International Journal of Scientific & Engineering Research, Volume 4, Issue 5, May-2013
ISSN 2229-5518

Performance Based Comparison Study of RSA and Elliptic Curve Cryptography

Rounak Sinha, Hemant Kumar Srivastava, Sumita Gupta

Abstract-Network is a collection of interconnected nodes which are spread over a large region. A node can be any device such as personal computer, mobile phone, tablet, WAP devices, pager, etc. Data is transmitted over channel of this network, which is prone to security threats such as loss of confidentiality, loss of integrity, fabrication attacks, etc. So, there is a need to secure this data transmission. It is achieved through Cryptography This paper introduces RSA algorithm and its concept in brief, and then proceeds to discuss Elliptic Curve Cryptography (ECC). The future scope of ECC has been pondered upon. In the process, a brief overview of Cryptosystems and different Cryptographic algorithms has been given. We have touched upon Symmetric Algorithms, which were based on the concept of using a common key. Brief details about Public Key Cryptography, a brainchild of Diffie and Hellman, have also been given. .RSA is one of the effective public key cryptographic algorithms, which needs time and memory; on the other hand ECC provides a strong alternative with its subtle features. Many research papers have been submitted on this cryptographic algorithm. Each paper has different perspective.

Index Terms— Attacks, Cryptography, Symmetric, PKC, RSA, ECC.

1 INTRODUCTION

Security often requires that data be kept safe from unauthorized access. And the best line of defense is physical security (placing the machine to be protected behind physical walls). However, physical security is not always an option (due to cost and/or efficiency considerations). Most computers are interconnected with each other openly, thereby exposing them and the communication channels that they use. Thus arises the problem of Network Security. This problem can be broken down into five requirements that must be addressed:

1. Confidentiality: assuring that private data remains private.
2. Authentication: assuring the identity of all parties attempting access.
3. Authorization: assuring that a certain party attempting to perform a function has the permissions to do so.
4. Data Integrity: assuring that an object is not altered illegally.
5. Non-Repudiation: assuring against a party denying a data or a communication that was initiated by them. [1], [2], [6], [13], [14]

- *Rounak Sinha is currently pursuing B.Tech in Computer Engineering from AmityUniversity, Noida, India, E-mail:rounaks@sify.com*
- *Hemant Kumar Srivastava is currently pursuing B.Tech in Computer Engineering from AmityUniversity, Noida, India, E-mail: hemant19914@gmail.com*
- *Sumita Gupta, Assistant Professor, Amity University, Noida, India, E-mail: sumitagoyal@gmail.com*

The solution lies in Cryptosystems. What do we understand by Cryptosystem? It 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. [6], [7], [8], [9], [11], [15]

2 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.

2.1 Symmetric Algorithms

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. For example, let us suppose that *Ram* wants to share some information *(i)* with *Rahim* secretly. What he is going to do is this:

Step1- Ram encrypts the information i using a key x, to get encrypted text

$$E(x, i) \qquad (1)$$

Step 2- Assuming that *Rahim* already has his copy of key x, he can easily decode it to discover the original information, *i.*

$$Dx(E(x,i) \rightarrow i \qquad (2)$$

2.1.1 ***Example illustrating Symmetric Algorithm***

Ram might want to transfer a number 64 to *Rahim*. Let us assume that the key available to him is 3. Now 64 is *i,* and 3 is x. Using equation (1), *Ram* encrypts *i* as

$$E(x,i) \rightarrow 64^3 \rightarrow 262144 \qquad (1a)$$

E(x,i) is now sent by *Ram* to *Rahim. Rahim* now decrypts the message using equation (2) as

$$Dx(E(x,i) \rightarrow Dx(262144)$$

$$\rightarrow \sqrt[3]{262144} = \quad 64 = i \qquad (2a)$$

2.1.2 ***Pros and Cons 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], [8]

2.2 Public Key Cryptography

Whitefield Diffie and Martin Hellman, two researchers working on cryptosystems, came up with the concept of Public Key Cryptography in 1976. This technique divided the all important *key* into two- *public* and *private.* In short, if *Ram* wanted to send some information *i* to *Rahim* now, the process would go like this:

Step 1- *Ram* and *Rahim* agree on two large prime numbers, *a* and *b*, which need not be kept secret.

