AN EFFICIENT VARIANT OF RSA CRYPTOSYSTEM WITH SEMANTIC SECURITY

SahadeoPadhye
School of Studies in Mathematics, Pt.Ravishankar Shukla University,
Raipur (C.G.),India.
Email: *sahadeo_mathrsu@yahoo.com*

Abstract. An efficient variant of RSA cryptosystem was proposed by Cesar [2]. He called it Rprime RSA. The Rprime RSA is a combination of Mprime RSA [3] and Rebalanced RSA [9, 1]. Although the decryption speed of Rprime RSA is 27 times faster than the standard RSA and 8 times faster than the QC RSA [6] in theoretically, yet due to the large encryption exponent, the encryption process becomes slower than the standard RSA. In this paper we tried to improve the efficiency of encryption process with less compromising with the decryption speed. In addition the proposed scheme is semantically secure.
Keywords: Public key cryptosystem, RSA, CRT, semantically secure.
2000 Mathematic Subject Classification No. 94A60.

1. Introduction

The first practical and most popular widely used public key cryptosystem (PKC) was proposed by Rivest, Shamir and Adleman, known as RSA PKC [7]. The security of RSA cryptosystem is based on the factorization of a large composite number. Although RSA is most popular and very easy to understand, yet the speed is slower than symmetric key cryptosystem (around 100 times slower than DES). To improve the efficiency of standard RSA scheme, there are so many variants have been proposed. In 1982, Quiquaster and Conver [6] introduced a fast deciphering algorithm to speed up the decryption process via Chinese Remainder Theorem (CRT), which we called QCRSA. The theoretical speed up of QCRSA is approximately 4 times faster than the standard RSA [1]. Some other variants are Batch RSA [4], Mprime RSA [3], Mpower RSA [8], Rebalanced RSA [1] and Rprime RSA [2]. All are improving the efficiency of decryption speed. Boneh and Shacham [1] gave a nice survey on all four variants (Batch RSA, Mprime RSA, Mpower RSA, Rebalanced RSA). The result given by Boneh and Shacham [1] was then extended by Ceaser [2]and proposed a fast variant of RSA scheme. He called it Rprime RSA, which is a combination of Mprime RSA [3] and Rebalanced RSA [1]. The theoretical speed up of Rprime RSA is about 4.8 times faster than the QC-RSA for the moduli of 1024 bits and it is 27

This work is supported under CSIR (JRF) scheme, India (2002).

times faster than standard RSA for 2048 bit moduli. Although, Rprime-RSA speed up the efficiency of decryption process, but due to the large encryption exponent e the encryption speed becomes very slow. In this article we extend the result given by Cesar [2] to improve the efficiency of encryption process with less compromising the speed of decryption process. In addition, the Rprime RSA is not semantically secure, where as the proposed cryptosystem is semantically secure.We also compare our proposed scheme with the DRSA cryptosystem given by Pointcheviel [10], a variant of RSA cryptosystem with the property of semantic security.

The rest of the section is as follows. In the section 2 we review the standard RSA and the QC-RSA, in the section 3 we describe Rprime RSA , in the section 4 we describe about the DRSA scheme and we introduce our proposed scheme in the section 5. At last, in the section 6 we analyze our scheme with comparison to the Rprime RSA and DRSA.

2. RSA Cryptosystem

In this section we give a brief review of three basic algorithm used to complete RSA scheme: Key generation, Encryption and Decryption. For the details about RSA, we refer the reader to the original paper of RSA [7].

2.1. **Key Generation.** To generate keys for the RSA scheme receiver R chooses two large primes p and q and computes $n = pq$. He then chooses an integer e less than and relatively prime to $\phi(n)$ and computes an integer d such that $ed \equiv 1 mod\ \phi(n)$. The public key and the secret key for the receiver R is (e, n) and d respectively. Plaintext and the ciphertext space is $0, 1, 2, ..., n-1$.

2.2. **Encryption.** To encrypt any plaintext M, the sender S computes $C = M^e (mod\ n)$ by using the public key of R and sends the ciphertext C to the receiver R.

2.3. **Decryption.** After getting the ciphertext C the receiver R computes $C^d (mod\ n) = M$ by using his secret key d.

In 1982, Quisquater and Couvreur [6] gave a fast deciphering algorithm for the RSA by using Chinese Remainder Theorem, which we called QC-RSA. In this algorithm, receiver first computes $M_p = C^{d_p} (mod\ p)$ and $M_q = C^{d_q} (mod\ q)$ where $d_p \equiv d (mod\ p-1)$ and $d_q \equiv d (mod\ q-1)$, in place of $M = C^d (mod\ n)$ and then he determines the plaintext M by M_p and M_q via Chinese Remainder Theorem. This algorithm is approximately 4 times as fast as evaluating $C^d (mod\ n)$.

