International Journal of Modern Computer Science (IJMCS)
Volume 4, Issue 6, December, 2016
ISSN: 2320-7868 (Online)

A Comparative Analysis of Public Key Cryptography

Ajit Karki
Assistant Professor
Deptt. of Information Technology.
The ICFAI University, Sikkim, India
Ranka Road Sichey, Gangtok, Sikkim.737101.
ajitkarki4@gmail.com

Abstract: In the globalization era, cryptography becomes more popular and powerful; in fact it is very important in many areas (i.e. mathematics, computer science, networks, etc). Cryptography is one of the main constituent of computer security. To meet a user's needs cryptographic algorithm needs to be selected on the basis of attributes like security and performance. Cryptography is one such way to make sure that confidentiality, authentication, integrity, availability and identification of user data can be maintained as well as security and privacy of data can be provided to the user. Encryption is the process of converting normal data or plaintext to something incomprehensible or cipher-text by applying mathematical transformations or formulae. These mathematical transformations or formulae used for encryption processes are called algorithms. Cryptography systems can be broadly classified into two categories: Symmetric encryption algorithms and Asymmetric encryption algorithms RSA and ECC are asymmetric key cryptographic algorithms. This paper provides an overview and comparison between the RSA cryptosystem and elliptic curve cryptography, which both focus on sending and receiving messages. RSA is the most popular public-key cryptosystem today but long- term trends such as the proliferation of smaller, simpler devices and increasing security needs will make continued reliance on RSA more challenging over time. Hence Elliptic Curve Cryptography (ECC) is a suitable alternative. Here, the algorithms studied and compared are RSA and ECC. The RSA Cryptosystem and elliptic curve cryptography theories are quite similar but elliptic curve cryptography is more complicated.

Keywords: Cryptography, authentication, RSA, ECC, asymmetric-key.

I. INTRODUCTION

Cryptography is an art of converting a message into an encoded unreadable form so that only the intended receiver can read and process it. The main aim of the cryptography is to provide security to the data from unauthorized access. The cryptography involves encryption and decryption. The transformation of original message into the format that is almost impossible to read without the appropriate knowledge is called encryption. Decryption is the reverse process of encryption. The encryption and decryption both require the use of some secret information i.e. key. Cryptography is one of the main constituent of computer security. To meet a user's needs cryptographic algorithm needs to be selected on the basis of attributes like security and performance. Cryptography is one such way to make sure that confidentiality, authentication, integrity, availability and identification of user data can be maintained as well as security and privacy of data can be provided to the user. Encryption is the process of converting normal data or plaintext to something incomprehensible or cipher-text by applying mathematical transformations or formulae. These mathematical transformations or formulae used for encryption processes are called algorithms. Cryptography systems can be broadly classified into two categories: Symmetric encryption algorithms and Asymmetric encryption algorithms RSA and ECC are asymmetric key cryptographic algorithms. This paper provides an overview and comparison between the RSA cryptosystem and elliptic curve cryptography, which both focus on sending and receiving messages. RSA is the most popular public-key cryptosystem today but long- term trends such as the proliferation of smaller, simpler devices and increasing security needs will make continued reliance on RSA more challenging over time. Hence Elliptic Curve Cryptography (ECC) is a suitable alternative. Here, the algorithms studied and compared are RSA and ECC. The RSA cryptosystem and elliptic curve cryptography theories are quite similar but elliptic curve cryptography is more Complicated. The idea of the RSA cryptosystem is based on three popular theorems which are Euler's Theorem, Fermat's Little Theorem and the Chinese Remainder Theorem. The reliability and strong security of the RSA cryptosystem depends on the degree of difficulty of integer factorization. In addition, the security of elliptic curve cryptography depends on the apparent difficulty of solving the elliptic curve discrete logarithm problem (ECDLP).

1. Objectives of cryptography:

There are four main objectives of cryptography:-

a). Confidentiality: It guarantees that the sensitive information can only be accessed by those users/entities authorized to unveil it.

b). Data integrity: It is a service which addresses the unauthorized alteration of data. This property refers to data that has not been changed, destroyed, or lost in a malicious or accidental manner.

c). Authentication: It is a service related to identification. This function applies to both entities and information itself. Two parties entering into a communication should identify each other.

d). Non-repudiation: It is a service which prevents an entity from denying previous commitments or actions.