Step 2- *Ram* decides another random large number x, and R is calculated as

$$R = b^x \, mod \, a \qquad (3)$$

Step 3-*Ram* sends the number *R* to *Rahim*.

Step 4- *Rahim* independently selects another large integer y randomly, to calculate *S*, such that

$$S = b^y \, mod \, a \qquad (4)$$

Step 5- Rahim sends the number *S* to *Ram.*

Step 6- Ram can now compute the secret key *K1* using

$$K1 = S^x \, mod \, a \qquad (5)$$

And the same can be done by *Rahim* to calculate

$$K2 = R^y \, mod \, a \qquad (6)$$

2.2.1 Example illustrating PKC Algorithm

Let the two prime numbers be chosen as a=11, b=7. Next, let *Ram* choose another random number, say, *x*=*3*. Using equation (3), *R* can be calculated as [1]

$$R = b^x \, mod \, a = 7^3 mod \, 11$$
$$= 343 \, mod \, 11 = 2 \qquad (3a)$$

This number, *R*, can now be sent to *Rahim*. *Rahim*, on the other hand, chooses to calculate S using equation (4) to give

$$S = b^y \, mod \, a = 7^6 mod \, 11$$
$$= 117649 \, mod \, 11 = 4 \qquad (4a)$$

This number, *S*, can now be sent to *Ram. Ram* now calculates private key *K1* using equation (5);

$$K1 = S^x \, mod \, a = 4^3 mod \, 11$$

$$\rightarrow 64 \, mod \, 11 = 9 \qquad (5a)$$

On the other side, *Rahim* too, calculates private key *K2* using equation (6);

$$K2 = R^y \, mod \, a = 2^6 \, mod \, 11$$
$$\rightarrow 64 \, mod \, 11 = 9 \qquad (6a)$$

As obvious, equations (5(a)) and (6(a)) give the same resultant key. The mathematical theory behind this is that the algorithm becomes quite secured, as it is exceptionally difficult to calculate logarithms in a finite field, as compared with the ease of calculating exponentiations in the same field. What happens here is, the first part of the key is known to everyone. The other two parts of the key have to be made available by *Ram* and *Rahim.* When *Ram* receives a key that is two-thirds complete from *Rahim*, he adds the remaining one-third part to complete his key. The same process is performed for *Rahim's* key. However, an important thing is that the two cannot compute each other's part of the key.

2.2.2 *Pros and Cons 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. [1], [6], [7], [8]

2.3 The RSA algorithm

RSA is used to denote the surnames of its devisers, Ron Rivest, Adi Shamir and Leonard Adleman. This, by far, remains the most popular technique in use since its inception in 1977. This technique makes the encrypted text virtually impossible to decode, what with its twin problems of factorizing very large numbers, and RSA problem (finding a value x, so that,

$$xe = c(mod\ y) \tag{7}$$

where *(e, y)* is the public key and *c* is the encrypted text). Considering the earlier problem of information exchange between *Ram* and *Rahim,* we proceed with the RSA algorithm now:[1]

Step 1- Two large prime numbers *(a, b)* are chosen, such that $a \neq b$. This is done randomly.

Step 2- $$P = ab \tag{8}$$

is calculated.

Step 3- Next, the totient is computed, to give

$$\emptyset(P) = (a-1)(b-1) \tag{9}$$

Step 4- Next, an integer n is chosen such that $1<n< \emptyset(P)$, and n is co prime to $\emptyset(P)$.

Step 5- m is computed to give

$$(m\ \times\ n)\ mod\ \emptyset(P) = 1 \tag{10}$$

Now, the public key consists of the modulus and the encryption exponent. While the private key, consists of the modulus and the decryption exponent. *Ram* transmits the *public key* to *Rahim*, and keeps the *private key* secret. a and b are deemed sensitive since they are the factors of P, and m can be calculated once the value of n is known. Now, with some more calculations, information i can be encrypted at *Ram's* end and be sent to *Rahim* who could then decrypt it.

