encryption/decryption process. For more information about speeding up RSA software implementations see [6].

Because the basic version of the RSA cryptosystem has no randomization component an attacker can successfully launch many kinds of attack, now we discuss some of these attacks.

1. Known plain-text attack: A known-plaintext attack is one where the adversary has a quantity of plaintext and corresponding cipher-text [6]. Given such a sorted set $S = \{\{p_1, c_1\}, \{p_2, c_2\} \ldots . \{p_r, c_r\}\}$ (where $p_i \in P$ plaintext set, $c_i \in C$ cipher text set, r < φ (n) is the order of Z*n) an adversary can determine the plaintext p_x if the corresponding c_x is in S. The modified version of the RSA described above use *k* as randomizing parameter; this can protect the encrypted text against known plain text attacks.

2. Small Public/Private exponent e/d Attack: To reduce decryption time, one may wish to use a small value of private exponent d or reduce the encryption time using a small public exponent e, but this can result in a total break of the RSA cryptosystem as Coppersmith [10] and M.Wiener [11] showed.

3. Johan Hstad and Don Coppersmith Attack: If the same clear text message is sent to more recipients in an encrypted way, and the receivers share the same exponent e, but different p, q, and n, then it is easy to decrypt the original clear text message via the Chinese remainder theorem [6]. Johan Hstad [7] described this attack and Don Coppersmith [8] improved it.

4. Common Modulus Attack: If also same message m is encrypted twice using the same modulus n, then one can recover the message m as follows: Let $c_1 = m^{e1} (\mathrm{mod}\, n)$, and $c_2 = m^{e2} (\mathrm{mod}\, n)$ be the cipher texts corresponding to message m, where gcd $(e_1, e_2) = 1$, then attacker recovers original message $m_1 = c_1^{\ a} * c_2^{\ b} \bmod n$ for $e_1 * a + e_2 * b = 1$. Using the extended great common divisor (GCD) one can determine a and b then calculate m without knowing private key d, this is known in the literature as the Common Modulus Attack that requires $O((\log K)^2)$, where k is maximum size of a or b.

5. Timing Attack: One attack on the RSA implementation is the Timing Attack; Kocher [9] demonstrated that an attack can determine a private key by keeping track on how long a computer takes to decrypt a message.

6. Adaptive chosen cipher text attacks (ACC attacks): In 1998, Daniel Bleichenbacher [12] described the first practical adaptive chosen cipher text attack, against RSA-encrypted messages using the PKCS#1v1[13] padding scheme (a padding scheme randomizes and adds structure to an RSA-encrypted message, so it is possible to determine whether a decrypted message is valid.) Bleichenbacher was able to mount a practical attack against RSA implementations of the Secure Socket Layer protocol (SSL) [14], and to recover session keys, here it is important to mention that such protocol is still often used in internet to secure emails and e-payment via internet. As a result of this work, cryptographers now recommend the use of provably secure padding schemes such as Optimal Asymmetric Encryption Padding, and RSA Laboratories has released new versions of PKCS#1 that are not vulnerable to these attacks.

7. Attacks on the factorization problem: Some powerful attacks on the RSA cryptosystem are the attacks on the factorization problem; the factoring algorithms to solve the factorization problem come in two parts: special purpose and general purpose algorithms. The efficiency of special purpose depends on the unknown factors, whereas the efficiency of the latter depends on the number to be factored. Special purpose algorithms are best for factoring numbers with small factors, but the numbers used for the modulus in the RSA do not have any small factors. Therefore, general purpose factoring algorithms are the more important ones in the context of cryptographic systems and their security.

A major requirement to avoid factorization attacks on the RSA cryptosystem is that p and q should be about the same bits length and

sufficiently large. For a moderate security level p and q should be at least 1024 bits length, this will result in a 2048 bit length for modulus n. furthermore p and q should be random prime number and not of some special case binary bit structure.

The following table summarizes the running time for some of the well known integer factoring algorithms where p denotes the smallest prime factor of n, and e = 2.718 is the Euler's constant.

Table 1. Factorization algorithms

Algorithm	**Runtime Estimation**
1.Pollards Rho [15]	O(p)
2.Pollards p − 1 [16]	O (p*) where p* is the largest prime factor of p − 1.
3.Williams p + 1 [17]	O (p*) where p* is the largest prime factor of p + 1.
4.Elliptic Curve Method (ECM) [18]	$O(e^{(1+o(1))(2\ln p \ln\ln p)1/2)})$
5.Quadratic Sieve (Q.S.) [19]	$O(e^{(1+o(1))(2\ln N \ln\ln N)1/2)})$
6.Number Filed Sieve (NFS) [20]	$O(e^{(1.92+o(1))(\ln N)1\backslash 3(\ln\ln N)2/3)})$

In 2010, the largest number factored by a general-purpose factoring algorithm was 768 bits long [21] using distributed implementation thus some experts believe that 1024-bit keys may become breakable in the near future so it is currently recommended to use 2048 for midterm security and a 4096-bit keys for long term security.