The solution lies in Cryptosystems. Cryptosystem is basically a combination of an Encrypting system, and a Decrypting system. Known under different technical names, Cryptosystems have been in use all over the world for centuries. Only that with the advent of modern day 'hackers', it has become necessary to further develop forms of cryptosystems which are more difficult to crack. One such popular technique happens to be RSA algorithm. Also, there

are Elliptic curves which boast of a rich history of over hundred years.

II. CRYPTOGRAPHIC ALGORITHMS

Cryptographic algorithms have evolved over time to answer various needs. Here, we take a look at some of them to understand why Elliptic Curve Cryptography becomes all that important.

Modern cryptography is heavily based on mathematical theory and computer science practice; cryptographic algorithms are designed around computational hardness assumptions, making such algorithms hard to break in practice by any adversary. It is theoretically possible to break such a system, but it is infeasible to do so by any known practical means. These schemes are therefore termed computationally secure; theoretical advances, e.g., improvements in integer factorization algorithms, and faster computing technology require these solutions to be continually adapted. There exist information-theoretically secure schemes that provably cannot be broken even with unlimited computing power—an example is the one-time pad—but these schemes are more difficult to implement than the best theoretically breakable but computationally secure mechanisms.

III. TYPES OF CRYPTOGRAPHY

Cryptography involves two main approaches:

- Symmetric-key cryptography.
- Asymmetric-key cryptography.

Symmetric-key cryptography: Same secret key is used for both encryption and decryption.

Asymmetric-key cryptography: Two different keys are used i.e. one for encryption and other for decryption.

3.1.1 SYMMETRIC KEY CRYPTOGRAPHY is also known as single-key, secret-key, and private key or one-key encryption. In this technique sender and recover share same key for encryption and decryption process. This technique was one of the simplest and earliest. It used the concept of a common key. The key was supposed to be some secret info shared by the sender and the receiver. Symmetric key algorithm is divided into two parts: first one is **BLOCK CIPHER** which is used for blocks of data. In this technique data is divided into blocks and then these blocks is used for encryption and decryption. Example of block cipher is AES, triple DES which is popular techniques of symmetric algorithms. And second one is **STREAM CIPHER** which operates on a single bit at a time. Transmitting the secret key on insecure network is also a curse of destroy the secrecy. There are many advantages of symmetric key cryptography like Symmetric key encryption is much faster. Single-key encryption does not require a lot of computer resources when compared to public key encryption. A different secret key is used for communication with every different party. If a key is compromised, only the messages between a particular pair of sender and receiver are affected. Communications with other people are still secure.

3.1.2 Advantages and Disadvantages of Symmetric Algorithms:

The one of the main **advantage** of Symmetric Algorithms is that they are undoubtedly quite simple and easy to implement. However, the same properties make them quite vulnerable to attacks.

The **disadvantages of Symmetric Algorithms** are:

- Once the key is found, the attacker can easily decode and destroy any of the information at will.
- Scalability is also an issue, as the number of keys required as compared to the number of participants in the message exchange equals about the square of the number of participants.
- Also, symmetric algorithms cannot be used for Digital Signatures.[1].

3.1.3 ASYMMETRIC KEY CRYPTOGRAPHY is also known as the public key cryptography. There are two types of key first one is public key which is used for encryption and second is private key which is used for decryption. Only a particular user/device knows the private key whereas the public key is distributed to all users/devices taking part in the communication. The major drawbacks of asymmetric ciphers are their speed and security strength; they are much slower than the symmetric algorithms and more vulnerable to intruder attacks but they make key exchange easier. Asymmetric popular ciphers RSA (Rivest,Shamir,Adleman), Elliptic curve, Diffie-hellman key exchange algorithm, Digital signature. Advantages of asymmetric key algorithm are it solves the problem of distributing the key for encryption. Everyone publishes their public keys and private keys are kept secret. Public key encryption allows the use of digital signatures which enables the recipient of a message to verify that the message is truly from a particular sender. The use of digital signatures in public key encryption allows the receiver to detect if the message was altered in transit. A digitally signed message cannot be modified without invalidating the signature.

