Applied Hybrid Cryptography in Key-pair Generation of RSA implementation

Sankalp Prakash, Mridula Purohit

ABSTRACT

A key is a crucial factor in any cryptographic system. Devoid of a key, the algorithm would not produce any fruitful result. Key is being used in encryption, decryption and other cryptographic algorithms such as digital signature schemes and MAC (message authentication codes). The only thing that should be kept secret in a sound cryptosystem is the key so the key generation is important in implementation of cryptographic algorithms. In this paper the key generation module of the developed application package has been discussed which generates private and public keys in pairs using RSA and the corresponding private keys generated are encrypted using DES algorithm to apply hybrid cryptography.

KEYWORDS

Hybrid cryptography, Key-pair Generation, Public Key, Private Key, RSA, DES

INTRODUCTION

In this fast-paced technological world the importance of information and communication systems is escalating with the increasing significance and quantity of data that is transmitted. Unfortunately the vulnerability of systems and data are highly rising due to variety of threats, such as unauthorized access and use, destruction, alteration and misappropriation. Cryptography is the foundation of all data as well as information security aspects. Classical cryptosystems are quite simple to understand, implement and to be broken. Many new forms of cryptographic algorithms came after the extensive expansion of computer communications. Cryptography is being used as a tool in an information technology security environment to protect sensitive and high value data that is vulnerable to unauthorized disclosure or undetected modification during transmission or in storage.

In the present scenario the cryptographic techniques have become the immediate solution to protect information against third parties. These techniques required that data and information should be encrypted with some sort of mathematical algorithm where only the party that shares the information could possible decrypt to use the information [13]. Within the context of any communication, there are some specific security requirements includes (1) Authentication which means the process of providing one's identity; (2) Confidentiality that ensures no one can read the message except the intended receiver; (3) Integrity for assuring the receiver, the received message has not been altered in any way from the original and (4) Non-repudiation is a mechanism to prove that the sender really send this message. [14]

Cryptography itself splits into here main branches:

KEY GENERATION OF RSA IMPLEMENTATION

Key generation consists of the following two tasks:

(1) Symmetric (or Private-Key) Cryptography: two parties have an encryption and decryption method for which they share a secret key.

Data Encryption Standard (DES) is perhaps the most well-known and widely used symmetric key cryptosystem in the world. It is a symmetric block algorithm written by IBM that encodes 64-bit blocks of data using a 56-bit key.

(2) Asymmetric (or Public-Key) Cryptography: a user possesses a secret key (private key) as in symmetric cryptography but also a public key.

RSA, developed by Ronald L. Rivest, Adi Shamir and Leonard M. Adleman in 1977 at MIT [11], is still most widely used and is believed to be secure given sufficiently long keys and the use of up-to-date implementations. [17] It was a practical public-key cipher based on the difficulty of factoring very large numbers. The idea is that it is easy to find two large primes but it is difficult to factor the product of the two primes.

(3) Hybrid Cryptography: Symmetric and asymmetric ciphers each have their own advantages and disadvantages. Symmetric ciphers are significantly faster (Schneier states "at least 1000 times faster") than asymmetric ciphers, but require all parties to somehow share a secret (the key) [1]. The asymmetric algorithms allow public key infrastructures and key exchange systems, but at the cost of speed. So a hybrid cryptosystem is protocol to combination of the specific advantages of the two presently used encryption methods – speed (symmetrical encryption) and security (asymmetrical encryption). In other words symmetric and asymmetric algorithms (and often also hash functions) are all used together.

In designing security systems, it is wise to assume that the details of the cryptographic algorithm are already available to the attacker. This principle is known as Kerckhoffs' principle – "only secrecy of the key provides security", or, reformulated as Shannon's maxim, "the enemy knows the system". The history of cryptography provides evidence that it can be difficult to keep the details of a widely-used algorithm secret. [15]

In the real world, key management is the hardest part of cryptography. Keeping the keys secret is much harder. Cryptanalysts often attack both symmetric and public key cryptosystems through their key management. So the security of a cryptographic algorithm rests in the key. If someone using a cryptographically weak process to generate keys then the whole system is weak [1].

