International Journal of Information Technology and Knowledge Management
July-December 2012, Volume 5, No. 2, pp. 239-242

IMPLEMENTATION OF PUBLIC KEY CRYPTOGRAPHIC SYSTEM: RSA

Punita Meelu[1] and Rajni Meelu[2]

ABSTRACT: In many business sectors secure and efficient data transfer is essential. To ensure the security to the applications of business, the business sectors use Public Key Cryptographic Systems (PKCS). An RSA system generally belongs to the category of PKCS. RSA encryption is one of the public-key methods that have been popular in last decade. In particular, the RSA algorithm is used in many applications. Although the security of RSA is beyond doubt, the evolution in computing power has caused a growth in the necessary key length. The performance characteristics of RSA are observed by implementing the algorithms for computation. In this paper, RSA was implemented through an encryption and decryption procedures over different key sizes.

Keywords: RSA, Pubic Key, Private Key, Encryption, Decryption.

1. INTRODUCTION

Because of wide uses of networks during the last decade and growing security requirements in communication, public-key cryptosystems have been regarded highly. The concept of public-key cryptography evolved from an attempt to attack two of the most difficult problems associated with symmetric encryption [2]. A key distribution under symmetric encryption requires either (1) that two communicants already share a key, which somehow has been distributed to them; or (2) the use of a key distribution center. The Rivest- Shamir- Adelman (RSA) cryptosystem is a well known public-key encryption method that is applied to many systems for encryption and decryption [3]. RSA is one of the oldest and most widely used public key cryptographic systems. It was the first algorithm known to be suitable for signing as well as encryption, and one of the first great advances in public key cryptography [1]. RSA is still widely used in electronic commerce protocols, and is believed to be secure given sufficiently long keys. The RSA method is mainly based on integer and factoring as a one-way function [2]. Asymmetric cryptography: Unlike the symmetric cryptography, asymmetric cryptography uses a pair of keys to encrypt and decrypt message. One of these two keys is known as public key as it is distributed to others and the other is called private key which is kept secret [8]. Normally public key is used to encrypt any message which can only be decrypted by the corresponding private key. There are essential properties that must be satisfied by the asymmetric cryptography [4].

(*i*) The key generation process should be computationally efficient.

(*ii*) Sender should be able to compute the cipher text by using the public key of the receiver for any message.

(*iii*) The receiver should be able to decrypt the cipher easily to plain text by using his own private key.

(*iv*) It is impossible or at least impractical to compute the private key from the corresponding public key.

(*v*) It is computationally infeasible to compute the plain text form the public key and cipher text.

RSA is the most widely used asymmetric encryption system or a public key encryption standard, the private key is kept secret but the public key is revealed to everybody in RSA [1]. There is no doubt that RSA provides security for large values of prime numbers [5]. Section 2 contains brief description of RSA Algorithm which is followed by implementation of RSA in section 3. Section 4 defines the applications of Public key cryptosystems. Section 5 shows the performance of RSA and finally the paper is ended with the Conclusion at Section 6.

2. OVERVIEW OF RSA CRYPTOSYSTEM

A public-key encryption scheme has six ingredients Figure 1[4]:

- Plaintext: This is the readable message or data that is fed into the algorithm as input.
- Encryption algorithm: The encryption algorithm performs various transformations on the plaintext.
- Public and private keys: This is a pair of keys that have been selected so that if one is used for encryption, the other is used for decryption. The exact transformations performed by the algorithm depend on the public or private key that is provided as input.

[1,2] Lecturer, Department of CSE, N.C. College of Engg., Israna, Panipat.
[1]*E-mail: punita.meelu@gmail.com,* [2]*rajni.meelu@gmail.com*

- Ciphertext: This is the scrambled message produced as output. It depends on the plaintext and the key. For a given message, two different keys will produce two different cipher texts.
- Decryption algorithm: This algorithm accepts the ciphertext and the matching key and produces the original plaintext.

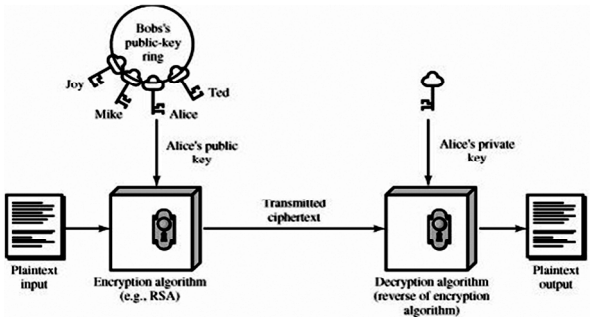

Figure 1: Public Key Encryption Scheme

In RSA, the plaintext and the cipher text are considered as integers between 0 and n-1, where n is the modulus. The typical size of n is 1024 bits. However, the recommended length of n is 2048 bits as 640 bits key is no more secure by now [2].

