International Journal of Information & Network Security (IJINS)
Vol.2, No.1, February 2013, pp. 103~108
ISSN: 2089-3299

An Efficient RSA Cryptosystem with BM-PRIME Method

Sushma Pradhan*, Birendra Kumar Sharma*
* School of Studies in Mathematics,Pt. Ravishankar Shukla University

Article Info

Article history:

Received Oct 5th, 2012
Accepted Oct 30th, 2012

Keyword:

RSA Cryptosystem
Mprime RSA
Batch RSA
QC-RSA
CRT

ABSTRACT

Using more than two factors in the modulus of the RSA cryptosystem has the arithmetic advantage that the private key computation can be speeded up by CRT. With this idea, we present an efficient combination of two variants of RSA cryptosystem (Batch and Mprime RSA) which makes the decryption process faster than the existing variants. It can not only speed up RSA decryption but also guarantee the security of RSA cryptosystem.

Corresponding Author:

Sushma Pradhan
School of Studies in Mathematics,
Pt. Ravishankar Shukla University,
Raipur, Chhattisgarh, India.
Email: sushpradhan@gmail.com

1. INTRODUCTION

The RSA cryptosystem due to Rivest, Shamir and Adleman [10] is one of the most popular public key cryptosystem and widely used to ensure privacy and authenticity of elctronic data. Several variants have been developed to enhance the property of RSA cryptosystem. Boneh [2] has given an excellent survey on simple variants of RSA such as(Batch RSA, Mprime RSA, Mpower RSA, Rebalanced RSA) those are designed to speedup RSA decryption in software.

We review two of the four variants (Batch RSA, Mprime RSA, Mpower RSA, Rebalanced RSA) they analyse in [2], with the goal of reducing the decryption and signature generation times of the original cryptosystem. Firstly, The RSA Multiprime is composed of a modulus N made up with at least three prime factors: $N = p_1, p_2 ... p_r$ with $r \geq 3$. The encryption process is the same as the classical RSA, but decryption and signature generation are performed by using Chinese Remainder Theorem (CRT) which speeds up these operations. Moreover parallel computation can be performed with r exponentiations.We compare the decryption work using the above scheme to the work done when decrypting a normal RSA ciphertext. Recall that standard RSA decryption using CRT requires two full exponentiations modulo n/2-bit numbers. In multi-prime RSA decryption requires b full exponentiations modulo n/b bit numbers.

Secondly, Batch RSA [1]- do a number of RSA decryptions for approximately the cost of one i.e, Fiat [1] showed that, when using small public exponents e_1 and e_2 for the same modulus N, it is possible to decrypt two ciphertexts for approximately the price of one.

The use of more than 2-primes in the RSA cryptosystem has the advantage that the private key operations can be speeded up using the CRT. An easy calculation shows that compared with 2-prime RSA, the theoritical speed up is by a factor of 9/4 for 3-prime RSA and for 4-prime RSA. In practice, a speed up of 1.73 for 3-prime RSA has been achieved [2]. In this, we propose a new method that we called BM-Prime RSA; it is combination of Batch RSA and MPrime (Multi Prime) RSA having the goal of reducing the

decryption time in cryptosystem. This method can not only speed up RSA decryption but also guarantee the security of RSA cryptosystem.

The rest of this paper is organized as follows. In next section, we first give the brief review of RSA cryptosystem. In section 3, we explain the Batch RSA and MPrime RSA. In section 4, we introduce our proposed scheme (BM-Prime RSA) and in section 5, with analysis of some possible attacks related to our proposed cryptosystem. Section 6, presents some results and we conclude in section 7 with some comments on BM-Prime RSA.

2. Review of the basic RSA system

We begin with brief review of the basic RSA pubic key system and refer to [10] for more information. The Key generation, Encryption and Decryption of RSA are as follows:

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 $ed = 1 \bmod \phi(n)$. 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 $C = M^e (\bmod n)$ 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 $C^d (\bmod n) = M$ by using his secret key d.

In 1982 a new technique that recovers M from C, by preprocessing the private key was introduced by J-J.Quisquater and C. Couvreur [8]. This method consists of calculating two integers dp = d mod (p − 1) and dq = d mod (q − 1), and two texts Mp and Mq, where $M_p = C^{dp} \bmod p$ and $M_q = C^{dq} \bmod q$. Applying the Chinese Reminder Theorem (CRT) [7] on Mp and Mq we recover the plaintext M. In this method, we refer to this technique as QC RSA, and to the version created by Rivest, Shamir and Adleman [10] as original RSA. This method is faster because it computes two exponentiations of n/2-bit integers instead of one exponentiation of n-bit integers. Thus we can theoretically speed up the decryption by four.