This journal is cited as : **JIMS 8i-Int'l J. of Inf. Comm. & Computing Technology(IJICCT)**

1. Determine two prime numbers, *p* and *q*,
2. Selecting e and calculating the *d*.

First consider the selection of *p* and *q* because the value $n = p \times q$ will be known to any potential opponent, to prevent discovery of *p* and *q* by exhaustive methods, these primes must be chosen from a sufficiently large set (i.e. p and q must be large numbers ~ 100 digits). But the method used for finding large primes must be reasonably efficient.

The following procedure is usually adopted for picking prime numbers:

1. Pick an odd integer at random using a pseudorandom number generator.
2. Pick an integer $a < n$ at random.
3. Perform the probabilistic primality test, such as Miller-Rabin. If *n* fails the test, reject *n* and go to step 1.
4. If *n* has passed a sufficient number of tests, accept *n;* otherwise go to step 2.

This is tedious procedure but it is performed only when a new key pair {KU, KR} in needed.

The traditional method of generating random numbers is to pass the results of a pseudorandom number generator through a hash function like MD5 or SHA and use the hash result. This is however much slower and instead faster approach was adopted in the development of this application. The developed application uses the shuffling method as outlined in Computer Generated Random Numbers [5]. Additionally it uses around 16 random entropy sources including different pseudorandom number generators. The basic method is outlines below:

Random Number Generation

- Start with an array of dimension around 200.
- Seed all the pseudorandom number generators by cascading one random number to seed to the next one. The first generator is seeded with system time.
- Fill the first 100 elements with the results of the first pseudorandom number generator.
- Fill the next 100 elements with the results of the second pseudorandom number generator.
- When the program wants a random number, randomly choose one from the array and send it to the program.
- Replace the number chosen in the array with a new random number from the main random number generator which uses all the 16 randomness entropy sources.

Primality Test

The random number values generated are made odd and then checked for primality using the probabilistic Miller-Rabin algorithm.

MILLER-RABIN (n, t)
INPUT: an odd number $n \geq 3$ and security parameter $t \geq 1$.
OUTPUT: an answer "PRIME" or "COMPOSITE" to the question: "Is n Prime?"

1. Write $n - 1 = 2^s r$ such that r is odd.
2. For I from 1 to t do the following
   - 2.1. Choose a random number integer a, $2 \leq a \leq n - 2$
   - 2.2. Compute $y = a^r \bmod n$
   - 2.3. If $y \neq 1$ and $y \neq n - 1$ then do the following:
     j ← 1.
     While $j \leq s - 1$ and $y \neq n - 1$ do the following:
     Compute $y \leftarrow y^2 \bmod n$
     If y = 1 then return ("COMPOSITE")
     $j \leftarrow j + 1$
     if $y \neq n - 1$ then return ("COMPOSITE")
3. Return ("PRIME").

Calculation of Private Key

Having determined prime numbers p and q, we select e and calculated using the Extended Eulid's algorithm:

EXTENDED_EULID(U,V)
INPUT: two nonnegative integers u and v.
OUTPUT: a vector (u1, u2, u3) such that uu1 + vv1 = u3 = gcd(u, v)

This journal is cited as : **JIMS 8i-Int'l J. of Inf. Comm. & Computing Technology(IJICCT)**

REMARK: Temporary vectors (v1, v2, v3) and (t1, t2, t3) are used in such a way that the following hold throughout the calculation:

ut1 + vt2 = t3 uu1 + vu2 = u3 uv1 + uv2 = v3

1. Initialize: set(u1, u2, u3) ← (1, 0, u) and (v1, v2, v3) ← (0, 1, v)
2. If v3 = 0 stop.
3. Set q ← ⌊u3/ v3⌋ and then set:
   (t1, t2, t3) ← (u1, u2, u3) – (v1, v2, v3)q
   (u1, u2, u3) ← (v1, v2, v3)
   (v1, v2, v3) ← (t1, t2, t3)
   Return to step2

THE HYBRID ENCRYPTION ALGORITHM

A hybrid encryption algorithm has the advantages of both the symmetric and asymmetric algorithms. This process involves the following steps:

(1) Generate key pair i.e. Public key KU= {e, n} value and Private Key KR= {d, n} using RSA key generation
(2) Save the Public Key {e, n}.
(3) Encrypt Private Key {d, n} by using DES algorithm of symmetric cryptography and then Save it in a file for better security.
(4) Now encrypt the message by using the Public Key KU.
(5) At the time of decryption, first the private key is decrypted and then the encrypted message can be decrypted using private key KR.

