# High Speed Implementation of RSA Algorithm with Modified Keys Exchange

Sami A. Nagar and Saad Alshamma
*Faculty of Electronic Engineering,*
*Sudan University of Science and Technology,*
*Khartoum, Sudan*
*elnajarsami@yahoo.com, Saaddoud2003@yahoo.com*

***Abstract*— This paper aims to speed up the implementation of the RSA algorithm during data transmission between different communication networks and Internet, which is calculated to generate the keys by a program prepared in a C # language and then save these values of the keys in the databases created by SQL Server 2008 R2.Within indexed tables, this stage is called RSA-Key Generations Offline as it is an inevitable stage carried out in each gateway before using the RSA algorithm. In RSA algorithm identical database must be used in all networks gateways, the creation of the database controlled by a special protocol programmed in a C # language called RSA Handshake Database Protocol, the protocol controls each gateway that runs a RSA-Key Generations Offline according to specific issues and necessaries. In this paper a new method to exchange the values of the keys between gateways, which are exchanged indexes (Indexes Exchange) refers to the fields that contain the values of public and private keys that are stored in the tables inside the database before starting to use RSA algorithm to encrypt and decrypt the data, rather than using the exchange of real values n, e, and d.**

***Keywords*- RSA, RSA-Key Generations Offline, RSA Handshake Database Protocol, Cryptography, Cryptosystem, secret-key, private-key, public-key, C#, Indexes exchange, SQL Server 2008 R2, Setid, Nid, Eid, Did.**

## I. INTRODUCTION

As the telecommunication network has grown explosively and the internet grows rapidly, information security becomes more and more significant. Cryptography is knowledge of protecting the secret information, and the cryptosystem can be distinguished into two types, secret-key cryptosystem and public-key cryptosystem.

In secret-key cryptosystem, the plaintext and the ciphertext are encrypted and decrypted by the same key, thus it is also called as symmetric cryptosystem. Though secret-key cryptosystem is easily to implement due to less computation, , it has several drawbacks, too many keys, key distribution problem, authentication and nonrepudiation problem.

The public-key cryptosystem evolves to solve the problems of symmetric cryptosystem, and RSA cryptosystem is the most popular approach. The RSA cryptosystem was developed in 1977 by Ronald L. Rivest, Adi Shamir, and Leonard Adleman at MIT and first published in 1978 [7].

Although RSA algorithm is very secure, it is rarely used in smart card, due to its long computation time. It is primarily used in the field of digital signatures, following the example. Some components, such as multi-leveled equations, graphics, and tables are not prescribed, although the various table text styles are provided. The formatter will need to create these components, incorporating the applicable criteria that follow. where the benefits of an asymmetric procedure can be fully realized.

RSA Cryptosystem, Asymmetric encryption is relatively slow and therefore unsuitable for encryption of large messages [11] [9][4] [10].

The RSA public-key cryptosystem was developed by R.L. Rivest, A. Shamir, and L. Adleman in 1978[7]. The RSA cryptosystem is simply the modular exponentiation.

The modulus n is the product of two large prime's p and q, public key and private key are obtained by:

$$e = d\text{-}1 \bmod Ø(n) \quad (1)$$

The encryption operation is performed using the public key n and e as follows:

$$C = Me \pmod n \quad (2)$$

Where M is the plaintext such that $0<M<n$ and C is the ciphertext which can be decrypted using the private key n and d as follows:

$$M = C\ d \pmod n \quad (3)$$

## II. RELATED WORK

Bahadori implements a novel approach for secure and fast key generation of the public key cryptographic algorithm of RSA, This method has been implemented on a typical smartcard equipped with a crypto-coprocessor and a true random number generator. An efficient method for generating the large random prime numbers is proposed that considerably reduces the total time required for generating a key pair [1]. That is up to 50% reduction in total generation time compared to the latest reported methods.