3.1.4 Advantages and disadvantages of Public Key Cryptography

Some advantages of Public Key Cryptography are:

- Key agreement is no issue at all here.
- Scalability, too, is not an issue.
- The techniques such as Digital Signatures used in PKC can also satisfy problems of Non-Repudiation and Authenticity.

Some disadvantages of PKC are:

- It is slower in comparison to Symmetric Algorithms.
- The size of the resultant encrypted text too, turns out to be larger than the original text.

IV. RSA ALGORITHM

RSA is a cryptosystem, which is known as one of the first practicable public-key cryptosystems and is widely used for secure data transmission. In such a cryptosystem, the encryption key is public and differs from the decryption key which is kept secret. In RSA, this asymmetry is based on the practical difficulty of factoring the product of two large prime numbers, i.e. on the factoring problem. RSA stands for Ron Rivest, Adi Shamir and Leonard Adleman, who first publicly described the algorithm in 1977[2].

4.1 Advantages and Disadvantages of RSA algorithm

Some advantages of RSA:

- The primary advantage is increased security.
- The use of Digital Signature makes it safe from repudiation.
- It may be used with Secret Key Cryptography.

4.1.1Problems with RSA algorithm

- Key generation is very slow.
- Speed of encrypting of text is slow.
- Message length should be less than the bit length otherwise algorithm will be fail.
- RSA is factorization based algorithm so that every time RSA initialization takes two large prime number **p** and **q**.
- If private keys of users are not available, it is vulnerable to impersonation.

4.1.2 RSA Key generation Encryption and Decryption

RSA involves a public key and a private key. The public key can be known by everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted in a reasonable amount of time using the private key. The keys for the RSA algorithm are generated in the following way

1. Choose two distinct prime numbers **p** and **q**.

2. Compute **n** = **p*q**.

3. Select the public key (i.e. the encryption key) **e** such that it is not factor of (**p-1**) and (**q-1**)

4. Select the public key (i.e. the decryption key) **d** such that the following equation is true.
(**d*e**) mod (**p-1**)*(**q-1**) = **1**.

2. Public key is pair of {**n, e**}
3. Private Key is pair of {**n, d**}

7. For encryption calculate the cipher text **CT** from the plain text **PT** as follows
CT=$\mathbf{PT}^{e}$ mod **n**

8. Send **CT** as the cipher text to the receiver.

9. For decryption, calculate the plain text **PT** from the cipher text **CT** as follows.

$$\mathbf{CT}^{d} \textbf{ mod } \mathbf{n}$$

4.1.3 RSA Encryption and decryption

Encryption uses the public key, so the cipher text corresponding to plain text m is

$$\mathbf{c = m^{e} (mod\ n)}$$

Decryption uses the corresponding private key, so

$$\mathbf{m=C^{d}\ (mod\ n)}$$

In simple terms signature generation is equivalent to decryption and signature verification is same as encryption.

V. ELLIPTIC CURVE CRYPTOGRAPHY

Elliptic Curve Cryptography (ECC) was discovered in 1985 by Victor Miller (IBM) and Neil Koblitz as an alternative mechanism for implementing public-key cryptography. Public-key algorithms create a mechanism for sharing keys among large numbers of participants or entities in a complex information system. ECC is based on discrete logarithms that are much more difficult to challenge at equivalent key lengths. The security of a public key system using elliptic curves is based on difficulty of computing discrete algorithms in the group of points on an elliptic curve defined over a finite field. Elliptic curve equation over a finite field Fp is

$$\mathbf{Y^2 = X^3 + ax + b \quad mod\ P}$$

Here, **y**, **x**, **a** and **b** are all within **Fp**, and p is **a** integers modulo **p**. **a** and **b** is the coefficients which determine what points will be on the curve. Curve coefficients have to fulfill one condition that is:

$$\mathbf{4a^3 + 27b^2 \neq 0}$$

This condition guarantees that the curve will not contain any singularities.

5.1. Point Representation

Representing a point on the curve is done in affine projection. Points which are represented in affine coordinates are vectors with an x and y component. Here **x** and **y** values are also integers modulo **p**. Point Operation: There are two basic operation in elliptic curve Point Addition and Point Doubling.