The encrypted text E can be calculated as

$$E = i^n\ mod\ P \tag{11}$$

The text can be then decrypted using

$$i = E^m\ mod\ P \tag{12}$$

2.3.1 *Example illustrating Symmetric Algorithm*

Let us take a situation where we have chosen *a*=11, *b*=7. Now, equation (8) gives

$$P = ab = 11 \times 7 = 77 \tag{8a}$$

Then, equation (9) gives

$$\emptyset(P) = (a-1)(b-1)$$
$$\rightarrow\ (11-1)(7-1)\ = 60 \tag{9a}$$

From here, we have to select an integer n based on the condition given in *STEP 4*. For convenience, let us select *n=7.*

We proceed to compute m, so that it satisfies equation (10). After some calculations, we can see that the integer *43* is a suitable candidate for m.

$(43\ \times\ 7)\ mod\ 60 = 1$ is true.

Let us suppose *Ram* wants to transfer the information, *i=64,* to *Rahim.* He would have to encrypt the information using equation (11), so that:

$$E = i^n\ mod\ P = 64^7\ mod\ 60$$
$$= 4398046511104\ mod\ 60 = 4 \tag{11a}$$

Now that we *Ram* has transferred the encrypted information, *E,* to *Rahim,* he can decrypt it to find *i*, using equation (12);

$$i = E^m\ mod\ P = 4^{43} mod\ 60 =\ 64 \tag{12a}$$

This is the same information that Ram sent.

Normally, numbers having a length of hundred or more digits are chosen, rendering any decryption attack virtually toothless. RSA technique in itself is quite simple. The actual challenge lies in the selection of the right keys. It might appear that an attacker knowing about the public key *n,* and the number *P,* could find the private key m trial and error. However for that to happen, the attacker would first need to find out the values of a and b. And once we choose sufficiently large numbers as a and b, it becomes virtually pointless to delve into trial and error method. In fact, Mathematical

research suggests that it would take about 70 years to calculate *a* and *b*, if *P* happens to be 100 digits-long.
As of now, 256-bit length keys are considered breakable in a few hours using a personal computer. The current recommended key-length is 2048-bits. Herein lays the catch, as this makes it harder for slower processor using machines to work effectively.[15], [16]

2.3.2 Pros and Cons 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.

Some Disadvantages of RSA:

- The main disadvantage is the slow speed of the encryption process.
- If private keys of users are not available, it is vulnerable to impersonation.[1], [10], [12]

2.4 Elliptic Curve Cryptography (ECC)

This Cryptography technique was discovered by Victor Miller (IBM) and Neil Koblitz (University of Washington) in 1985. The Elliptic Curve Discrete Logarithm problem belongs to the class of NP-Hard problems. It is on this problem that the ECC is based on. It is a public key cryptography technique. It is based on elliptic curves over finite fields. Normally, an elliptic curve is defined by the equation,

where *a* and *b* are constants.

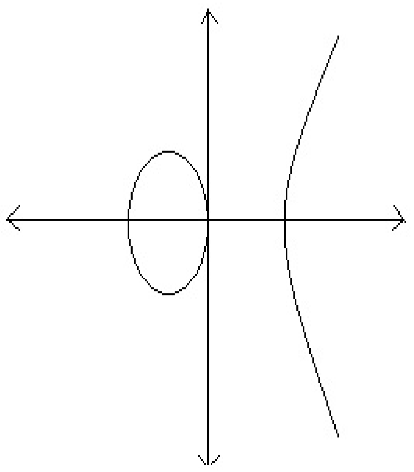
Fig 1: elliptic curve for $y^2=x^3-x$

.

Different variants of this equation exist. The graphs for two such variants have been shown in Fig 1 and Fig 2.

