Efficient Implementation of RSA Algorithm with MKE

Sami A. Nagar and Dr. Saad Alshamma

Sudan university of Science and Technology Electronic Engineering College / Department of Communication - Khartoum - Sudan

Abstract

The aim of this paper is to improve the implementation of RSA algorithm in some certain situations. This improvement is based on the ideas of H. Ren-Junn, S. Feng-Fu, Y. Yi-Shiung and C. Chia-Yao and allows us to speed up the transmission of data between various nodes in networks.

We realized our idea in the C# language and in the database that was created by SQL Server 2008 R2. Identical database must be used in all networks gateways. The special protocol (RSA Handshake Database Protocol) was designed for controlling the creation of this database. Each gateway, that runs a RSA-Key Generations Offline procedure, is controlled according to specific issues and necessaries. This stage is called RSA-Key Generations Offline.

We propose the new method for exchanging values of the keys among gateways. Roughly speaking gateways exchange indexes (Indexes Exchange) that refer to fields that contain the values of public and private keys.

Key words: RSA, RSA Handshake Database Protocol, RSA-Key Generations Offline.

1 Introduction

The well-known RSA algorithm is very strong and useful in many applications. But it is not used so often in smart cards for its big computational cost.

Bahadori [BAH 10] implemented the new approach for secure and fast key generation of a key pair for RSA. This method was implemented on a typical smartcard equipped with a crypto-coprocessor and a random generator. An efficient method for generating a large random prime number is proposed and this considerably reduces the total time required for generating a key pair. That is up to 50% reduction in total generation time compared to the latest reported methods.

Blackburn [BLA 00] proposed a joint method RSA key generation shared between a user and a certification authority (CA). The CA is convinced that the user's key has been well

generated, but does not obtain significant information about the user's secret RSA decryption key.

H. Ge and S. R. Tate [HGE 06] proposed the efficient authentication method for secure communication in a set of devices that have a single trusted administrator. An example of such system is a set of devices owned by a single person, with emphasis on the simplicity and efficiency of the protocol. While known techniques can solve this problem, they show how specific properties of their setting can allow more efficient solution, which is more appropriate for embedded processors with limited computational capabilities. Specifically, a device using the proposed protocol can authenticate itself using only about 15% of the computation required by a standard RSA signature-based authentication. The proposed scheme is secure under the strong RSA assumption and the computational Diffie-Hellman assumption.

To generate the RSA keys efficiently on a low-power handheld device, Chen et al. [TIA 06] proposed two improved protocols and claimed that their protocols are secure against the collusion attack. The one is a standard RSA key generation protocol and the other is an unbalanced version. This letter point out a weakness in Chen et al.'s unbalanced RSA key generation protocol. If the servers collude with each other they can derive the user's secret prime with high probability that enable the decryption of any ciphertext.

A. Selby and C. Mitchell [SEL 89] proposed two new algorithms that facilitate the implementation of RSA in software. Both algorithms essentially deal with performing modular arithmetic operations on very large numbers, which could be of potential use to applications other than RSA. One algorithm performs modular reduction and the other performs modular multiplication. Both algorithms are based on the use of look-up tables to enable the arithmetic computations to be done on a byte by byte basis .

The theory of Public Key Cryptography, one of the important topics discussed in the conference PKC 2011 is ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization; generic constructions for chosen-ciphertext secure attribute based encryption; expressive key-policy attribute-based encryption with constant-size ciphertexts.

H. Ren-Junn, et al [REN 05] proposed an efficient method to implement RSA decryption algorithm. RSA cryptosystem is the most attractive and popular security technique for many applications, such as electronic commerce and secure Internet access. It has to perform modular exponentiation with large exponent and modulus for security consideration. The RSA cryptosystem takes great computational cost. In many RSA applications, user uses a small public key to speed up the encryption operation. However, the decryption operation has to take more computational cost to perform modular exponentiation by this case.

H. Ren-Junn, et al proposed an efficient decryption method not only based on Chinese remainder theorem (CRT) but also the strong prime of RSA criterion. The proposed decryption method only takes 10% computational costs of the traditional decryption method. It also reduces 66% computational costs than that of decryption methods based on CRT only.

In a word, the speed of our proposed method is almost 2.9 times faster than the decryption method based on CRT only. The proposed method enhances the performance of the RSA decryption process.

2 Implementation of RSA Algorithm

2.1 Offline RSA-Key Generations

In this paper we increase the speed of the RSA implementation by generating keys offline and storing them in different databases before the beginning of using a RSA key pair in encryption/ decryption processes.

