Journal of the Nigerian Association of Mathematical Physics
Volume **13 (November, 2008), 289 - 296**

Comparative analysis of Rivest Shamir Adleman (RSA), Rabin and El Gamal public key cryptosystems

[1]M. S. Olajide,. [2]O. S. Adewale, [2]B. K. Alese and [2]A. O. Adetunmbi
[1]Department of Computer Science, Adeyemi College of Education, Ondo, Nigeria.
[2]Department of Computer Science, Federal University of Technology, Akure, Nigeria.

Abstract

Eavesdropping is as old as human race. The ever increasing dependence on the use of computer and the telecommunications technology (ICT) has heightened the desire to protect and preserve the integrity of networks. This takes the challenge to the level of security. This paper attempts to conduct a comparative analysis of the three public key cryptosystems (RSA, Rabin and El Gamal) that are thought to be secure. Each of the cryptographic schemes is appraised with a view to exposing their strengths and weaknesses. A platform is considered in the paper for the schemes based on some denominators for indepth comparisons.

***Keyword*:** Cryptosystem, Security, Eavesdropping, Networks

1.0 Introduction

The increasing rate of dependence of individuals, corporate organizations and governments on information probably accounts for why there is explosive growth in the demand for computers nowadays. The interconnections of these computers using telecommunications could be another reason (Olajide, 2006 [8]). The interconnections have evolved tools of different capabilities to enhancing the performance of these networks.

However, the revolutionary trend in communications has ushered in open networks, which are enjoyed world over. People with malicious intents abuse the non-restricted activities to explore the potentials on the given network systems for their personal aggrandizement. The security of open network becomes a hope concern. An effective solution to security communications over open networks is provided by cryptography (Wembo, 2004 [10]).

Cryptography is the field concerned with linguistic and mathematical techniques for securing information, particularly in communications. Cryptography relies upon two basic components: an algorithm (or Cryptographic methodology) and a key. In modern cryptographic system, algorithms are complex mathematical formulas and keys are strings of bits. For two parties to communicate, they must use the same algorithm (or algorithms that are designed to work together). In some cases, they must also use the same key. Many cryptographic keys must be kept secret; sometimes algorithms are also kept secret (NIST, 1991 [7]).

Eric *et al* (2005 [3]) described cryptography as a vast and complex subject in which a little knowledge about the field could be very helpful with respect to security. Cryptography is said to play an important role in an overall security and a potent tool with security scheme and a potent tool with mathematical proofs to back up the level of security.

Though, cryptography can be very secure when used properly, the human aspect is very essential. A good and tested cryptographic algorithm could have the security provided wide through human unpredictable nature.

In general, cryptography is best understood by breaking it into four main areas or primitives. These primitives include

- Random number generation
- Symmetric encryption

- Asymmetric encryption
- Hash functions.

Using these primitives, or building blocks, all areas of cryptography are constructed. Infact, some of the primitives are used to build other primitives. In this paper, only symmetric encryptions and asymmetric encryptions will be considered.

Encryption is similar to encoding in that the process transforms some original text or object into another formula. Symmetric encryption also known as secret key encryption uses a common key to scramble or unscramble a message. The common key is said to be a secret key. An example of this type of encryption is called Caeser Cipher (Cobb, 1996 [2]). In asymmetric encryption, two different keys, one of which is referred to as public and the other private are used to scramble and unscramble message(s).

George *at al* (2003 [4]) remark that in secret key cryptography, the complexity of the assumption that a common key is distributed securely among the parties involves is cumbersome. This is seen as a major drawback to the scheme. This draw back necessitated the introduction of the public key cryptosystems. In public key encryption, two keys are employed: one is used to encrypt and is known as public key and the other used to decrypt and is called private key. The private key can be published or made available using a prescribed scheme for parties involved.

The concept of public key cryptosystem was invested by White field Diffie and Martin Hellman, and independently by Ralph Merkle. Ralph Merkle and Martin Hellmann invented the first known as public key cryptosystem in 1974. It was based on puzzles that were easy to solve if you knew the private key, and hard if you did not. The security of the system was eroded due to some circumstances that part of the encryption could be reversed in a sequence of results spanning 10 years. A cryptosystem is a suite of protocols ciphers, key management and user prescribed actions implemented together as a system (Kessler, 2005 [5]).