3. Batch RSA and M-Prime RSA

3.1. Batch RSA: Fiat [1] have shown that, using small public exponents e_1 and e_2 for the same modulus N, it is possible to decrypt two ciphertexts for approximately the price of one. Fait generalized the above observation to the decryption of a batch of b RSA ciphertext. We have b pairwise relatively prime public keys $e_1, e_2 .... e_b$, all sharing a common modulus N. we have b encrypted messages $C_1, C_2 .... C_b$, where Ci is encrypted using the exponent ei. We wish to compute $M_i = C_i^{1/e_i}$ for $i = 1,...b$. Fiat described this b-batch by processing a binary tree for small values of b $(b \le 8)$. One sets $e = \prod_i e_i$ and $A_0 = \prod_i C^{e/e_i}$ (where the indices range over $i = 1,...b$). Then one calculates $A = A_0^{1/e} = \prod_{i=1}^{b} C^{1/e_i}$. For each i one computes Mi as:

$$M_i = C_i^{1/e_i} = \frac{A^{\alpha_i}}{C_i^{\alpha_i - 1/e_i} . \prod_{j \neq i} C_j^{\alpha_i / e_i}}$$

The Batch decryption is extended by QCRSA [8]. Here b-batch requires b modular inversions whereas Fiats tree-based method requires 2 b modular inversions, but fewer auxiliary multiplications. Note that since b and the ei's are small, the exponents in above equation are also small.

With standard 1024-bit keys, batching improves performance significantly. With b = 4, RSA decryption is accelerated by a factor of 2.6; with b = 8, by a factor of almost 3.5. Note that a batch size of

more than eight is probably not useful for common applications, since waiting for many decryption requests to be queued can significantly increase latency.

3.2. Mprime RSA: Mprime RSA was introduced by Collins [5], who modified the RSA modulus so that it consists of k primes $N = p_{1,} p_2 ... p_k$ instead of the traditional two prime's p and q. The key generation, encryption and decryption algorithms are as follows:

3.2.1. Key generation: The key generation algorithm receives as parameter the integer k, indicating the number of primes to be used. The key pairs public and private are generated according to the following steps:

(1) Compute k distinct primes $p_1, p_2, ... p_k$, each one $\left\lceil \frac{\log n}{k} \right\rceil$ bits in length and $N = \prod_{i_1}^{b} p_i$.

(2) Compute e and d such that $d = e_i \bmod(N)$, where $\gcd(e, \phi(N)) = 1. \Phi(N) = \prod_{i=1}^{b} (p_i - 1)$.

(3) For $1 \le i \le k$, compute $d_i = d \bmod(p_i - 1)$.

3.2.2. Encryption: Given a public key N, e_i and a message $M \in Z_N$ encrypt M exactly as in the original RSA, thus $C = M^e (\bmod N)$

3.2.3. Decryption: To decrypt a ciphertext C, first calculate Mi = Cd i mod pi for each i, 1 _ i _ k. Next, apply the CRT to the Mi0s to get. $M = C^d (\bmod N)$.

Observe that this method considers reducing the time expense by modular exponentiation evaluating a larger number of exponentiations with reduced moduli and private exponents. In this way, instead of evaluating, in decryption, a single exponentiation using $\lceil \log N \rceil$dlogNe-bit modulus and with a large private exponent, we will have k exponentiations on moduli of $\lfloor ((\lceil \log n \rceil)) \setminus k \rfloor$ bits and on a reduced private exponent, which is more efficient.

4. Our Proposed BM-Prime-RSA Method

Currently, the recommended key length is 1024 bits. But in the future, we can expect it to be longer. Then it will be possible to have a modulus with more than three prime factors. We assert that the Batch RSA and Mprime RSA methods can be effectively combined. The general idea of this scheme is to use the key generation algorithm of Batch -RSA (modified version b primes) together with the decryption algorithm of Mprime RSA. The new key generation, encryption and decryption algorithm are as follows:

4.1. Key Generation: Let N be the RSA modulus n = log (N) and b be the batch size.

(1) Compute b distinct primes $p_1, p_2, ... p_b$, each one $\left\lceil \frac{\log n}{b} \right\rceil$ bits in length and $N = \prod_{i_1}^{b} p_i$.

(2) Compute e and d such that $d = e_i \bmod(N)$, where $\gcd(e, \phi(N)) = 1. \Phi(N) = \prod_{i=1}^{b} (p_i - 1)$.