The RSA algorithm is comprised of three sub algorithms that are described below:

2.1 Key Generation Algorithm

RSA public and private key pair can be generated by the following procedure. Choose two random prime numbers p and q such that the bit length of p is approximately equal to the bit length of q.

The key set is generated by using the following algorithm:

1. Select two large prime numbers p and q such that $p \neq q$.
2. Compute modulus $n = p*q$
3. Compute $\varphi(n)$ such that $\varphi(n) = (p-1)*(q-1)$.
4. Choose a random integer e satisfying $1<e<\varphi(n)$ and $gcd(e, \varphi(n)) = 1$
5. Compute the integer d, such that $e*d = 1 \bmod \varphi(n)$. (n, e) is the public key, and (n, d) is the private key

- n is known as the modulus.
- e is known as the public exponent or encryption exponent or just the exponent.
- d is known as the secret exponent or decryption exponent.

2.2 Encryption

Encryption refers to algorithmic schemes that encode plain text into non-readable form or cipher text, providing privacy. Encryption is done by using the following steps:

1. Obtain the recipient's public key (n, e).
2. Represent the plaintext message as a positive integer m.
3. Compute the cipher text $c = m^e \bmod n$.
4. Send the cipher text c to receiver.

2.3 Decryption

Decryption refers to algorithmic schemes that decode cipher text or non-readable text into readable form or pain text. Message is decrypted by using the following steps:

1. Receiver uses his own private key (n, d) to compute $m = c^d \bmod n$.
2. Extracts the plaintext from the integer representative m.

3. IMPLEMENTATION OF RSA

This section includes a general technical introduction to the Eclipse Platform. The Eclipse Platform is designed for building integrated development environments (IDEs), and arbitrary tools. It presents a technical overview of its architecture and study of how the Eclipse Platform was used to build a full-featured Java development environment. Eclipse Software Development Kit (SDK) is both the leading Java integrated development environment (IDE) and the single best tool available for building products based on the Eclipse Platform [11]. The Eclipse SDK, a critical piece of the Eclipse tapestry, is a combination of the efforts of several Eclipse projects, including Platform [http://eclipse.org/platform], Java Development Tools (JDT [http://eclipse.org/jdt]), The Eclipse Platform contains the functionality required to build an IDE [13]. However, the Eclipse Platform is itself a composition of components; by using a subset of these components, it is possible to build arbitrary applications.

Eclipse Platform is more than just a foundation for building development environments. It is a foundation for building arbitrary tools and applications. One of the key benefits of the Eclipse Platform is realized by its use as an integration point. Building a tool or application on top of Eclipse Platform enables the tool or application to integrate with other tools and applications also written using the Eclipse Platform[12]. The Eclipse Platform is turned in a Java IDE by adding Java development components (e.g. the JDT) and it is turned into a C/C++ IDE by adding C/C++ development components (e.g. the CDT [http://eclipse.org/cdt])[13]. It becomes both a Java and C/C++ development

environment by adding both sets of components. Eclipse Platform integrates the individual tools into a single product providing a rich and consistent experience for its users [11]. Perhaps the most obvious thing that the Eclipse Platform provides is a managed windowing system. Although the Eclipse Platform is built on a mechanism for discovering, integrating, and running modules called plug-ins.

The following is a brief summary of features:

- Java projects
  - Java source (*.java) files arranged in traditional Java package directories below one or more source folders.
  - JAR libraries in the same project, another project, or external to the workspace.
  - Generated binary class (*.class) files arranged in package directories in a separate output folder.
- Browsing Java projects
  - In terms of Java-specific elements: packages, types, methods, and fields.
  - Arranged by package, or by supertype or subtype hierarchy.
- Editing
  - Java source code editor.
  - Keyword and syntax coloring (including inside Javadoc comments).
  - Separate outline shows declaration structure (automatic live updates while editing).
  - Compiler problems shown as annotations in the margin.
  - Declaration line ranges shown as annotations in the margin.
- Search
  - Find declarations of and/or references to packages, types, methods, and fields.
  - Search results presented in search results view.
  - Search results reported against Java elements.
- Compile
  - JCK-compliant Java compiler.
  - Compiler generates standard binary *.class files.
  - Incremental compilation.
  - Compiles triggered manually upon demand or automatically after each change to a source file (i.e., workspace auto-build).
  - Compiler problems presented in standard tasks view.
- Run
  - Run Java program in separate target Java virtual machine.
  - Supports multiple types of Java virtual machine (user selectable).
  - Console provides stdout, stdin, stderr.
  - Scrapbook pages for interactive Java code snippet evaluation.
- Debug
  - Debug Java program with JPDA-compliant Java virtual machine.
  - Set breakpoints and step through method source code.
  - Inspect and modify fields and local variables.
  - Expression evaluation in the context of a stack frame.