3. Rprime RSA

In a paper Ceaser [2] introduced an efficient variant of RSA by combining Mprime-RSA [3] and Rebalanced RSA [1]. He called that Rprime-RSA. In that scheme Rebalanced RSA was used for key generation together with the

decryption algorithm of Mprime RSA. Some other possibilities of combinations are also analyzed by him and concluded that Rprime is the batter than all other combinations. The Key generation, Encryption and Decryption process of Rprime RSA is as follows.

3.1. **Key Generation.** To generate keys for the Rprime RSA scheme, the receiver R proceeds as follows:

(1) Chooses an integer $s \leq \frac{\lceil log\ n \rceil}{k}$.
(2) Generates k distinct random primes of $\lfloor \frac{log\ n}{k} \rfloor$ bits $p_1, p_2, ..., p_k$ with $gcd\ (p_1 - 1, p_2 - 1, ...p_k - 1) = 2$, and calculate the modulus $n = p_1 p_2 ... p_k$.
(3) Generates k random numbers of s-bits $d_{p_1}, d_{p_2}, ...d_{p_k}$ such that $gcd\ (d_{p_i}, p_i - 1) = 1\ \forall i = i, 2, 3, ...k$ and $d_{p_1} \equiv d_{p_2} \equiv ... \equiv d_{p_k} mod\ 2$.
(4) Find d such that $d \equiv d_{p_i} mod\ (p_i - 1)$.
(5) Calculate e such that $e \equiv d^{-1} mod\ \phi(n)$.

The public keys for the receiver R are (e, n) and the private keys are $(p_1, p_2, p_3, ...p_k, d_{p_1}, d_{p_2}, ..., d_{p_k})$

3.2. **Encryption.** The encryption process for the Rprime RSA is same as that for the standard RSA. To encrypt any plaintext M sender S computes $C = M^e mod\ n$, sends the ciphertext C to the receiver R.

3.3. **Decryption.** To decrypt the ciphertext C, the receiver R first calculate $M_i = C^{d_i} mod\ p_i$ for each $i = 1, 2, 3, ..., k$. R computes the plaintext $M (= C^d (mod\ n))$ with the help of above congruence equations via CRT. The use of CRT takes negligible time with comparison to k exponent because of the choice of primes.

The theoretical speed up of Rprime RSA with comparison to QC-RSA is given by $\frac{(log\ n)k}{4s}$.

4. DRSA Cryptosystem

The standard RSA and other variants are not semantically secure [5]. Pointcheviel [10] gave a new Dependent RDA (DRSA) problem and proposed a cryptosystem with the property of semantic security. The protocol for DRSA system is as belows:

4.1. **Key Generation.** The key generation for DRSA scheme is same as that for the standard RSA scheme. To generate keys in DRSA scheme, the receiver R chooses two large primes p and q and computes $n = pq$. R then determines an integer e less than and relatively prime to $\phi(n)$ and computes an integer d such that $ed \equiv 1 mod \phi(n)$. The public key and the secret key for the receiver R is (e, n) and d respectively. The prime p and q are also kept secret.

4.2. **Encryption.** To encrypt any plaintext $M \in Z_n$, sender S first randomly selects an integer $k \in Z_n^\star$ and sends the complete ciphertext (C_1, C_2) to the receiver R. Where, $C_1 = k^e mod\ n$ $C_2 = M(k+1)^e mod\ n$.

4.3. **Decryption.** To decrypt the ciphertext (C_1, C_2), R first computes $C_1^d mod\ n = k$ and then he obtains the original plaintext by computing $M = \frac{C_2}{(k+1)^e} mod\ n$.

5. Proposed Cryptosystem

In this section, we propose a cryptosystem to improve the encryption speed of Rprime RSA with a very less compromising the decryption process. The Key generation, Encryption and the Decryption process are as follows.

5.1. **Key Generation.** To generate keys the receiver R takes an integer $s < \frac{\lceil log\ n \rceil}{k}$ and executes the following stapes:

(1) Generates k distinct random primes of $\lceil \frac{log\ n}{k} \rceil$ bits $p_1, p_2, p_3, ..., p_k$ with $gcd(p_1 - 1, p_2 - 1, ..., p_k - 1) = 2$ and calculate $p_1 p_2 p_3 ... p_k$.
(2) Generates k distinct random numbers of s-bits such that $d_{p_1}, d_{p_2}, ...d_{p_k}$ such that $gcd\ (d_{p_i}, p_i - 1) = 1\ \forall i = i, 2, 3, ...k$ and $d_{p_1} \equiv d_{p_2} \equiv ... \equiv d_{p_k} mod\ 2$. .
(3) Finds d such that $d \equiv d_{p_i} mod\ (p_i - 1)$.
(4) Calculates e such that $e \equiv d^{-1} mod\ \phi(n)$.