The complete process can be viewed in the figure 1.

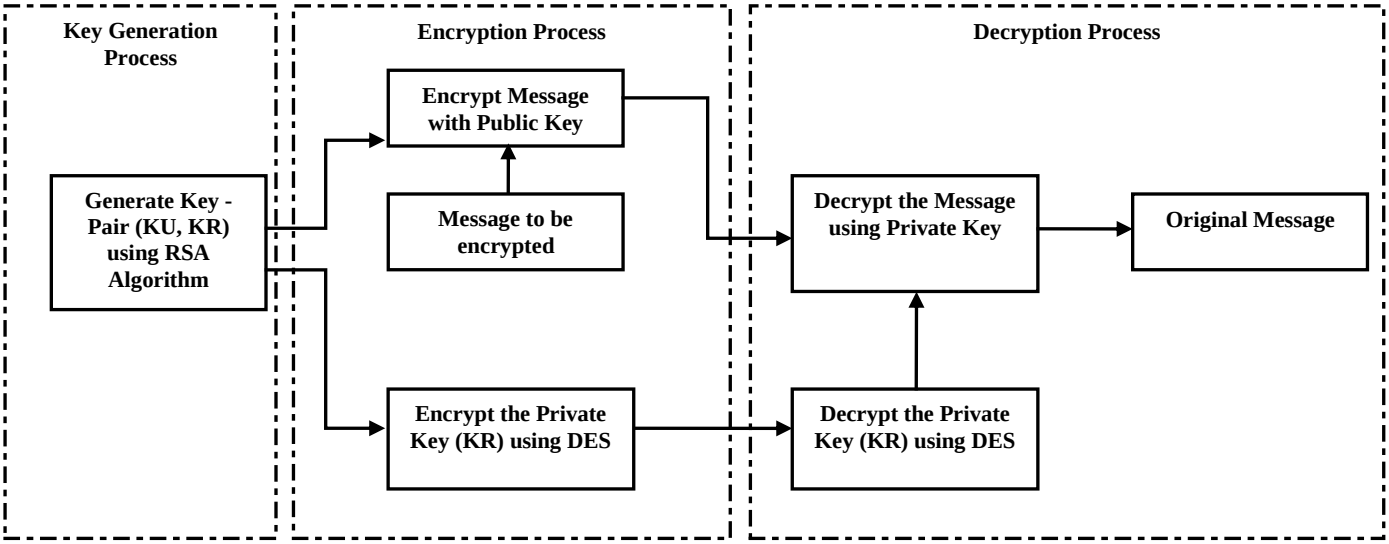

Figure 1: Hybrid Cryptography in RSAAPP

THE DEVELOPED APPLICATION (RSAAPP)

The key generation process generates the public and private keys in pairs. If required the keys can be viewed in hex format after generation. The corresponding private keys generated are encrypted using DES after taking 8-character password as user input. Both keys are made read only. The figures 2 to 5 below show the execution of the developed application i.e. RSAAPP-

This journal is cited as : **JIMS 8i-Int'l J. of Inf. Comm. & Computing Technology(IJICCT)**