Now, the described RSA security improvement in this paper can protect us against the following attacks:

Table 2. Improved RSA is immune against the following attacks

Attack	**Justification**
1.Known plain-text attack	Is not possible as described above

2. Small public exponent e	Is not possible due to the use of random integer k
3. Hasted and Coppersmith attack	Is not possible because every msg. have unique ki
4. Common Modulus Attack	Is not possible because every msg. have unique ki
5. Timing Attack	Using k in encryption and decryption process will make it difficult to distinguish between the time for k and the time for public e or private key d
6.ACC attacks	One can use randomized integer k instead of secure padding

This will make the RSA cryptosystem more secure compared with the basic version of the RSA cryptosystem. The modification makes the RSA cryptosystem semantically secure that means an attacker cannot distinguish two encryptions from each other even if the attacker knows (or has chosen) the corresponding plaintexts.

4. COMPARISION WITH STANDARD RSA CRYPTOSYSTEM

We can compare the new scheme to the RSA cryptosystem. For the latter, the natural security parameter is n = the logarithm of the RSA modulus. The public and secret keys of RSA have size O (n), and both encryption and decryption require time $O(n^3)$ (using ordinary multiplication algorithms). For our new scheme, the natural security parameter is the dimension k. The keys for the new system are relatively large: size $O(k^3)$ for the public key and $O(k^2)$ for the secret key. However, the time required for encryption is only O (n) and no multiplications are needed. Decryption requires time $O(k^3)$, comparable to RSA (again using ordinary multiplication algorithms).

5. CONCLUSION

In this paper, we briefly discussed the improve security of the RSA public-key cryptosystem. This improvement use randomized parameter to change every encrypted message block such that even if the same message is sent more than once the encrypted message block will look different. The major advantage gained in the security improvement described above is making RSA system immune against many well known attacks on basic RSA cryptosystem; thus making the RSA encryption more secure. This is

essential because RSA is implemented in many security standards and protocols and a weak RSA may result in a whole compromised system. Although the security improvement make RSA more secure nevertheless it should be noted that the RSA modulus n bit length should be at least 2048 to ensure a moderate security and to avoid powerful attacks on the discrete logarithm and factorization problem. This security consideration and other mentioned in literature should be used to define an improved version of the RSA.

REFERENCES

[1] D. Kahn., *The Code breakers: The comprehensive History of Secret Communication from Ancient to the Internet*, Scribner, 1967.

[2] W. Diffie and M.Hellman, New directions in cryptography, *IEEE Transactions on Information Theory*, vol. 22 (1976), 644-654.

[3] R. L. Rivest, A. Shamir, and L. Adelman, A method for obtaining digital signatures and public key cryptosystems*" Commun. Of the ACM*, Vol.21 (1978), 120-126.

[4] T. ElGamal, A public-key cryptosystem and a signature scheme based on discrete logarithms, *IEEE Transactions on Information Theory*, Vol. 31 (1985), 469-472.

[5] N. Koblitz, Elliptic curve cryptosystems, *Mathematics of Computation*, Vol.48 (1987), 203-209.

[6] A. Menezes, P. van Oorscot and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, ISBN: 0-8493-8523-7, 1999

[7] Hstad and M. Nslund, The Security of all RSA and Discrete Log Bits, *Journal of the ACM*, Vol.51, No.2 (2004), 187-230.

[8] Don Coppersmith, Small Solutions to Polynomial Equations, and Low Exponent RSA Vulnerabilities, *Journal of Cryptology*, Vol. 10, No. 4, (Dec. 1997).

[9] P. Kocher, Timing attacks on implementations of Diffie-Hellman, RSA, DSS and other systems. *Advances in Cryptology*, Vol. 1109 (1996), 104-113.

[10] Don Coppersmith, Matthew K. Franklin, Jacques Patarin, Michael K. Reiter, Low Exponent RSA with Related Messages, *EUROCRYPT (1996*), 1-9.

[11] M. Wiener., Cryptanalysis of short RSA secret exponents, *IEEE Transactions on Information Theory*, Vol. 36(1990), 553- 558.

[12] Daniel Bleichenbacher., Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS #1, *Advances in Cryptology CRYPTO '98 Lecture Notes in Computer Science*, Vol. 1462 (1998).

[13] PKCS#1: RSA Cryptography Standard, website: http://www.rsa.com/rsalabs/node.asp?id=2125

[14] The Transport Layer Security (TLS) Protocol, version 1.2, website: http://tools.ietf.org/html/rfc5246.

[15] J. M. Pollard, A Monte Carlo method for factorization, *BIT Numerical Mathematics*, Vol. 15, No. 3 (1975), 331-334.

[16] J. M. Pollard, Theorems of Factorization and Primality Testing", *Proceedings of the Cambridge Philosophical Society*, Vol.76, No. 3 (1974), 521-528.

[17] H. C. Williams., A p+1 method of factoring, *Mathematics of Computation*, Vol. 39, No.159 (1982), 225-234.

[18] B. Dixon, A.K. Lenstra, Massively parallel elliptic curve factoring, *Advances in Cryptology-EUROCRYPT' 92 Lecture Notes in Computer Science*, Vol. 658, (1993), 183-193.

[19] C. Pomerance, The quadratic sieve factoring algorithm, *Advances in Cryptology Lecture Notes in Computer Science*, Vol.209 (1985), 169-182.

[20] J. Buchmann, J. Loho, J. Zayer, An implementation of the general number field sieve, *Advances in Cryptology-CRYPTO' 93 Lecture Notes in Computer Science*, Vol.773 (1994), 159-165.

[21] RSA Laboratories, the RSA Factoring Challenge http://www.rsa.com/rsalabs/node.asp-id=2092