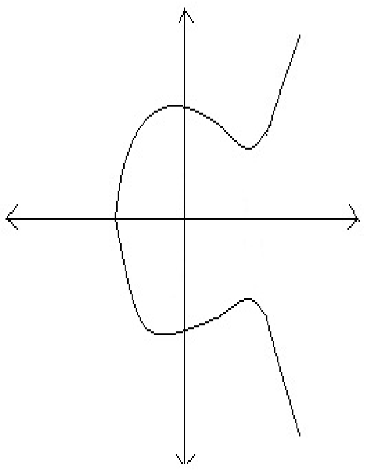
Fig 2: elliptic curve for $y^2=x^3-x+1$

2.4.1 Addition and Multiplication on Elliptic Curves

For a point on an elliptic curve, we can define two points on the same which would add up to give that third point, while following rules of normal addition. We can also define multiplication *kN* where *k* is a positive integer and *N* is a point as the sum of *k* copies of *N*. For instance,

2.4.2 Process of Elliptic Curve Cryptography

A non-secret Elliptic curve and a non-secret fixed curve point P are agreed upon by the users, say, *Ram, Rahim and Ali.* A secret random integer, R_k, is chosen by *Ram* as his secret key, and he publishes his public key, the curve point $R_p=R_kP$. The same process is followed by *Rahim and Ali.* Now if *Ram* wishes to communicate some secret message to *Ali*, he can simply compute R_kA_p, and use the result as the secret key for a conventional symmetric block cipher. *Ali* can compute the same number by calculating A_kR_p, since

The security of the scheme is based on the assumption that it is difficult to compute *k* given *F* and *kF*.

2.4.3 Process of choosing a Fixed curve:

Normally, a field has addition, subtraction, multiplication and division operations. Always, a result that is in the field is produced by these operations, with the exception of division by zero, which happens to be undefined. First of all a finite field is selected. If the field happens to be *GF(k)* where k happens to be a large prime number, the xy term gets omitted, and the equation becomes

$$y^2 = x^3 + ax^2 + b \qquad (16)$$

$where\ 4a^3 + 27b^2 \neq 0$

and, if the field happens to be *GF(2^m)*, the xy term is included to give the equation as

$$y^2 + xy = x^3 + ax + b \qquad (17)$$

Fields *GF(p^m)* with both p > 2 and m > 1 are not considered here.

2.4.4 Process of choosing a Fixed point

On an Elliptic curve in the field *GF(p^m)*, for any point *P*,

$$\lim_{k\to\infty} kP \to 0 \qquad (18)$$

For some i and j, *j>i*, iP=jP. This means that for *s=j-i, sP=0.* The *order of the point* is the minimum value of *s* for which this is true. Also, *s* must divide the *order of the curve.* From the security point of view, the curve and fixed point are chosen so that the order of the fixed point F is a large prime number. Also, the order of the fixed point should also satisfy the MOV condition to prevent certain possible attacks. Now, computing *k* from *kF* and *F* takes roughly $2^{x/2}$ operations given that the order of the fixed point F is an *x*-bit prime. And this means that compared to RSA, the public keys and signatures can be much smaller.

2.4.5 Pros and Cons of ECC algorithm

ECC has evolved as a main challenger to RSA.

Some advantages of ECC are:

- Increased speed.
- Less memory requirement.
- Smaller key size

Disadvantage of ECC:

- Relatively, it is much less explored and known about. [3], [4], [9] ,[11]

2.5 ECC and RSA

When it comes to ECC, the basic operation is point addition, which is known to be very expensive. That makes it very unlikely to discover a sub-exponential attack on ECC in the near future (even though some particular curves of ECC are prone to them and can be avoided easily, as it is quite easy to distinguish them). Meanwhile, RSA already has a known sub-exponential attack. So, the bits requirement for RSA generated key pair is supposed to rise much faster than that for ECC generated one.

Also, for a similar level of security, the numbers involved in ECC are smaller as compared to RSA, as can be derived from the data displayed in tables 1, 2, 3, and 4. Hence, the number of transistors required for security also gets reduced in case of ECC. And since PKC is used to transfer short messages, ECC comes in handy as it is faster than RSA for short messages. The bandwidth requirement becomes quite sim$Where\ b \neq 0$ase of longer messages.

