Dhaka Univ. J. Sci. 70(2): 15-17, 2022 (July)

ISSN 1022-2502, E-ISSN 2408-8528
DOI: https://doi.org/10.3329/dujs.v70i2.62600

A Variant of the RSA Cryptosystem with Smaller Keys

D M Zunayed Kamal Nibir*, Salma Nasrin and Sarker Md. Sohel Rana

Department of Mathematics, University of Dhaka, Dhaka-1000, Bangladesh

(Received: 6 January 2022 ; Accepted: 25 August 2022)

Abstract

In this paper we introduce an efficient variant of the RSA cryptosystem which will need lesser memory for key storage which is lessen the computational cost. Introduction of the RSA cryptosystem by Rivest, Shamir, and Adleman in 1978 was a revolutionary step in cryptography. The regular RSA system needs huge cost for decryption due to large size of the private decryption key. The proposed variant will be a solution to this conundrum.

Keywords: Public key cryptosystem, RSA, security, cryptography.

I. Introduction

Information is said to be the most valuable asset of this era. So the protection of information is the main challenge nowadays. Information is most vulnerable while passing, as it can be intercepted during the transfer. The art of cryptography has been used to conceal the data for a long time in the history of mankind. Before the introduction of computers, cryptography depended on the use of pen, paper and other mechanical aids. Until the 1970s, secure cryptography relied on the same encryption and decryption keys. Two occasions since then made the secure passing available to public using entangled keys which are not easy to intercept: the making of a public information encryption standard (DES)[1,2] in 1975, and the development of public-key cryptography[3] in 1976. After that the development of the RSA cryptosystem[4] by Rivest, Shamir, and Adleman represented an incredible benefit in the field of cryptography. The main fallback for the RSA cryptosystem is the decryption process. Generally the decryption exponent d is large, which increases the costing of decryption. To overcome this problem, mathematicians have introduced some variations to the RSA cryptosystem and some are still being proposed. Initially a comparison of Batch RSA[5], Multiprime RSA[6], Multipower RSA[7] and Rebalanced RSA[8] was published in 1999 by Boneh[9]. In 2003, RPrime RSA[10] was proposed combining the variants Rebalanced RSA and Multiprime RSA methods. In this paper we propose an efficient variant to this cryptosystem where we enhance the memory allotment using smaller primes with larger power.

II. The RSA Cryptosystem

We briefly review the working procedure of the RSA cryptosystem in this section. The security of this system depends on the difficulty of factoring a very large integer. The RSA cryptosystem is based on the following Theorem[11] of number theory. Here $\phi(n)$ denotes Euler's totient function, which gives the number of co-primes of n not exceeding n.

Theorem: Let p, q be distinct primes. Let $n = pq$ and

$$m = \phi(n) = (p - 1)(q - 1).$$

Let $e, d \in \mathbb{Z}_m$ such that $ed \equiv 1(\mathrm{mod}(m))$. Then for every integer x, $x^{ed} \equiv x(\mathrm{mod}(n))$.

The values n and e constitute the public key of RSA system. If the information x is to be passed, the □ mapping $f(x): \mathbb{Z}_n \to \mathbb{Z}_n$ gives enciphering algorithm where $f(x) = x^e\ (\mathrm{mod}\ (n))$. Deciphering is done by the mapping $g(x): \mathbb{Z}_n \to \mathbb{Z}_n$ where $g(y) = y^d\ (mod\ (n))$. This is valid by the theorem stated above as $g(f(x)) = (x^e)^d\ (\mathrm{mod}\ (n)) = x$. Hence g is the inverse of f. Here e and d are respectively called the enciphering and deciphering exponents.

The difficulty of breaking the code depends on finding the factors of n, which are p, q. If factors are available, the eavesdropper can find d as n and e are public and thus can find x from the code y. Using the usual convention we will call the sender Alice, the receiver Bob and the eavesdropper as Eve.

There are mainly three steps of the RSA cryptosystem:

- Key creation
- Encryption
- Decryption

Key creation: In this step Bob, the receiver, chooses two prime numbers p and q. Then he computes the numbers $n = pq$ and $\phi(n) = (p - 1)(q - 1)$. The number n is called the *modulus*. Then he chooses the number e with the property such that $gcd(e, (p - 1)(q - 1)) = 1$. This is the *encryption exponent* and the pair (n, e) constitute the public key. Receiver exposes the public key so that anybody can send a message to him. He solves the congruence $de \equiv 1(\mathrm{mod}\ \phi(n))$ to obtain d. As factors of n are not public, Eve cannot find $\phi(n)$ and hence d is secure. This value d is the *decryption exponent* and (n, d) is the private key.

*Author for correspondence. e-mail: zunayed.kamal@du.ac.bd

Encryption: Alice converts her plaintext to an integer m such that $1 \leq m \leq n$. Then she uses Bob's public key (n, e) to compute $c \equiv m^e \pmod{n}$. Alice sends the ciphertext c to Bob.