Blackburn proposed a joint method RSA key generation by a user and a certification authority (CA). The CA is convinced that a user's key has been well generated, but does not obtain

significant information about the user's secret RSA decryption key [2]. peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

H. Ge and S. R. Tate are proposed an efficient authentication method for secure communication among a set of devices that have a single trusted administrator, with protocols presented for authentication and authenticated key exchange. An example of such a setting would be 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 th strong RSA assumption and the computational Diffie-Hellman assumption [3].

To generate the RSA keys efficiently on a low-power handheld device, Chen et al. 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 [12].

A. Selby and C. Mitchell are proposed two new algorithms that facilitate the implementation of RSA in software are described. Both algorithms are essentially concerned 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 [8].

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.

In Cryptanalysis of the RSA subgroup assumption from TCC 2005; sub-linear, secure comparison with two non-colluding parties; oblivious transfer with hidden access control policies; chosen ciphertext secure encryption under factoring assumption revisited; and parallel decryption queries in bounded chosen ciphertext attacks .

H. Ren-Junn, et al are 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 are 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 operation [6].

## III. IMPLEMENTATION OF RSA ALGORITHM

### *A. Offline RSA-Key Generations*

In this paper we increased the RSA implementation speed by generated keys offline and stored in different databases before starts using the RSA key pair in encryption/ decryption processes.

RSA-Key Generations Offline is a new software component we developed by using C# language to increases the speed of RSA implementation [5] [13], also we need database engine to save the calculated values inside two tables, table one includes the values of p, q, n and Ø (n), and table two includes e and d values.

$$n = p \times q \tag{4}$$

$$Ø(n) = (p-1)(q-1) \tag{5}$$

$$e = \text{relatively prime to } Ø(n) \tag{6}$$

$$d = e\text{-}1 \bmod Ø(n) \tag{7}$$

Figure 1 shows each tow tables create a new set and each set it has a unique set ID called Setid.

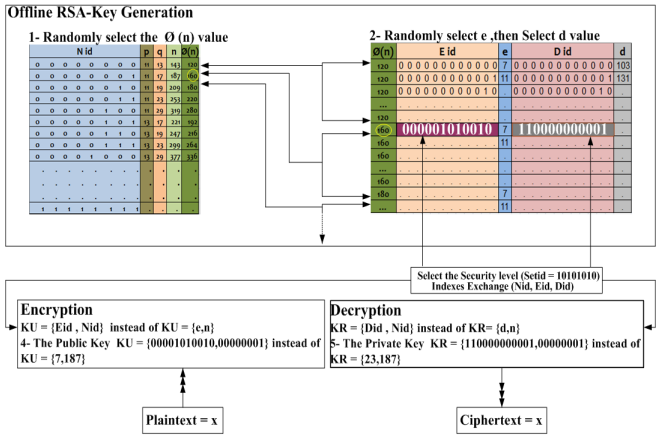


Figure 1. Offline RSA-Key Generations and Indexes Exchange

Database has many numbers of sets, these number of sets determined by many factors, for example the prime numbers p, q length and their possibilities to produce n values, the Setid makes the searching for exact set fast and easy , we added difficulty concept to know which set we are using now.

### *B. Online Encryption and Decryption Processes*

In this paper we proposed four security levels each level has own database and consists of many sets, these levels identifiers by possibility of e values and the key length see table I.

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

We select SQL Server 2008 R2 as database engine for creation the databases and their sets which contents the keys values, also we select SQL Server 2008 R2 to keep 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.

TABLE I. SECURITY LEVELS

| Security Level | Key Length (bit) |
|---|---|
| Low | 512 |
| Medium | 1024 |
| Medium-High | 2048 |
| High | 4096 |

In this paper we proposed 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 flow between gateways and this infer of uses the RSA-Key Generations Offline Algorithm, in figure 2 we explain schematic of RSA Algorithm Processes.