- **POINT ADDITION:** Adding two points is not easy in curve, adding their x- and y-components and taking them modulo p. connecting the two points via a line and then intersecting that line using curve. Point addition is works only two points which are not same.
- **POINT DOUBLING**: Point doubling comes into play if two points shall be added which are identical, i.e.

R = P + P

These are the calculations needed to get R. Note that a, which is needed for calculating s, is one of the curve parameters:

$$\mathbf{R_y = S.\ (P_x - r_x)-P_y}$$

Elliptic curve cryptosystem parameters are:

- **P**: The prime number which defines the field and curve operate on finite field **Fp**.
- **a** and **b** are two coefficients which define the curve.
- **G**: The generator or base point. It has two separate integers $\mathbf{g_x}$ and $\mathbf{g_y}$.
- **n**: The order of the curve generator point **G**.
- **h**: The cofactor of the curve. It is the quotient of the number of curve-points.

5.2 Generate a key in elliptic curve:

To get the private key, choose a random integer $\mathbf{d_A}$, so that

$$\mathbf{0 < d_A < n}$$

Then getting the accompanying public key $\mathbf{Q_A}$ is equally trivial, you just have to use scalar point multiplication of the private key with the generator point **G**:

$$\mathbf{Q_A = d_A \,.\, G}$$

In the elliptic curve public and private key are not equally exchangeable the private key dA is a integer, but the public key $\mathbf{Q_A}$ is a point on the curve.

- **Encryption** We want to encrypt data with the public key $\mathbf{Q_A}$ that we just generated. Again, first choose a random number **r** so that,

0 < r <n

Then, calculate the appropriate point **R** by multiplying **r** with the generator point of the curve:

R= r. G

Also multiply the secret random number **r** with the public key point of the recipient of the message:

S= r.Q_A

Now, **R** is publicly transmitted with the message and from the point **S** a symmetric key is derived with which the message is encrypted.

- **Decryption** Now, we receive a message which is encrypted with a symmetric key. With the message we receive a value of **R** in plain text.

S= d_A. R

By just multiplying your private key with the publicly transmitted point R, we will receive the shared secret point S, from which we can then derive the symmetric key. Now substitute the values:

S= d_A . R = d_A . r. G = r. (d_A . G) = r. Q_A

Elliptical curve cryptography is a method of encoding data files so that only specific individuals can decode them. ECC is based on the mathematics of elliptic curves and uses the location of points on an elliptic curve to encrypt and decrypt information. It increases the size of the encrypted message significantly more than RSA encryption. ECC algorithm is more complex and more difficult to implement than RSA.

5.3 Key pair generation: Input Elliptic curve domain parameter (**p,E,P,n**) Output Public key Q and private key d.

1. Select **d** =**R** [**1, (n-1)]**
2. Compute **Q**=**d*P**.
3. Return (**Q, d**).

The first task is to encode the plain text message m to be sent as an **x-y** point Pm. It is the point Pm that will be encrypted as cipher text and subsequently decrypted. To encrypt and send a message **Pm** to **B**, **A** Chooses a random positive integer **k** and produces the the cipher text Cm = {**K*P, Pm + k*Q**}, where **Q** is **B**'s public key. The sender transmits the point **C1=k*P** and **C2=Pm+K*q** to the recipient. To decrypt the cipher text, B multiplies by the first point in the pair by B's secret key and subtract the result from the second point as **Pm+k*q-d(k*P)=Pm+k(d*P)-d(kP)=Pm**.

5.4 Elliptic Curve Encryption Input: Elliptic curve domain parameter (**p, E, P, n**), public key **Q**, plain text m Output: Cipher text **Cm**

1. Represent the plain text **m** as a point **Pm** in **E** (**Fp**).
2. Select **k** [**1, (n-1**)].
3. Compute **C1=k*p**
4. Compute **C2=Pm+K*q**.
5. Return (**C1, C2**).

5.4 Elliptical Curve Decryption Input: Elliptic curve domain parameter (**p,E,P,n**), private key **d**, Cipher text **Cp**.

Output: Plaintext **m**.