Decryption: Bob takes the ciphertext c and using his private key (n, d) computes $m_1 \equiv c^d \pmod{n}$. The value m_1 he computes is equal to m.

In this way Bob retains the original message sent to him after the decryption procedure.

III. Variants of RSA

The main fallback for the RSA cryptosystem can be said as the decryption process. Generally the decryption exponent d is large, which increases the costing of decryption. We are describing some modifications which are improved variant of the RSA cryptosystem.

Batch RSA

In this variant[5], two ciphertexts are decrypted for single cost when small public exponents are used for same modulus n. The insight is viewing the deciphering of RSA in a different way: Deciphering a ciphertext in RSA is taking the eth root modulo n, where e is the encrypting exponent. If c is the ciphertext, we can write $c^d \pmod{n} \equiv c^{\frac{1}{e}} \pmod{n}$, where $de \equiv 1 \pmod{n}$. The encryption method for batch RSA is same as the regular RSA. This method is efficient when the public exponents are small. Otherwise it will slow down the decryption making the process more expensive. Also the modulus must be the same and public exponents must be distinct for both the messages[12]. With standard 1024-bit keys, a considerable improvement is observed[5].

Multiprime RSA

In 1982, a new decryption method for RSA was published[13] which used the Chinese Remainder Theorem for decryption process. In 1998, Multiprime RSA[6] was introduced. After that some modifications[14] were given in 2018. It uses product of k primes as the modulus instead of just two. Though only one encryption exponent is used, k decryption exponents are formed.

Multipower RSA

In 1997, Takagi published a fast decryption method using n-adic expansion[15]. The algorithm is attached in appendix A. Then in 1998, He proposed Multipower RSA[7] where the modulus is $n = p^k q$. This system reduced the key size and enhanced the memory allocation.

Rebalanced RSA

Rebalanced RSA[8] enhances the timing of the decryption algorithm by passing the work to the encryption algorithm. We may take d to be small, but increases the vulnerability of RSA[16]. So d is chosen as a large number, but $d(\text{mod }(p-1))$ and $d(\text{mod }(q-1))$ are small numbers.

RPrime RSA

This scheme[10] amalgamates the key generation algorithm of Rebalanced RSA (modified for k primes) and the decryption algorithm of Multiprime RSA.

IV. Proposed System

In our proposed scheme, we are trying to enhance the memory usage taking smaller primes with larger power. In Multipower RSA[7], the modulus used is $n = p^k q$. We are using a modulus $n = p^k q^l$. The algorithm of this scheme are as follows:

Key creation: In this step two distinct primes p and q are chosen and $n = p^k q^l$ is computed. We compute $L = \text{lcm}((p-1), (q-1))$ and find out e and d where $de \equiv 1(mod\, L)$ and $gcd(e, L) = 1$. We take $d_p \equiv d(mod\,(p-1))$ and $d_q \equiv d(mod\,(q-1))$. Here (n, e) is the public key and (p, q, d_p, d_q) is the secret key.

Encryption: Encryption is the same as original RSA. The plaintext is converted to an integer m such that $1 \leq m \leq n$. Then the public key (n, e) is used to compute $c \equiv m^e$ (mod n). Finally the ciphertext c is sent to receiver. Here m must be coprime with n.

Decryption: For decryption process at first $m_1 \equiv c^{d_1}\,(mod\, p)$ and $m_2 \equiv c^{d_2}\,(mod\, q)$ are computed. Hence $m_1^e \equiv c\,(mod\, p)$ and $m_2^e \equiv c\,(mod\, q)$. After that using the process of n-adic expansion[15] m_p and m_q are computed. The pseudocode for n-adic expansion is:

Pseudocode for n-adic expansion:

Here $\text{mod}(p, q)$ is $p(\text{mod } q)$.

We find out m_p in Multipower RSA and our proposed system where p^k is in the modulus.

Input: c, e, p, d_p, k and Output: m_p

```
• $k_0 = \text{mod}(c^{d_p}, p)$
• $a = k_0$.
• $for\ i = 1 : k - 1$
        $f = \text{mod}(a^e, p^{i+1})$;
        $e1 = \text{mod}(c - f, p^{i+1})$;
        $b1 = e1/(p^i)$;
        $in = \text{mod}((e.f)^{-1}, p)$;
        $k_0 = mod(in.a.b1, p)$;
        $a = a + (p^i).k0$;
$end$
• $m_p = a$.
```

Finally we use the Chinese Remainder Theorem to compute m such that $m \equiv m_p \pmod{p^k}$ and $m \equiv m_q \pmod{q^l}$ to obtain our required decrypted message.

The pseudocode for the Chinese Remainder Theorem:

Here $\text{mod}(p, q)$ is $p(\text{mod } q)$.