Many algorithms have been proposed for public key cryptosystem. However, some of them have been found insecure, and other impractical, because the keys are too long or the cipher text is much longer than the plaintext. Still, others are good only for encryption, or only for signing. According to Schneier (1996 [9]), only three algorithms that are thought to be secure, practical, and good for both signing and encryption are RSA, Rabin and El Gamal..

In this paper, comparative analysis of RSA, Rabin and El Gamal cryptosystem is carried out. A study of each of the cryptosystems is conducted one after the other to enhance in-depth analysis. A review of each of the algorithms is presented in Sections 2, 3 and 4 respectively. Section 5 provides the common platform to address the comparative issues. Experimental results are presented in section 6 and some conclusions are drawn in section 7.

2.0 RSA cryptosystem

The RSA scheme was the first secure public key cryptosystem. The idea of public key cryptosystems was published a few years earlier by Diffie and Hellman. RSA was invented in 1978 by Ron Rivest, Adi Shamir and Leonard Adleman, and takes its name from their initials.

It is the most widely used public key cryptosystem and provides both secrecy and digital signatures. The scheme gets its security from the difficulty of factorizing large numbers. This means that the message to be coded first must be mapped onto one or more numbers. One way to do this is to take the bit pattern of the underlying electronic form of the message and interpret that bit pattern as more positive integer.

The plaintext form of the message is usually denoted M and the encryption version *C*. If the encryption is denoted as $E(M)$, and the decryption function as D(C), the RSA encryption and decryption functions can be written as

$$C = E(M) = M^e \bmod n \tag{2.1}$$

and

$$M = D(C) = C^d \bmod n \tag{2.2}$$

Hence both the encryption and the decryption function involve exponentiation mod *n*. The public key consists of the pair $\{e, n\}$ and the private key is the pair $\{d,n\}$. The numbers *e*,*d*, and n are all integer. Some requirements must be met for this system to work and these include:

- Applying the decryption function to the encrypted text must reproduce the original message

$$D(E(M)) = M \text{ or } M^{ed} \bmod n = M \tag{2.3}$$

This condition places restrictions on the possible values of *e* and *d*. If at all such values exist.

- The above condition implies that *e* and *d* might be mathematically related. Hence, it must be infeasible to determine the value *d* given *e* and *n*.

- Application of encryption and decryption functions should be relatively easy.

All these conditions can be met, the first one being the most mathematically involved. The following property, which follows Euler's theorem, gives a due to choose the numbers *e*,*d*, and *n*. If *n* is the product of two prime numbers, $n = pq$ and $0 < m < n$ then for any possible integer *K*.

$$M^{k\phi(n)+1} \bmod n = m \tag{2.4}$$

Here $\phi(n)$ is the Euler totient function which is the number of positive integers less than n and relatively prime to n. It follows that $\phi(n)$ is an integer such that $1 < \phi(n) < n$.

Equation 2.4 shows a function applied to the number m returns the same number m again. This means that if one chooses

$$ed = k\phi(n)+1 \tag{2.5}$$

The encryption and decryption functions (1) and (2) are each others inverse, as required. In summary, the RSA keys are generated as follows:

- Select two prime numbers *p* and *q*
- Calculate the modulus $n = p \times q$.
- Calculate $\phi(n) = (p - 1) \times (q - 1)$.
- Select an integer e, such that $1 < e < \phi(n)$ and $\gcd(\phi(n), e) = 1$.
- Calculate the modular inverse *d*, which is such that $e \times d \bmod \phi(n) = 1$ with the extended form of Euclid's algorithm (4).
- The public key consists of the pair $\{e,n\}$.
- The private key consists of the pair $\{d,n\}$

Calculation of *d*, the modular inverse of e modular $\phi(n)$, is very difficult because only *n* is publicly known, not $\phi(n)$. The calculation of $\phi(n)$ is easy with the aid of the prime factors *p* and *q* but these are not publicly known. This actually is the core of the RSA method, the calculations of $\phi(n)$ is very difficult without the prime factors of the argument *n*. The reason is that, without knowing the prime factors of *n*, the only way to calculate $\phi(n)$ is to use methods that are computationally as demanding as factoring *n*.