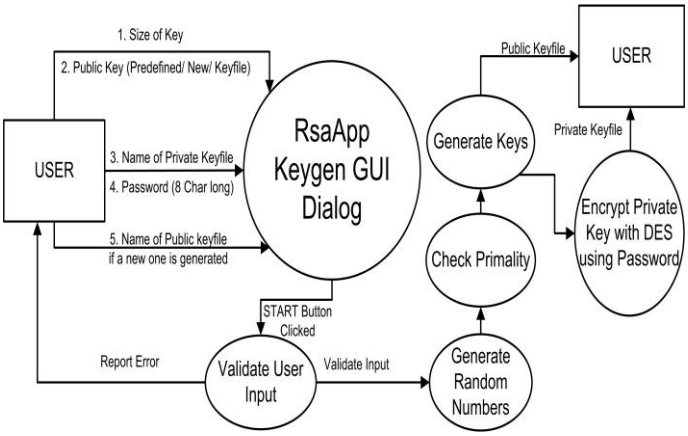

Figure 2: DFD for Key Generation Process in RSAAPP

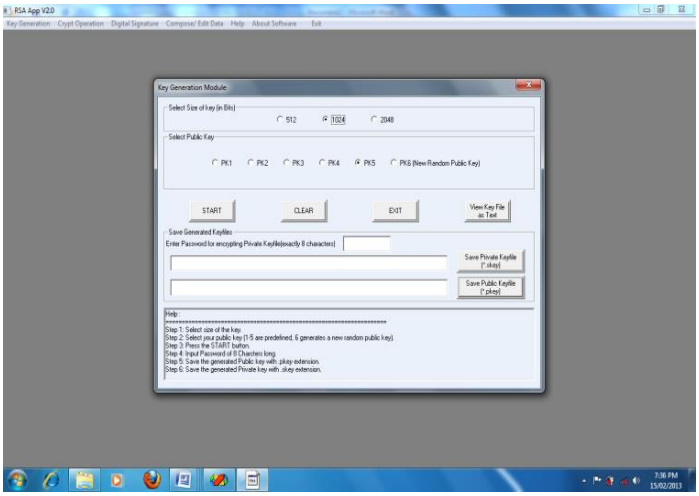

Figure 3: Key Generation User Interface

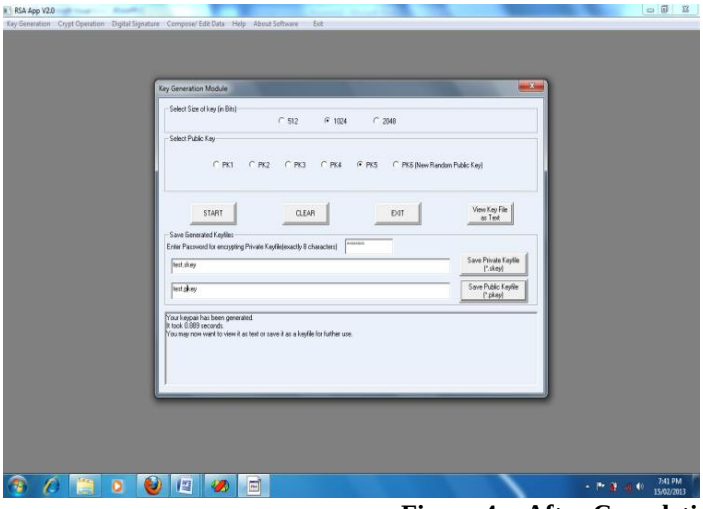

Figure 4 : After Completing Key Generation Process

This journal is cited as : **JIMS 8i-Int'l J. of Inf. Comm. & Computing Technology(IJICCT)**