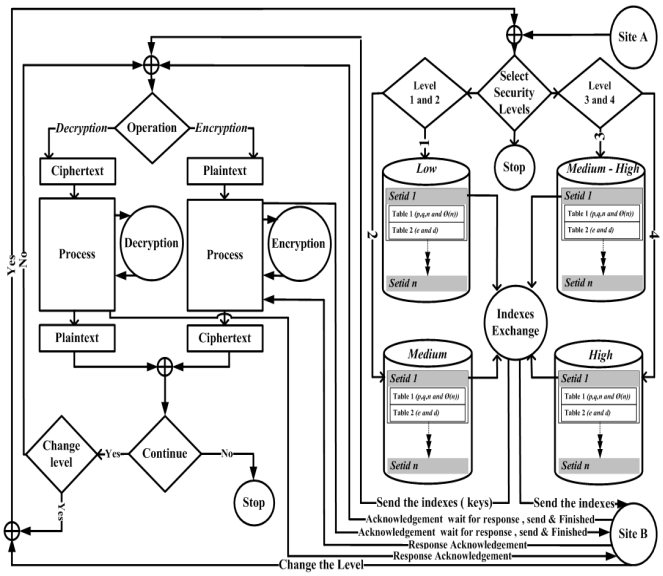


Figure 2. Schematic of RSA Algorithm Processes

In this paper we proposed 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.

## IV. 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 II explaining how the indexes will be exchanged instead of n, e and d values.

TABLE II

EXAMPLE OF USE THE INDEXES EXCHANGE INSTATED OF KEYS EXCHANGE

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

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.

## V. 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:
  1. 1000 Mbps.
  2. 100 Mbps.
  3. 4 Mbps.

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.

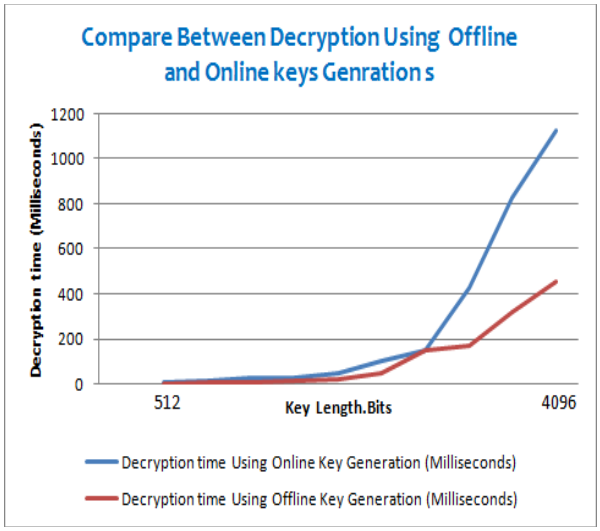


Figure 3. Compare between decryption processes using online and offline RSA- key generations

## VI. CONCLUSION

In this paper, we speedup 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 database.

We proposed 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 proposed a new method of keys exchange to increase the difficulty for any one 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.

## ACKNOWLEDGMENT

The authors wish to thanks Dr. Izzeldin Ibrahi Mohamed Abdelaziz and Dr. Mohammed Nadzier b. Marsono from Faculty of Electrical Engineering University Technology Malaysia (UTM), Johor Malaysia for their comments and valuable suggestion.

The authors wish to thank the management of YIC Foundation for their continued support and understanding.

## REFERENCES

[1] 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 SmartCard" NEWCAS Conference (NEWCAS), 2010 8th IEEE International, 2010, pp. 265-268.

[2] S. R. Blackburn and S. D. Galbraith "Certificationof secure RSA keys" Electronics Letters, vol. 36, pp. 29-30, 2000.

[3] H. Ge and S. R. Tate "Efficient Authenticated Key-Exchange for Devices with a Trusted Manager" Information Technology: New Generations, 2006 (ITNG2006). Third International Conference on, 2006, pp.198-203.

[4] J. Joshi, et al. "Network Security" Morgan Kaufmann, 2008.

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

[6] 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.

[7] 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.

[8] A. Selby and C. Mitchell "Algorithms for software implementations of RSA" Computers and Digital Techniques, IEE Proceedings E, vol. 136, pp. 166-170, 1989.

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

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

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

[12] 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.

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