(3) For $1 \le i \le b$, compute $d_i = d \bmod(p_i - 1)$

Public Key= $< n, e_{1,} e_2 .... e_b >$; Private Key= $< n, d_{1,} d_2 .... d_b >$.

4.2. Encryption: We have b encrypted messages $c_1, c_2, ..., c_b$ where ci is encrypted using the exponent ei, i.e,

$$C_1 = M_1^{1/e_1} \bmod N$$

$$C_2 = M_2^{1/e_2} \bmod N$$

.
.
.

$$C_i = M_i^{1/e_i} \bmod N \qquad 1 \le i \le b$$

4.3. Decryption: To decrypt a ciphertext c, First calculate $M_i = C_i^{1/e_i} \bmod p_i$ for each i, $1 \le i \le b$. Next, apply the CRT to the Mi's is to get $M = C^{1/e_i} \bmod N$. The CRT step takes negligible time compared to the b-exponentiation.

5. Security and Analysis

(1) The security of BM-Prime RSA depends on the difficulty of factoring integers of the form $N = p_1, p_2, \ldots p_b$ for b > 2. The fastest known factoring algorithm (NFS) cannot take advantage of this special structure of N. However, one has to make sure that prime factors of N do not fall within the range of the ECM which is analyzed in [9]. In our proposed method, the combination makes the decryption process faster. In other words by reducing each c_i (of batch RSA) modulo p_i $(1 \le i \le b)$ combing later these results through the CRT, the decryption process is going faster.

(2) In 2-prime RSA, if the public exponent is small a polynomial time method by Boneh, Durfee and Frankel [3] exists that completely recovers the private exponent once it is partially exposed. However in the 3 and 4 prime case these methods become totally ineffective. The partial key exposure attack for a medium public exponent (i.e, $\sqrt{n}$), also by Boneh, Durfee and Frankel fails in the multiprime case because instead of solving a quadratic congruence, solving congruences of degree r where not all coefficient are known is required.

(3) This result suggests that BM-Prime RSA does not only allow for faster decryption using CRT. But, also somewhat more secure than 2-prime RSA.

(4) Efficiency Performance: We compare the decryption work using the above scheme to the work done when decrypting a normal RSA ciphertext. Recall that RSA decryption using CRT requires two full exponentiations modulo n/2 bits numbers. In our BM-Prime decryption requires b full exponentiation modulo n/b bits numbers.

6. Results

In order to get a better estimate of the performance of decryption of BM-Prime RSA, we compare it with other variants.

6.1. Speed Comparision

We should not analyze cryptographic algorithms with a fixed key length; rather evaluate speed and memory requirements depending on the key length, so that our results wont be out of date if the recommended key length becomes larger in future.

For 768-bits moduli the variant that exhibits better performance would be Batch RSA, but for 1024 and 2048 bits moduli BM-prime RSA presents the best performance. While the speedup of Batch, MPrime and MPower variants is fixed regardless of the size of the used moduli, speedup of the Rebalanced and the RPrime variants [11] significantly increases with larger moduli. This happens because the consideration s fixed and equal to 160 bits (remember that s is the size of the exponent used in decryption algorithm), while this exponent increases for all other variant.

For the applications that prioritize the performance the decryption and the signature generation, the best choice is Bm-Prime RSA, which for 2048-bits moduli got a gain of 30 percent with relation to Rebalanced RSA and is there-fore about 27 times faster than original RSA. Another fact that favors this

variation is that current systems that use MPrime RSA can easily be adapted to it; it is enough to modify the key generation algorithm or create a hybrid key system.

6.2. Memory Comparision

The idea of reducing the decryption time in detriment of the encryption, used by Rebalanced RSA and BM-prime RSA, seems first sight not to present advantages in practical terms. However, there are applications where the balancing characteristic of these algorithms is desirable. Consider, for instance, a situation where the signature generation is executed much more often than verification. A bank, for example, can emit many digital signatures in a single day (in documents, receipts), while the user that receives this signature, has usually a much smaller burden. In this situation is reasonable to transfer the computational effort demanded for the signatures generation to the party verifying them.

Another example is provided by applications running on handheld devices (PDAs), which generally possess limited computational resources. In communications with servers (or even with notebooks or desktop computers), we could leave the task of decryption (fast) for the small device, and the encryption (slow) for the computers with more computational resources. A still better alternative would be to use an implementation of MPrime RSA with keys of the MPrime and BM-Prime RSA, with the use depending on the type of communication (desktop/desktop, or desktop/handheld), in other words, to use a scheme of hybrid keys.