2.1 Selection of encryption exponent e and decryption d m RSA

The encryption exponent $e = 3$ commonly used in practice; in this case, it is necessary that neither $p - 1$ nor $q - 1$ be divisible by 3. This result in a very fast encryption. However, another encryption exponent used in practice is $e = 2^{16} + 1 = 65537$. This number has an advantage over $e = 3$ in that it resists copper smith attacks. Copper smith attack showed how n can be factored in polynomial time if the high order $k/4$ bits of *p* are known.

It may be desirable to select a small decryption exponent d in order to improve the efficiency of decryption

2.2 Attacks on RSA.

These are various attacks on RSA, which are indeed paramount to performance of the scheme. These include

- Forward Search Attack - If the message space is smaller predictable an adversary can decrypt such cipher text after encrypting all possible messages. Salting such message is a simple method to prevent such attacks.
- Small Encryption Exponent: If e = 3 is used in encryption of say three entitles whose modulo are n_1, n_2 and n_3, then A would send c_1, $= m^3 \bmod n_i$, for i = 1,2,3. The eavesdroper can easily attack the ciphertext using some mathematical assumptions. A pseudorandomly generated bitstring of appropriate length could be appended to the plaintext messages prior to encryption.
- Common modulus attack: if distinct encryption/decryption exponent pair (e_i, d_i) is distributed by a central authorities to each entries in the network, any entity in the network, any entity with knowledge of any pair can determine the decryption exponents of all entities in the network. An eavesdropper (any entity within the network) could recover any message within the network with high probability.
- Message concealing- It is possible in RSA to have a message sent without concealing. Such message are easily attacked. There exist in RSA other attacks such as cycling, multiplication and factoring.

2.3 The RSA Problem

Against Chosen plaintext attack, the security of RSA lies on the difficulty of computing the root of a ciphertext c modulo a composite integer n. This is the so-called RSA problem. This problem can be defined as this:

Input $N = pq$ with p,q prime numbers; e: an integer such that $\frac{\gcd(e, Cp-1)(q-1)=1}{\gcd(e, Cp-1)(q-1)=1}$; $c \in Z_N^*$

OUTPUT the unique integer $m \in Z_N^*$ satisfying $m^e \equiv c \frac{(\text{mod } N)}{(\text{mod } N)}$

2.4 The Integer factorization problem

The difficulty of the RSA problem depends, in twin, on the difficulty of the integer factorization problem. The definition of Integer Factorization Problem (IF Problem) is as this. If INPUT *N*: odd composite integer with at least two distinct prime factors

OUTPUT prime p such that *p\N*

In both problems, it is assumed that they are difficult under properly chosen parameters.

3.0 Rabin cryptography

Rabin developed a public-key cryptosystem based on the difficulty of computing a square root modulo composite integer. Rabin's work has a theoretic importance. It provided the first provable security for public-key cryptosystems. The security of the Rabin cryptosystem is exactly the intractability of the integer factorization problem (just as in RSA) (Wembo, 2004)

The Rabin cryptosystem is specified in the algorithm below:

Key Generation

- choose two random prime numbers p and q such that $|p| \approx |q|$
- compute $N = pq$
- pick a random integer $b \in_u ¢^*_N$
- publicize (N,b) as public key and (p,q) as private key.

Encryption

To send a message $m \in ¢^*_N$ to a recipient in form of ciphertext. It is represented thus

$$c \leftarrow m(m+b)(\text{mod} N) \text{ for } m < N.$$

The decryption computation involves computing square roots modulo *N*. From the study of the square rooting problem, the difficulty of this problem is computationally equivalent to that of factoring N. Predominnatly, there are four square roots generated from above. One of these square roots mod *n* is the original plaintext *m*.

3.1 Security of Rabin public key cryptosystem

The task faced by a passive adversary is to recover plaintext m from the corresponding ciphertext c. This is precisely the square root problem associated with Rabin algorithm. Hence, assuming that factoring n is computationally intractable, the Rabin public-key encryption is provably secure against a passive adversary.

Moreover, even though Rabin Scheme is provably secure against a passive adversary, the scheme succumbs to a chosen-ciphertext attack. It is also susceptible to the following attacks: small encryption exponent e, forward search, multiplicative properties.