RSA-Key Generations Offline is the new software component which we developed by using C# language to increase the speed of RSA implementation [NAG 11] [WEL 01], also we needed database engine for saving calculated values into two tables. Table "one" contains values of p, q, n and $\varphi(n)$, and table "two" contains e and d values.

$$n = pq, \tag{1}$$

$$\varphi(n) = (p-1)(q-1), \tag{2}$$

$$GCD(e, \varphi(n)) = 1, \tag{3}$$

$$d = e^{-1} \pmod{\varphi(n)}. \tag{4}$$

The figure 1 shows each tow tables create a new set and each set has a unique set ID which is called Setid.

Database has many numbers of sets, these numbers of sets are determined by many factors, for example the prime numbers p, q length and by their possibilities for producing n values, the Setid turns the searching for exact set fast and easy, we add difficulty concept for knowing what set we are using now.

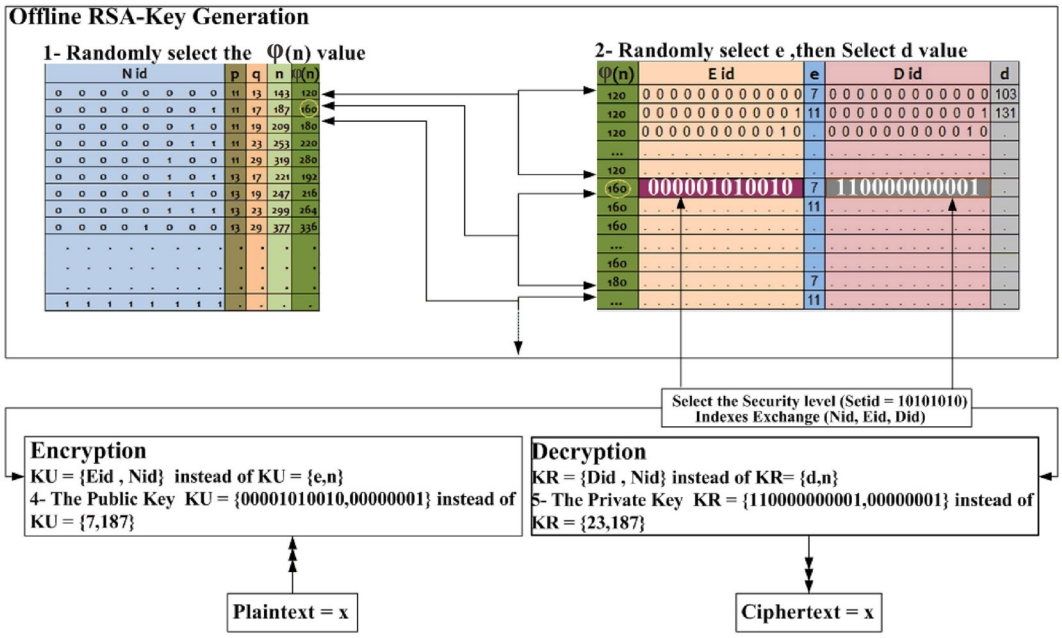

Figure 1. Offline RSA-Key Generations and Indexes Exchange.

2.2 Online Encryption and Decryption Processes

In this paper we propose four security levels. Each level has own database and consists of many sets, these levels identifiers by property of e values and the key length see table 1.

Security Level	Key Length
Low	512 bits
Medium	1024 bits
Medium -High	2048 bits
Security High	4096 bits

Table 1. Security Levels.

The gateways (users) must select the same security level or change the security level before starting the encryption and decryption processes.

We select SQL Server 2008 R2 as database engine for creation of databases and their sets which content the keys values. Also we select SQL Server 2008 R2 for keeping our database saves and secure, by encrypted all data without increasing database size or impacting performance and it has Guard against security breaches if backups or disks are lost or stolen.

We propose to use RSA key pair between LAN's / WAN's gateways instead of users.

Using of private and public keys between gateways that means the RSA encryption/decryption algorithm now is suitable for large amount of data flowing between gateways and this infer of uses the RSA-Key Generations Offline Algorithm, in figure 2 we explain schematic of RSA Algorithm Processes.

We also propose a new protocol called RSA Handshake Database Protocol, this protocol responsible for creation the identical RSA-Key Generations Offline databases in all network gateways and organize database update if require and execute the procedure for each new gateway want to use the RSA-Key Generations Offline database with existing gateways.

The RSA Handshake Database Protocol saves the selected security level (database), which set selected in the security level (Setid), keys indexes and another data in working information table.

The RSA algorithm starts using the data from working information table for encryption/ decryption processes between network gateways.

The RSA Handshake Database Protocol controls all initially processes and any changes in the security levels and key length between the gateways or new gateway would like to join an existing session.