7. Conclusion

We conclude that as the numbers of primes factors in the modulus increases, the attack become more complex, which result in that the attack apply in fewer instances or becomes totally ineffective, or do not seem to extend at all. While our result showed that BMPrime RSA is less vulnerable to current attacks on RSA.It can not only speed up RSA decryption but also guarantee the security of RSA cryptosystem. The benefit of our method is lower computational cost for the decryption and signature primitives, provided that the CRT (Chinese Remainder Theorem) is used. Better performance can be achieved on single processor platforms, but to a greater extent on multiprocessor platforms, where the modular exponentiations involved can be done in parallel. All the RSA variants we discuss apply equally well to digital signatures, where they speed up RSA signing.

REFERENCES

[1] A. Fiat., " Batch RSA ". *Advances in Cryptology: Proceedings of Crypto '89*, pp. 435-175, 1989.
[2] D. Boneh and H. Shacham, " Fast variant of RSA ", *RSA laboratories*, 2002.
[3] D.Boneh, G. Durfee and Y. Frankel, " Exposing an RSA private key given a small fraction of its bit ", *Advances in cryptology- ASIACRYPT'98*, vol. 1514, LNCS, pp. 25-34.
[4] Cesar Alison Monticro Paixao, " An efficient variant of the RSA cryptosystem ", *Cryptology ePrint Archive*, pp. 159, 2003.
[5] Collins T, Hopkin D, Langford S. and Sabin M., " Public key cryptographic apparatus and method ", [*US patent*] 5, 848, 159, 1997
[6] D. Boneh, " Twenty Years of Attacks on the RSA Cryptosystem. ", *Notices of the American Mathematical Society* 46(2), pp. 203213, 1999.
[7] A. Menezes, P. Van Oorschot, and S. Vanstone. " Handbook of Applied Cryptography ". CRC Press, 1997.
[8] J. Quisquater and Couvruur, "Fast decyhpering algorithm for RSA public key cryptosystem*", Electronics Letters*, vol 01, pp. 905-907, 1982.
[9] R. Silverman and S. Wagstaff, " A Practical Analysis of the Elliptic Curve Factoring Algorithm. ", Math. Comp. 61(203), 445462, 1993.
[10] R. L.Rivest, A. Shamir, and L. Adlemann, " A method for obtaining digital signature and public key cryptosystem", *Communication of the ACM*, vol. 2, pp.120-126, 1978.
[11] S. Cavallar,B. Dodson, A. K. Lensra, W. Lioen, P. Montgomery, B. Murphy, H.Riele, K. Aardal, J. Gilchrist, G. Guillerm, P. Leyland, J. Marchand, F. Morain, A. Muffet, C. Putnam and P. Zimmermann, " Factorization of a 512-Bit RSA Modulus. ", *Proceedings of Eurocrypt 2000*, vol-1807(LNCS), pp. 111, Springer-Verlag, 2000.
[12] X. S. Li, *et al.*, "Analysis and Simplification of Three-Dimensional Space Vector PWM for Three-Phase Four-Leg Inverters," *IEEE Transactions on Industrial Electronics*, vol. 58, pp. 450-464, Feb 2011.
[13] R. Arulmozhiyal and K. Baskaran, "Implementation of a Fuzzy PI Controller for Speed Control of Induction Motors Using FPGA," *Journal of Power Electronics*, vol. 10, pp. 65-71, 2010.
[14] D. Zhang, *et al.*, "Common Mode Circulating Current Control of Interleaved Three-Phase Two-Level Voltage-Source Converters with Discontinuous Space-Vector Modulation," *2009 IEEE Energy Conversion Congress and Exposition, Vols 1-6*, pp. 3906-3912, 2009.

[15] Z. Yinhai, *et al.*, "A Novel SVPWM Modulation Scheme," in *Applied Power Electronics Conference and Exposition, 2009. APEC 2009. Twenty-Fourth Annual IEEE*, 2009, pp. 128-131.

BIOGRAPHY OF AUTHORS

Sushma Pradhan received the B.Sc, M.Sc and M.Phill degree in Mathematics Pt. Ravishankar Shukla University, Raipur, Chattigarh, India in 2002, 2004 and 2007. She joined School of Studies in Mathematics, Pt. Ravishnakra Shukla University, Raipur, India for her Research work. She is a life time member of Cryptology Research Society of India (CRSI). Her area of interest is Public Key Cryptography and Integer factorization Problem.

Birendra Kumar Sharma Professor, School of Studies in Mathematics, Pt.Ravishankar Shukla University Raipur (C. G.) India. He has been working for long time in the field of Non Linear Operator Theory and currently in Cryptography. He and his research scholars work on many branches of public key cryptography. He is a life member of Indian Mathematical Society and the Ramanujan Mathematical Society.