Table 1. Comparable Key Size (in bits) [3]

Symmetric Algorithms	ECC	RSA
80	163	1024
112	233	2240
128	283	3072
192	409	7680
256	571	15360

Table 2. Key Generation Performance [3]

Key Length (bits)		Time (s)	
RSA	ECC	RSA	ECC
1024	163	0.16	0.08
2240	233	7.47	0.18
3072	283	9.80	0.27
7680	409	133.90	0.64
15360	571	679.06	1.44

Table 3. Signature Generation Performance [3]

Key Length (bits)		Time (s)	
RSA	ECC	RSA	ECC
1024	163	0.01	0.15
2240	233	0.15	0.34
3072	283	0.21	0.59
7680	409	1.53	1.18
15360	571	9.20	3.07

Table 4. Signature Verification Performance [3]

Key Length (bits)		Time (s)	
RSA	ECC	RSA	ECC
1024	163	0.01	0.23
2240	233	0.01	0.51
3072	283	0.01	0.86
7680	409	0.01	1.80
15360	571	0.03	4.53

Table 5. ECC and RSA: an overview [4, 5]

Parameters	**ECC**	**RSA**
Computational Overheads	Roughly 10 times than that of RSA can be saved	More than ECC
Key Sizes	System parameters and key pairs are shorter for the ECC.	System parameters and key pairs are larger for the RSA.
Bandwidth saving	ECC offers considerable bandwidth savings over RSA	Much less bandwidth saving than ECC
Key Generation	Faster	Slower
Encryption	Much Faster than RSA	At good speed but slower than ECC
Decryption	Slower than RSA	Faster than ECC
Small Devices efficiency	Much more efficient	Less efficient than ECC

3 CONCLUSION AND FUTURE SCOPE

We have been witnessing a tremendous growth in the use of Internet nowadays. The world has been digitized. It is very much normal to expect a scenario in near future where almost all the work will become dependent on Internet. This would require a lot of softwares. And for every software generator there are hundreds of hackers. That means, in case of an attack, it will take seconds to destroy world's largest economies through Internet. Hence we require strong algorithms to secure the networks.

Our findings suggest that RSA key generation is significantly slower than ECC key generation for RSA key of sizes 1024 bits and greater. Considering there are affordable devices that can break RSA keys smaller than 1024 bits in a matter of days, the cost of key generation can be considered as a factor in the choice of public key systems to use when using digital signatures, especially for smaller devices with lesser computational resources. The difference in their key sizes grows exponentially to maintain the same relative power as compared to the average computing power available. In fact, RSA Security on their own have admitted on their website that ECC is the technique to be in demand in the future.

However, the fact remains that ECC was discovered during the process of trying to find out new ways to attack on systems using RSA techniques. That means, ECC is still in the process of being researched upon and a vast area remains unexplored. RSA, on the other hand, is well researched and trusted. That can be a hindrance.

REFERENCES:

[1] A. Kahate," Cryptography and Network Security", TMH-2003.
[2] P. Zimmermann,"Introduction to Cryptography", 2000.
[3] N. Jansma and B. Arrendondo,"Performance Comparison of Elliptic Curve and RSA Digital Signatures", 2004.
[4]The Study Material- http://www.thestudymaterial.com/presentation-seminar/electronics-presentation/248-ellip.html?start=3
[5] V.B. Kute et al,"A software comparison of RSA & ECC**"**, *IJCSA Vol 2, No.1*, April/May 2009.
[6] S. Vanstone ," Handbook of Applied Cryptography", CRC Press, 1996.
[7] W. Stallings," Cryptography and Network Security", 2006.
[8]Wikipedia.org- http://en.wikipedia.org/wiki/Cryptography
[9]Wikipedia.org- http://en.wikipedia.org/wiki/Elliptic_curve
[10]Wikipedia.org- http://en.wikipedia.org/wiki/RSA
[11] R. Zuccherato,"Elliptic Curve Cryptography Support in Entrust", 2000.
[12] RSA Laboratories TR201,"High-Speed RSA Implementation",*Technical report*, November 1994.
[13] B.A. Forouuzan,"Data Communication and Networking", (4/e, Tata McGraw-Hill).
[14] W.Stallings,"Network Security Essentials, Applications and Standards", (2/e , Pearson Education).
[15] R.L. Rivest et al,"A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", 1983.
[16] B. Kaliski,"The Mathematics of the RSA Public-Key Cryptosystem", 1989.