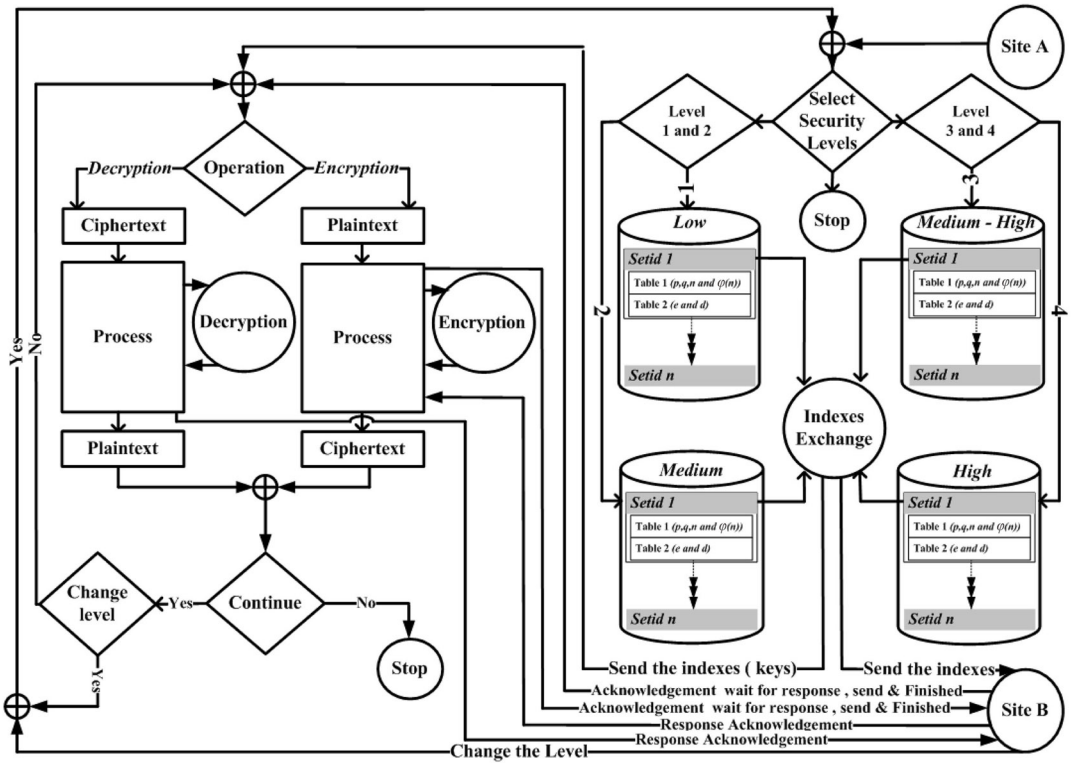

Figure 2. Schematic of RSA Algorithm Processes.

3 Exchange the keys indexes

In this paper we proposed a new method called Indexes exchange, where we use the Indexes exchange instead of keys exchange between different gateways, example in table 2 explaining how the indexes will be exchanged instead of n, e and d values.

Keys Exchange		Indexes Exchange	
n	160	Nid	00000001
e	7	Eid	000001010010
d	23	Did	110000000001

Table 2. Example of use the Indexes Exchange instated of keys exchange.

By using the indexes exchange instead of keys exchange it will be very hard to get the n, e and d values even if you know the indexes of these values.

4 Experiments and results

With using RSA-Key Generations Offline Algorithm and different keys lengths, the decryption processes is 2.5 times faster than online RSA keys generations.

The timings were made on a 2.8GHz Pentium by using the below factors:

- Block size is 2048 bits.
- Different bandwidths:
  (a) 1000 Mbps.
  (b) 100 Mbps.
  (c) 4 Mbps.

The figure 3 shows the compare between RSA decryption process by using RSA-Key Generations Offline method and online RSA key generation's method, decryption by RSA-Key Generations Offline is faster than using normal RSA key generations.