1. Compute **Pm** =**C2-d*C1**

2. Compute (**Pm**).

VI. Security level RSA vs. ECC

When we talk about computation, symmetric key cryptography is less expensive than RSA based Public key. So in case of a security protocol, there should be less use of public key algorithms .Also by developing more efficient public key techniques, performance can be increased[5]. When we look at an alternative to RSA, the eyes goes on Elliptic Curve Cryptography (ECC). Same level of security can be achieved using ECC as compared to RSA with Smaller key sizes in Mobile Wimax sizes. It has also been proved through research that 160-bit ECC provides same level of security as 1024-bit RSA and 224-bit ECC provides same level of security as 2048-bit RSA [8].So we can say that in the same security level, ECC's smaller key sizes offer better computational efficiency ,also there are energy, memory and bandwidth saving. Figure: 1and figure: 2 shows Security level of ECC and RSA.

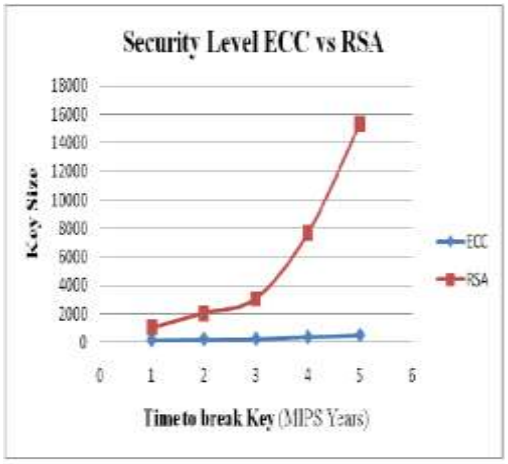

Fig.1: Security levels of ECC and RSA

Courtesy: V. S. Miller, Use of Elliptic Curves in Cryptography.

ISSN: 2320-7868 (Online)

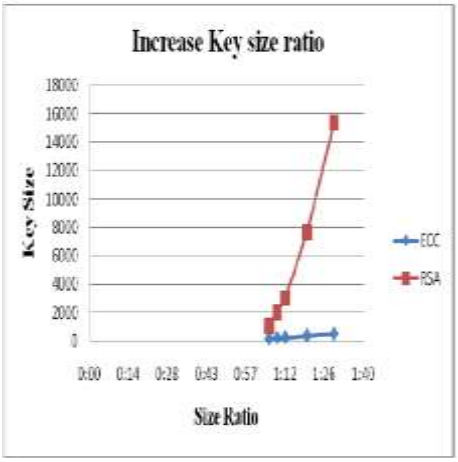

Fig. 2. Security levels of ECC and RSA
Courtesy:N. Demytko, A New Elliptic Curve Based Analogue of RSA.

So, we can say that it is really good to include ECC in Mobile Wimax in order to replace RSA based cryptography. One primary concern is level of security in today's world. Table 1 shows security comparison of Various Algorithm-key Size Combinations.

S.no	Security (Bits)	Symmetric Encryption Algorithm	RSA KeySize	ECC key Size	Key Size Ratio	Time to break (MIPS Years)
1	80	Skipjack	1024	160	1:6	3 million years
2	112	3DES	2048	224	1:9	10^{16}Year
3	128	AES-128	3072	256	1:12	1E+12
4	192	AES-192	7680	384	1:20	1E + 20
5	256	AES-256	15360	512	1:30	1E +36

TABLE 1. Security Comparison of Various Algorithm-Key Size Combinations.
Courtesy: Mugino Saeki, "Elliptic curve cryptosystems", School of Computer Science,McGill University.

Most of the people from cryptographic field say that at least 128 bits of security is offered by current systems, but what's all about. This is actually not the case of key length as people think. Key length and specific algorithm together offers security. For example, it is being thought that 128 bits of security can be achieved with 128-bit AES keys, 256-bit Elliptic Curve keys, and 3072-bit RSA keys. Now if we ignore implementation issues, these algorithms will offer the same level of security. As we know that RSA algorithm currently employ 1024 or 2048 bit keys, which are less secured when we compare it with AES-128. In Mobile WiMAX , RSA is reported to be 10 time slower as compared to ECC taking into account performance at 128-bit security levels, for private key operations such as signature generation or key management. It also expands at 256-bit security levels, where RSA is 50- to 100-times slower. Figure: 3 shows RSA vs ECC encryption time and Figure: 4 shows RSA vs ECC decryption time [6].