3.2 Redundancy

A drawback of Rabin's public key scheme is that the receiver is faced with the task of selecting the correct plaintext from among possibilities. This ambiguity in decryption can easily be overcome in practice by adding pre-specified redundancy to the original plaintext prior to encryption. For instance, the last 64 bits of the message may be replicated. Then, with high probability, exactly one of the four square roots m_1, m_2, m_3 and m_4 of a legitimate ciphertext c will possess this redundancy. The receiver will select this as the plaintext and if none of the square roots of c possesses this redundancy, then, the receiver should reject c as fraudlent. Thus, redundancy is used to eliminate chosen-ciphertext attack.

4.0 El Gamal cryptosystem

The El Gamal cryptosystem is an asymmetric key encryption, which is based on Diffie-Hellman key agreement scheme. The scheme was invented by an Egyptian cryptographer/Taher ElGamal in 1984. The security of El Ganal lies in the computational difficulty of discrete logarithm problems (Mrezes *et al*, 1996).

The ElGamal cryptosystem is specified in the algorithm below:

Key Generation

- choose a random prime number p'
- compute a random multiplicative generator element g of F_p^*;
- pick a random number $x \in_u ¢_{p-1}$ as private key;
- compute the public-key by $y \leftarrow g^x \pmod{p}$;
- publicize (p,g,y) as the public-key, and keep x as the private key.

(* similar to the case of the Differ-Hellman key exchange protocol, a system-wide users may share the common public parameters (p,q).*)

Encryption

To send a plaintext message m < p to a receiver, the sender picks $k \in_u ¢_{p-1}$ and computes ciphertext pair (C_1, C_2) as follows:

$$\begin{cases} c_1 \leftarrow g^k \pmod{p} \\ c_2 \leftarrow y^k m \pmod{p} \end{cases}$$

Decryption

To decrypt ciphertext (c_1, c_2), the receiver computes $m \leftarrow c_2 / c_1^x \pmod{p}$. The decryption calculation does restore the plaintext *m*. Since

$$c_1^x \equiv (g^k)^x \equiv (g^x)^k \equiv y^k \equiv c_2 / m \pmod{p}.$$

The division in the decryption step needs to use extended Euclid algorithm which is generally more costly than multiplication. However, the receiver may avoid the division by computing

$$m \leftarrow c_2 c_1^{-x} \pmod{p}.$$

One may verify that this decryption method works, but notice that –x here means p-1-x. all entries may elect to use the same p and generator y, in which case p and y need not be published as part of the public key. This results in public keys of smaller sizes. An additional advantage of having a fixed base y is that exponentiation can be expedited via pre-computations. A potential disadvantage of common system-wide parameters is that large moduli p may be warranted.

4.1 Efficiency of El Gamal

The encryption process requires two modular exponentiation, namely $g^k \pmod{p}$ and $(y^x)^k \bmod p$. These exponentiations can be speed up by selecting random exponents *k* having some additional structure; for example, exponents having low hamming weights. However, care must be taken that the number of exponents is large enough to preclude a search via a baby-step giant step algorithm.

One shortcoming of ElGamal encryption is that there is message expansion by a factor of 2. That is ciphertext is twice as long as the corresponding plaintext. Randomization of encryption process in ElGamal. ElGamal encryption is one of many encryption schemes which utilizes randomization in the encryption process. Others include McEliece encryption and Goldwassel-Micali encryption, and Blum-Goldwassel probabilistic encryption. Deterministic encryption scheme such as RSA may also employ randomization in order to circumvent some attacks.

The fundamental idea behind randomized encryption techniques is to use randomization to increase the cryptographic security of an encryption process through one or more of the following methods:

- Increasing the effective size of the plaintext message space;
- Precluding or decreasing the effectiveness of chosen-plaintext attacks by virtue of a one-to-many mapping of plaintext to ciphertext; and
- Precluding or decreasing the effectiveness of statistical attacks by leveling the a priori probability distribution of inputs.

5.0 Comparative analysis

Having considered the three cryptosystems in section 2 one after the other, it becomes expedient to carry out their comparative analysis. The analysis employs some features that can be evaluated based on the functionality and strength of each of the cryptosystems. Thus, a platform is therefore created.