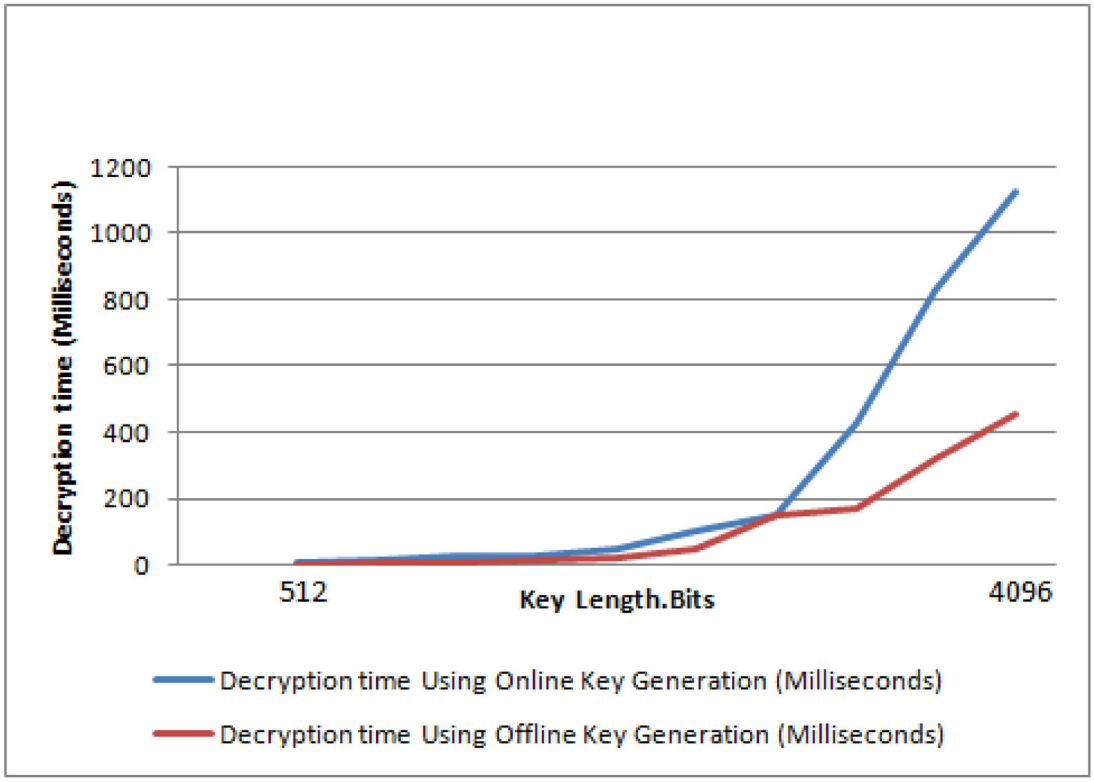

Figure 3. Compare between decryption processes using online and offline RS-key generations.

5 Conclusion

In this paper, we speed up the RSA algorithm through developed a new generation keys method called RSA-Key Generations Offline to generate and saved all keys values in tables within the database.

We propose four security levels, each level has its own database and numbers of sets, these levels identified according to the e values and keys length, before start using the RSA algorithm between gateways must get a Ready Acknowledgment from RSA Handshake Database protocol, this protocol responsible for creation or update the identical gateways database, level selections (Setid) and establishment the algorithm between gateways.

In this paper we propose the new method of keys exchanging of increase the difficulty for anyone (adversary) who knows the exchanged values between gateways, and then try to get the n, e and values, this method we called Indexes exchange, where we exchange the indexes Nid, Eid, Did instead of n, e, d values.

References

[BAH 10] M. Bahadori, M. R. Mali, O. Sarbishei, M. Atarodi and M. Sharifkhani "A novel approach for secure and fast generation of RSA public and private keys on Smart-Card" *NEWCAS Conference (NEWCAS), 2010 8th IEEE International, 2010, pp. 265-268.*

[BLA 00] S. R. Blackburn and S. D. Galbraith "Certification of secure RSA keys" *Electronics Letters, vol. 36, pp. 2930, 2000.*

[HGE 06] H. Ge and S. R. Tate "Efficient Authenticated Key-Exchange for Devices with a Trusted Manager" *Information Technology: New Generations, 2006 (ITNG 2006). Third International Conference on, 2006, pp. 198-203.*

[JOS 08] J. Joshi, et al. *"Network Security"* Morgan Kaufmann, 2008.

[NAG 11] C. Nagel, B. Evjen, J. Glynn, K. Watson and m. Skinner *"Professional C# 2008"* Wrox, 2011.

[REN 05] H. Ren-Junn, S. Feng-Fu, Y. Yi-Shiung and C. Chia-Yao "An efficient decryption method for RSA cryptosystem" *Advanced Information Networking and Applications, 2005 (AINA 2005). 19th International Conference on, 2005, pp. 585-590 vol.1.*

[RIV 78] R. L. Rivest, A. Shamir and L. Adleman "A method for obtaining digital signatures and public-key cryptosystems" *Communications of the ACM, vol. 21, pp. 120-126, 1978.*

[SEL 89] A. Selby and C. Mitchell "Algorithms for software implementations of RSA" *Computers and Digital Techniques, IEEE Proc., vol. 136, pp. 166-170, 1989.*

[STA 00] W. Stallings *"Network security Essentials: Applications and Standards"* Pearson Education India, 2000.

[STA 03] W. Stallings *"Cryptography and network security* vol. 2" prentice hall, 2003.

[STA 95] W. Stallings *"Network and internetwork security: principles and practice"* Prentice-Hall, Inc., 1995.

[TIA 06] C. Tianjie and M. Xianping "Collusion Attack on a Server-Aided Unbalanced RSA Key Generation Protocol" *Communication Technology, 2006(ICCT 2006). International Conference on, 2006, pp. 1-3.*

[WEL 01] M. Welschenbach *"Cryptography in C and* C++" Springer-Verlag New York, 2001.