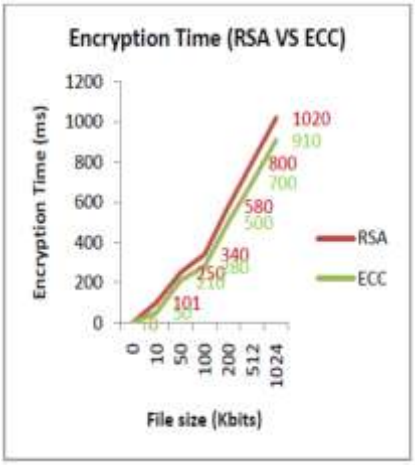

Fig. 3: RSA VS ECC Encryption time Courtesy: J. Borst, "Public key Cryptosystems using elliptic curves".

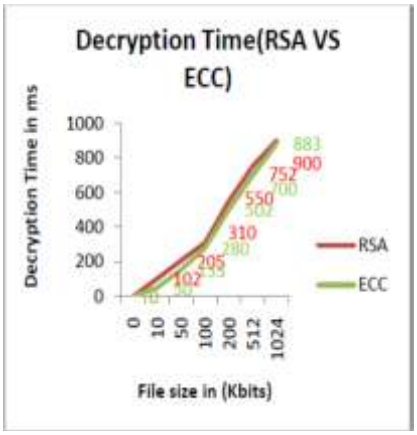

Fig.4: RSA VS ECC Decryption time
Courtesy: Aleksandar Jurisic and Alfred Menezes, Elliptic Curves and Cryptography.

VII. CONCLUSION

The paper reviews asymmetric key algorithms RSA and ECC. RSA is the most widely used public key technology today but the use of more simpler connected devices and demand for higher level of security will make continued reliance on RSA more challenging over time. These trends highlight a clear need for an efficient public key cryptosystem that can lower the capacity threshold for small devices to perform strong cryptography and increase a server's capacity to handle the secure communication. The RSA keys will need to grow to 2048 bits. ECC is an efficient alternative of RSA as a mean of improving SSL performance without restoring to expensive special purpose hardware. Compared to its traditional counterparts, ECC offers the same level of security using much smaller keys .This results in faster computations and saving in memory power and band width that are especially important in constrained environment, e.g. mobile phones, PDA's and smart cards. ECC offers equal security for a far smaller key size, thereby reducing processing overhead [8]. In today's world the internet is used for the secure communication. So the strong algorithms are required to provide the security to the data. The comparison table of RSA and ECC shows that ECC takes less time for encryption as well as decryption and also generates key of smaller size than

ISSN: 2320-7868 (Online)

RSA. So it can be concluded that ECC is more efficient than RSA. In future, comparison of other cryptographic algorithms

can be done and efforts can be done to reduce the execution time of RSA.

REFERENCES

[1] B.Schneier. Applied Cryptography. John Wiley and Sons, second edition, 2012.

[2] Cryptography and Elliptic Curves, koblitz, second edition, 2011.

[3] Julio Lopez and Ricardo Dahab, "An overview of elliptic curve cryptography", May 2011.

[4] V. Miller, "Uses of elliptic curves in cryptography", Advances in Cryptology -CRYPTO'85, LNCS 218, pp.417-426, 2011.

[5] Botes, J.J., Penzhorn, W.T., 1994. An implementation of an elliptic curve cryptosystem. Communications and Signal Processing. COMSIG-94. In Proceedings of the 1994 IEEE South African Symposium, 85 -90.

[6] Mugino Saeki, "Elliptic curve cryptosystems", M.Sc. thesis, School of Computer Science, McGill University, 2010.

[7] J. Borst, "Public key cryptosystems using elliptic curves", Feb. 2010.

[8] Aleksandar Jurisic and Alfred Menezes, "Elliptic Curves and Cryptography", Dr. Dobb's Journal, April 2010.

[9]Robert Milson, "Introduction to Public Key Cryptography, april 2009.

[10] Aleksandar Jurisic and Alfred J. Menezes, Elliptic Curves and Cryptography, 2008.

[11] V. S. Miller, "Use of Elliptic Curves in Cryptography". Advances in Cryptology CRYPTO'85, New York, Springer-Verlag.