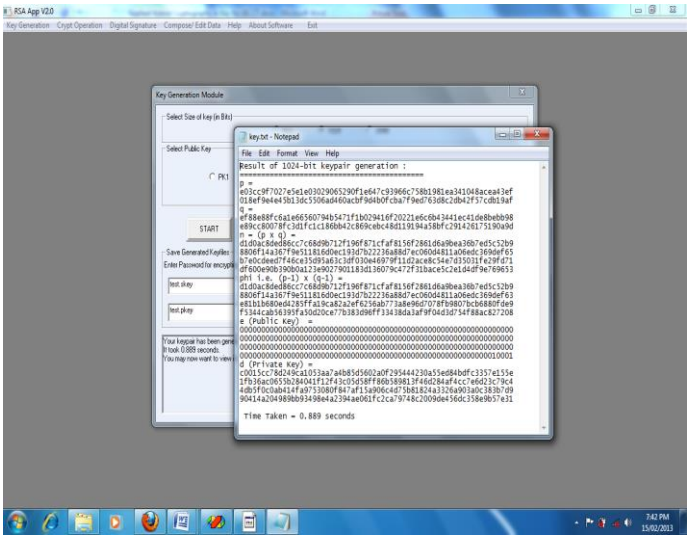

Figure 5: Result of the Key Generation

CONCLUSION

Soundness of Cryptosystem relies upon two basic factors: (1) algorithm and (2) key. The algorithm is a mathematical function, and the key is a parameter used by that function. A key is often easier to protect because it is a small piece of information, and easier to change if compromised. Thus, the security of an encryption system in most cases relies on some key being kept secret but practically it is very difficult. An attacker who obtains the key by theft, extortion, dumpster dives or social engineering can recover the original message from the encrypted data.

Cryptanalysts often attack both symmetric and public key cryptosystems through their key management. So the security of a cryptographic algorithm rests in the key. Therefore in the developed application the key generation process is made modular, efficient and fast enough so that it can generate the highly secured key-pairs in reasonable time and then the generated private key is being protected using DES which provides the advantages of hybrid cryptosystem in the RSAAPP.

REFERENCES

[1] Bruce Schneier, *Applied Cryptography: Protocols, Algorithms and Source code in C*, 2nd ed. John Wiley & Sons, New York, 1996.

[2] Alexander W. Dent, "Hybrid Cryptography," August 2004, information Security Group, Royal Holloway, University of London Egham Hill, Egham, Surrey, TW20 0EX,UK.

[3] Alfred Menezes, Paul C. van Oorschot and Scott A. Vanstone, *Handbook of Applied Cryptography*. CRC Press, Boca Raton, FL, 1996.

[4] ChristofPaar and Jan Pelzl, *Understanding Cryptography: A Textbook for Students and Practitioners*. London: Springer Monograph Series, 2009

[5] David W. Deley, "Computer Generated Random Numbers," 1991, at http://www.virtualschool.edu/mon/Crypto/RandomNumberMath.

[6] RSA Algorithm online at http://www.di-mgt.com.au/rsa_alg.html

[7] Dennis Hofheinz and EikeKiltz, "Secure Hybrid Encryption from Weakened Key Encapsulation," in *Advances in Cryptology - CRYPTO 2007*. Springer, 2007, pp. 553–571.

[8] L. Granboulan, "RSA hybrid encryption schemes," Cryptology ePrint Archive, Report 2001/110, 2001, at http://eprint.iacr.org/2001/110.ps

[9] K. Kurosawa and Y. Desmedt, "New Paradigm of Hybrid Encryption Scheme," in *Advances in Cryptology CRYPTO 2004*, ser. Lecture Notes in Computer Science, M. Franklin, Ed., vol. 4622. Springer-Verlag, 2004, pp. 426–442.

[10] MihirBellare and Phillip Rogaway, "Introduction to Modern Cryptography," in *UCSD CSE 207 Course Notes*, 2005, p. 207, http://www.cs.ucdavis.edu/rogaway/classes/227/spring05/book/main.pdf

[11] R. L. Rivest, A. Shamir, and L. M. Adleman, "A method for obtaining digital signatures and public key cryptosystems," *Comm. ACM*, vol. 21, issue 2, pp. 120–126, Feb. 1978.

[12] S. Subasree and N. K. Sakthivel, "Design of a New Security Protocol Using Hybrid Cryptography Algorithms," *International Journal of Research and Reviews in Applied Sciences*, vol. 2, no. 2, February 2010.

[13] Onwutalobi Anthony-Claret Department of Computer Science University of Wollongong "Using Encryption Technique"

This journal is cited as : **JIMS 8i-Int'l J. of Inf. Comm. & Computing Technology(IJICCT)**

[14] Whitfield Diffie & Martin E. Hellman "Privacy and Authentication: An Introduction to Cryptography", *proceedings of the IEEE*, vol.67, no.3, 1979.
[15] Key (cryptography) - Wikipedia, the free encyclopedia at http://en.wikipedia.org/wiki/Key_(cryptography)
[16] RSA - Wikipedia, the free encyclopedia at http://en.wikipedia.org/wiki/RSA.

This journal is cited as : **JIMS 8i-Int'l J. of Inf. Comm. & Computing Technology(IJICCT)**