Let us solve $x \equiv m_p(mod\ p)$ and $x \equiv m_q(mod\ q)$

```
Input: $p, q, m_p, m_q$ and Output: $m$
• $p_1 = mod(p^{-1}, q)$;
• $q_1 = mod(q^{-1}, p)$;
• $x = mod(p.p_1.m_q + q.q_1.m_p, p.q)$.
```

V. Numerical Validation

We present a numerical example to support our result. The computing of m_p,m_q and m in decryption part are directly done by MATLAB program written using the pseudocodes mentioned in part IV.

Key creation: Bob selects the prime numbers $p = 17$ and $q = 29$ with powers $k = 2$ and $l = 2$. Then the modulus is $n = p^2.q^2 = 289.841 = 243049$. Here, $L = \text{lcm}(16,28) = 112$. Also let us assume that he selects $e = 3$ as the encryption exponent, which is coprime to 112. Then solving $3d \equiv 1\ (mod\ 112)$ he gets $d = 75$. Thus $d_p \equiv 75(\text{mod } 16)$ or $d_p = 11$ and $d_q \equiv 75(\text{mod } 28)$ or $d_q = 19$. So the public key is $(243049{,}3)$ and the private key is $(17{,}29{,}11{,}19)$.

Encryption: Alice converts the plaintext to integer 38025 which is less than 243049 and they are coprime. Then she calculates $c \equiv 38025^3(mod\ 243049) \equiv 72832(mod\ 243049)$. Then she sends this integer 72832 to Bob.

Decryption: Using the n-adic expansion Bob gets $m_p = 166$. In a similar way he finds that $m_q = 180$. Now for m, she uses the Chinese Remainder Theorem on these two congruences $x \equiv 166(\text{mod } 289)$ and $x \equiv 180(\text{mod } 841)$. The value Bob computes is Alice's message $m = 38025$.

VI. Conclusion

Generally, the RSA modulus is product of two large primes, hence the computing time is much high and a large amount of memory is used. This new system uses product of powers of primes as modulus. With this new modulus the memory required will be less and computing time will be low. Also, public key is the same as regular RSA, so interceptor cannot decide which decryption is to be used. It has the drawback that the message must be coprime to the modulus.

Acknowledgement

This research is fully funded by the "Centennial Research Grant" at the University of Dhaka.

References

1. Office of the Federal Register, National Archives and Records Administration, 1975. *Federal Register,* **40(52)**, 12134-12138.
2. Office of the Federal Register, National Archives and Records Administration, 1975. *Federal Register*, **40(149)**, 32395-32414.
3. Diffie, W., M. Hellman, 1976. New directions in cryptography. *IEEE Trans. Inform. Theory,* **IT-22(6)**, 644-654.
4. Rivest, R., A. Shamir and L. Adelman, 1978. A Method for Obtaining Digital Signatures and Public-key Cryptosystems. *Communications of ACM*, **21(2)**, 120-126.
5. Fiat, A., 1997. Batch RSA. *J. Cryptology,* **10**, 75-88.
6. Collins, T., D. Hopkins, S. Langford, and M. Sabin, 1998. Public Key Cryptographic Apparatus and Method. s.l. Patent No. 5,848,159.
7. Takagi, T., 1998. Fast RSA-type cryptosystem modulo $p^k q$. *In: Advances in Cryptology - CRYPTO '98.* Springer Berlin Heidelberg, 318-326.
8. Wiener, M., 1990. Cryptanalysis of short RSA secret exponents. *IEEE Transactions on Information Theory*, **36(3)**, 553-558.
9. Boneh, D., 1999. Twenty years of attacks on the RSA cryptosystem. *Notices of the American Mathematical Society (AMS)*, **46(2)**, 203-213.
10. Paixão, C., 2003. An efficient variant of the RSA cryptosystem. *IACR Cryptology ePrint Archive*, **2003**, 159.
11. Nagpaul, S. and Jain, S., 2005. *Topics in Applied Abstract Algebra.* s.l.:Thomson Brooks/Cole.
12. Verma, S. and D. Garg, 2009. Improvement over Public Key Cryptographic Algorithm. *In: IEEE International Advance Computing Conference (IACC 2009)*, 734-739.
13. Quisquater, J. and C. Couvreur, 1982. Fast decipherment algorithm for RSA public-key cryptosystem. *Electronics Letters*, **18(21)**, 905-907.
14. Kamardan, M., N. Aminudin, N. Che Him, S. Sufahani, K. Khalid, and R. Roslan, 2018. Modified Multi Prime RSA Cryptosystem. *Journal of Physics: Conference Series*, **995**, 012030.
15. Takagi, T., 1997. Fast RSA-type cryptosystems using n-adic expansion. *In:Advances in Cryptology - CRYPTO '97.* Springer Berlin Heidelberg, 372-384.
16. Boneh, D. and G. Durfee, 2006. Cryptanalysis of RSA with Private Key d Less than $N^{0.292}$. *IEEE Trans. Inf. Theor.*, **46(4)**, 1339–1349.