5.1.1 Comparative elements of public-key cryptosystems

There are some common denominators in public-key cryptosystems which can conveniently be used for comparisons. These include

- Mathematical Background
- Key Generation
- Encryption
- Decryption
- Implementation factors
    - Bandwidth
    - Speed
    - Size
- Security
- Modification of plaintext

5.1.2 Mathematical background

The RSA cryptosystem derives its strength and security on the intractability of the integer factorization problem. The difficulty has made the scheme strong enough to withstand decades of cryptanalysis. Rabin cryptosystem has the advantage that the problem on which it is based is provably as hard as integer factorization problem. Thus, the security of Rabin Scheme relies on the intractability of integer factorization problem. The ElGamal public key has its strength and security on the intractability of the discrete logarithm problem.

5.1.3 Key generation

All asymmetric cryptosystems use both public-key and private key. The public key is necessary for encoding plaintext (message), which can be published, while only the recipient of the message must posses the private key. In Rabin cryptosystem, the public key and private key are (N,b) and (p,q) respectively. In RSA, the public key and private key are (e,N) and (d,N) respectively. The public key and private key in ElGamal cryptosystem are p,g,y and x respectively.

5.1.4 Encryption

For the encryption of Rabin cryptosystem, the public key is used to produce a ciphertext out of the plaintext. The ciphertext c is determined by $c = m^2 \bmod n$. To encrypt message m in ElGamal cryptosystem, it is required to invent some random number k and set the ciphertext c to be a pair, the ciphertext is represented thus: $c = (g^k, m.y^k)$. Notice that you can encrypt the same message in many different ways by choosing different k's.

In case of RSA, to encrypt a message m to a ciphertext, the resulting equation is $c = m^e \bmod(n)$.

5.1.4 Decryption

To decode the ciphertext, the private key is necessary. Rabin cryptosystem decryption is represented by the quadratic equation $m^2 + bm - c \equiv 0(\bmod N)$. This involves computing square roots modulo N. Four square roots are generated. In RSA, the decryption process is represented by $D(E(M)) = M$ or $M^{ed} \bmod n = m$. In ElGamal, to decrypt ciphertext (c_1,c_2) translate to $m \leftarrow c_2 / c_1 \, x(\bmod p)$.

5.1.5 Implementation patterns

The message expansion by factor of 2 in ElGamal cryptosystem is responsible for the consumption of larger bandwidth. Thus, more memory space was utilized by the ciphertext. The implementation time was 3 seconds. Also, in Rabin cryptosystem, sizeable bandwidth is consumed due to the elongation of the message by the addition of redundancy and the generation of four square roots. The implementation time for Rabin is 0.9 second while that of RSA is 1.5 seconds. The RSA cryptosystem makes use of fast modular multiplication, fast modular exponentiation and Chinese remainder and consumes very moderate memory space.

5.1.6 Security

The security of the RSA cryptosystem is based on two mathematical problems namely the problem of factoring very large number and the RSA problem. The RSA problem is defined as the task of taking nth roots modulo a composite n; recovering a value in such that $m^e = c \bmod n.$ RSA cryptosystem is faced with the following attacks:

- Forward search attack
- Cyclic attacks
- Common modulus attack
- Attacks resulting from small encryption exponent e
- Attacks resulting from small decryption exponent d
- Adaptive chosen-ciphertext attack

The great advantage of the Rabin cryptosystem is that the code can only be broken if the code breaker is capable of efficiently factoring the public key. It has been proven that decoding the Rabin cryptosystem is equivalent to the factorization problem, unlike in RSA and will remain so until a general solution for the factorization problem is discovered. An eavesdropping would have no chance today of break the code.
Rabin cryptosystem is susceptible to attacks similar to those on RSA and they include

- attack resulting from small encryption exponent e
- forward search attack
- adaptive chosen-ciphertext attack

The problem of breaking the ElGamal cryptosystem is equivalent to solving the Diffie-Hellman problem. The security of ElGamal cryptosystem is based on the discrete logarithm problem. The ElGamal cryptosystem utilizes randomization in the encryption process. The idea behind randomization is to increase the cryptographic security of an encryption process through one or more of the following methods.