The receiver R makes publically available to the keys (e, n) and keeps secret to the keys $(p_1, p_2, p_3, ...p_k, d_{p_1}, d_{p_2}, ..., d_{p_k})$

5.2. **Encryption.** To encrypt any message $M \in Z_n$, sender S chooses a random integer $l \in Z_n^\star$ and computes
$C_1 = (l+1)^e mod\ n$,
$C_2 = Ml^{-1} mod\ n$.
Sends the ciphertext (C_1, C_2) to the receiver.

5.3. **Decryption.** Receiver R who knows the secret key $(p_1, p_2, p_3, ...p_k, d_{p_1}, d_{p_2}, ..., d_{p_k})$ decrypts the ciphertext (C_1, C_2) as below:

R first computes $l_{p_1} = C_1^{d_{p_1}} mod\ p_1, ..., l_{p_k} = C_1^{d_{p_k}} mod\ p_k$ and then he computes l form the above congruence equations via Chinese Remainder Theorem. Finely computes $M = C_2 l mod\ n$.

6. Efficiency and Security Analysis

6.1. **Semantic Security.** An intuitive argument that the proposed cryptosystem is semantically secure against chosen plaintext attack is as follows. In order to determine any information about the plaintext m from the ciphertext (C_1, C_2), attacker need to have some information about $l^{-1}(mod\ n)$, where l is randomly chosen element in $Z_n{}^*$. The only way to ascertain any information about the value of $l^{-1}(mod\ n)$ is to first compute l (it is not sufficient to compute some partial information about l; it is necessary to

have complete information about l in order to obtain any information about $l^{-1}(mod\ n)$, as l is randomly chosen). It is not possible without knowing the secret key d.

6.2. **Comparison With Rprime RSA.** In the Rprime-RSA [2] scheme, the encryption exponent e is taken very large, hence the efficiency of encryption process of the Rprime-RSA cryptosystem becomes very slow. In our proposed scheme, since the pairs like $((l+1)^e mod\ n,\ l^{-1} mod\ n)$ can be computed well in advance, therefore the encryption process requires only one multiplication modulo n, where as in the Rprime-RSA one exponentiation to the power e modulo n is required. In this way, we can say that the encryption process is very fast with comparison to not only Rprime-RSA but also other four variants of RSA (Batch RSA [4], Mprime RSA [3], Mpower RSA [8], Rebalanced RSA [1]). In our proposed scheme, the decryption process requires to compute one extra multiplication modulo n besides the cast on one exponent to the power e modulo n. Thus the decryption speed is computationally as expansive as Rprime-RSA. In addition, the Rprime RSA is not semantically secure whereas our proposed cryptosystem is semantically secure.

6.3. **Comparison with DRSA scheme.** The efficiency of encryption process of D-RSA and our proposed scheme both is same. In the DRSA cryptosystem, the decryption process requires to compute one exponentiation to the power e and to the power d modulo n, one inversion and one multiplication under modulo n. Where as in our proposed scheme, it is require to compute one exponent to the power d modulo n and one multiplication modulo n on average. In this way we conclude that the our proposed scheme is computationally less expensive with comparison to the D-RSA scheme. Hence our proposed scheme is more efficient than that of the D-RSA scheme.

References

[1] Boneh D. and Shacham H., Fast variant of RSA, RSA laboratories, 2002.
[2] Cesar Alison Monteiro Paixao, An efficient variant of the RSA cryptosystem. eprint Archieve/2003.
[3] Collins T., Hopkin D., Langford S. and Sabin M., Public key cryptographic apparatus and method. US patent #5, 848,159, Jan 1997.
[4] Fiat A., Batch RSA. Advances in Cryptology Proceeding of Crypto'89, LNCS pp.175-185, 1989.
[5] Goldwasser S. and Micali M., Probabilistic encryption. Journal of Computer and System Sciences . v.28 pp 270-299, 1984.
[6] Fast decipherment algorithm for RSA publickey cryptosyste. Electronics Letters vol.18, pp.905-907(1982).
[7] Ravist R.L. , Shamir A. and Adllemann L., A method for obtaining digital signature and public key cryptosystems. Comm. Of the ACM 21 , 2 pp. 120-126 (1978).
[8] Takagi T., Fast RSA type cryptosystem modulo pkq. Crypto'98, LNCS vol.1462, Springer Verlag 318-326(1998).

[9] Wiener M., Cryptanalysis of short RSA secret exponents. IEEE Transaction on Information Theory , pp.36(3) 553-558, 1990.
[10] David Pointcheval, New public key cryptosystem based on the dependent RSA problem. Eurocrypt'99 LNCS Springer-Verlag, Vol. 1592, pp.239-254(1999).