4. APPLICATIONS FOR PUBLIC-KEY CRYPTOSYSTEMS

Public-key systems are characterized by the use of a cryptographic algorithm with two keys, one held private and one available publicly [8]. Depending on the application, the sender uses either the sender's private key or the receiver's public key, or both, to perform some type of cryptographic function. In broad terms, we can classify the use of public-key cryptosystems into three categories:

(*i*) **Encryption/decryption**: The sender encrypts a message with the recipient's public key.

(*ii*) **Digital signature**: The sender "signs" a message with its private key. Signing is achieved by a cryptographic algorithm applied to the message or to a small block of data that is a function of the message.

(*iii*) **Key exchange**: Two sides cooperate to exchange a session key. Several different approaches are possible, involving the private key(s) of one or both parties.

5. RESULT

This paper presents the implementation of RSA through an encryption and decryption procedures, which are readily available for commercial use. Experiments were conducted on different text sizes. The results obtained in encryption and decryptions of RSA were given in seconds. The computing times are presented in tabular form in Table1.

Table 1
RSA Encryption and Decryption Execution Timings

Text Size	*Encryption*	*Decryption*
128 bits	0.0549	0.0549
256 bits	0.1098	0.1098
512 bits	0.2197	0.1648
1024 bits	0.3846	0.3296
2048 bits	0.7142	0.6593

6. CONCLUSION AND FUTURE SCOPE

This paper presents an encryption and decryption procedures of RSA over different text size i.e. 128 bits, 256 bits, 512 bits, 1024 bits and 2048 bits. The code can be used to encrypt a word file, picture file, binary file and a text file. It concludes that encryption requires more time as compared to decryption timing for the same text size over the same key as shown in Table 1 (for 512 bits, 1024 bits and 2048 bits) . As an asymmetric cryptosystem, RSA solves the problem inherent distributing the secret key [8]. The major drawback of RSA is its greater computational overhead due to its large key which must be improved so it can be use for the large key in future.

REFERENCES

[1] R.L. Rivest, A. Shamir, and L. Adleman. "A Method for Obtaining Digital Signatures and Public-key Cryptosystems", *Communications of the ACM*, **21(2)**, 120-126, February 1978.

[2] RSA Data Security, Inc. The RSA Factoring Challenge. http://www.rsa.com/rsalabs/node.asp?id=2092.

[3] Whitefield Diffie, Martin E Hellman "New Directions in Cryptography", *IEEE Information Theory*, June 23-25, 1975 and IEEE International Symposium on Information Theory, Sweden, June 21-24, 1976.

[4] William Stallings. Cryptography and Network Security Principles and Practices. Prentice Hall, November 16, 2005.

[5] RSA Laboratories, High Speed RSA Implementation, RSA Data Security, November 1994.

[6] M. Shand and J. Vuillemin, "Fast Implementation of RSA Cryptography", Proc.11th IEEE Symp. Computer Arithmetic, E.E. Swartzlander, M.J.

[7] Knuth DE. The Art of Computer Programming, **2** (Semi Numerical Algorithms), 2nd ed. Addison-Wesley: Massachusetts, 1980.

[8] Douglas R. Stinson, "Cryptography Theory and Practice", Chapman & Hall/CRC Press, 3rd Edition, pp. 211-214, 2006.

[9] Ajay C. Shantilal, "A Faster Hardware Implementation of RSA Algorithm", Irwin, and G.A. Jullien, eds., pp. 252-259, June 1993.

[10] E.F. Brickell, "A Survey of Hardware Implementation of RSA", Advances in Cryptology, Proc. CRYPTO '89, pp. 368-370, 1990.

[11] www.eclipse.org/downloads/moreinfo/jre.php.

[12] http://www.vogella.de/articles/EclipseRCP/article.html.

[13] http://www.zdnet.com/blog/burnette/view-live-references-with-eclipse-33m1-and-java-6-mustang/154.