- Increasing the effective size of the plaintext message space;
- Precluding or decreasing the effectiveness of chosen-plaintext attacks by virtue of a one-to-many mapping of plaintext to ciphertext;
- Precluding or decreasing the effectiveness of statistical attacks by leveling the a priori probability distribution of inputs.

RSA may also employ randomization in order to circumvent some attacks mentioned above.

5.1.7 Modification of plaintext message

The concept of salting of message is a vital phenomenon in RSA. Salting is simply a procedure, which involves appending a generated pseudorandom bit string of appropriate length to a plaintext to avert attack. Small encryption exponent sometimes necessitate salting. Also, if a message space is small, salting is also paramount. Salting the plaintext message in Rabin and RSA cryptosystems circumvent various attacks.

However, aside from salting, the addition of redundancy prior to encryption is another means of averting attacks in Rabin encryption. Redundancy is a process which involves replicating a portion of the original message longer before encryption. ElGamal utilizes randomization in the encryption process.

6.0 Experimental setup and results

C++ programming language was used for the implementation of the three algorithms on Pentium IV 1.5GHz processor. The three algorithms were subjected to the same plaintext message "The *vacation is approaching and I look forward to traveling home to see my relations. I received a call two days from my dad to come home for my holiday. Seeing you then. Have a nice time. …"*

From our experiment, Rabin codes took 0.9 second to generate ciphertext while RSA and ElGamal consume the largest memory space and bandwidth because the ciphertext size is twice the plaintext size. Rabin consumes a little more memory space than RSA. This is as a result of the four different ciphertext generated by Rabin. ElGamal algorithm implementation was much more complex than RSA and Rabin because of the pre-computation required for generating the public key and the complexity of discrete logarithm problems.

Rabin encryption is an extremely fast operation as it only involves a single modular squaring. By comparison, RSA encryption with e =3 takes one modular squaring. Rabin decryption is slower than encryption, but comparable in speed to RSA decryption. Finally, ElGamal required two modular exponentiations for encryption.

7.0 Conclusion

The need for effective security on global networks cannot be over-emphasised particularly now that many security solutions come both in hardware and software modes. The choice of the right cryptosystem with outstanding features will assist the security designers to enhance their activities. Rabin cryptosystem has features that could stimulate the security designers to beckon to its direction.

However, the evaluation of the three cryptosystems has yielded significant results. From the study, some inherent features of the three schemes have adequately been exposed side by side. It is a known fact that there are different users of the global networks. Each user has its own interest. The security designer should consider the different interests to designing their network security solution tools. It may be desirable to design tools which employ ElGamal for the use of the military and diplomatic corps to enhance the concealment of message to ultimate level. Rabin could be deployed to design tools, which have the propensity for solving e-commerce problems. Dairist, lovers and academic may settle for devices that employed RSA. In our future research work, we intend to look at cryptanalysis of the three cryptosystems in details.

References

[1] Alese, B.K. (2000), Vulnerability Analysis of Encryption/Decryption Techniques of Computer Network Security, M.Tech Thesis, Federal University of Technology, Akure, Nigeria.

[2] Cobb, S. (1996) PC and LAN Security, McGraw-Hill Publishing Company.

[3] Eric, C, Ronald, K. and James, W.C. (2005), Network Security Bible, Wiley Publishing Inc, Indianapolis, Indiana.

[4] George, S., James, X.D., Alan, G.B., Barbara, J.M., Alan, S. (2003), Information Technology Security, World Bank Publishing, Washington.

[5] Kessler, G.C. (2005), An overview of Cryptographic, Auerbach Publishing.

[6] Menezes, A, Van oorschorchot, P.C. and Vanstone, S.A. (1996), Handbook of Applied Cryptography, CRC Press.

[7] NIST (1991), National Institute of Standards and Technology, An Introduction to Computer Security, Special Publication.

[8] Olajide, M.S. (2006), Comparative Analysis of Rivest Shamir Adleman (RSA), Rabin and El Gamal Public key Cryptosystems, M.Tech Thesis, Federal University of Technology, Akure, Nigeria.

[9] Schneier, B (1996) Applied Cryptography, J.Wiley and Sons.

[10] Wembo, M. (2004), Modern Cryptography: Theory and Practice, Pearson Education Inc. Publishing