A Comparative Study of Public Key Cryptosystem based on ECC and RSA

Arun kumar[*], Dr. S.S. Tyagi[*], Manisha Rana[**], Neha Aggarwal[#], Pawan Bhadana[#]

*Computer Science & Engineering
[**]Electronics & Communication Engineering
Manav Rachna International University
Faridabad, India
#Computer Science & Engineering
B.S.A. Institute of Technology & Management
Faridabad, India

Abstract—The paper gives an introduction to the public key cryptography and its use in applications such as Key Agreement, Data Encryption and Digital Signature. The main emphasize is on some public key algorithms such as RSA and ECC along with the idea how ECC is better and more secure method of encryption in comparison to RSA and other asymmetric cryptosystems.

Keywords: Asymmetric Key Cryptography, ECC, RSA

I. INTRODUCTION

Encryption [13] algorithm plays an important role for information security guarantee. Encryption is the process of transforming plaintext data into cipher text in order to conceal its meaning and so preventing any unauthorized recipient from retrieving the original data. The main task of encryption is to ensure secrecy. Companies usually encrypt their data before transmission to ensure that the data is secure during transit. The encrypted data is sent over the public network and is decrypted by the intended recipient. The encryption algorithms are usually summarized into two popular types: Symmetric key encryption and Asymmetric key encryption.

In Symmetric key encryption, only one key is used to encrypt and decrypt data. The key should be distributed before transmission between entities. Therefore, key plays an important role in Symmetric key encryption [12]. Strength of Symmetric key encryption depends on the size of key used. For the same algorithm, encryption using longer key is harder to break than the one done using shorter key.

The representative Symmetric key cryptography algorithms include RC2, DES, 3DES, RC5, Blowfish, and AES, which use certain- or variable-length key [13]. Asymmetric key encryption is used to solve the problem of key distribution. In Asymmetric key encryption, private key and public key are used. Public key is used for encryption and private key is used for decryption (E.g. RSA, Digital Signatures and ECC).

In public key cryptography [4], each user or the device taking part in the communication have a pair of keys, a public key and a private key, and a set of operations associated with the keys to do the cryptographic operations. Only the particular user/device knows the private key whereas the public key is distributed to all users/devices taking part in the communication. Since the knowledge of public key does not compromise the security of the Algorithms, it can be easily exchanged online.

The paper has been organized as follows: section II describes the two asymmetric algorithms, that is, RSA and ECC; section III illustrates the various ECC operations; section IV shows the advantages of ECC and last section concludes the proposed work.

II. ASYMMETRIC ALGORITHMS

In this section, an overview and cryptanalysis of some of the asymmetric algorithms are reviewed.

A. RSA

The RSA algorithm [13] is named after Ron Rivest, Adi Shamir, Len Adleman, invented in 1977. It can be used for both public key encryption and digital signatures. Its security is based on the difficulty of factoring large integers. The first asymmetric cryptosystem to have seen widespread use is also one of the most accessible illustrations of this principle in action. RSA gets its security from the difficulty of factoring very large numbers. The difficulty of getting the plaintext message back from the cipher text and the public key is related to the difficulty of factoring a very large product of two prime numbers.

As an illustration of this: take two very large prime numbers — say, 200 digits long, and then multiply them together. Now the result retrieved has two particular properties:

(i) It is very large (about 400 digits in length),
(ii) It has two, and exactly two factors, both prime numbers--the two primes which are just multiplied together.

The two prime numbers can be given easily from which the product can be calculated. But finding the primes given only the product is more difficult. So much more, in fact, that once the numbers get adequately large, it is almost impossible to find them. You simply cannot assemble enough computing power to do so. So the multiplying of two large prime numbers together is the (relatively) easy forward function in this asymmetric algorithm. Its inverse, the factor finding operation is considerably more difficult, and in practical terms, it’s intractable. The RSA system [14] employs this fact to generate public and private key pairs. The keys are functions of the product and of the primes. Operations performed using the cryptosystem is arranged in such a manner that the operations need to be tractable are required to perform the relatively easy forward function — multiplication. Conversely, the operations need to make difficult — finding the plaintext from the ciphertext using only the public key — require performing the inverse operation — solving the factoring problem.

B. ECC

Elliptic Curve Cryptography (ECC) [3] is a public key cryptography. In public key cryptography, each user or the device taking part in the communication generally have a pair of keys, a public key and a private key, and a set of operations associated with the keys to do the cryptographic operations. Only the particular user knows the private key whereas the public key is distributed to all users taking part in the communication. Some public key algorithm may require a set of predefined constants to be known by all the devices taking part in the communication. ‘Domain parameters’ in ECC is an example of such constants. Public key cryptography, unlike private key cryptography, does not require any shared secret between the communicating parties but it is much slower than the private key cryptography.

The mathematical operations of ECC is defined over the elliptic curve $\mathbf{y^2 = x^3 + ax + b}$**,** where $\mathbf{4a^3 + 27b^2 \neq 0}$. Each value of the ‘a’ and ‘b’ gives a different elliptic curve. All points (x, y) which satisfies the above equation plus a point at infinity lies on the elliptic curve. The public key is a point in the curve and the private key is a random number. The public key is obtained by multiplying the private key with the generator point G in the curve. The generator point G, the curve parameters ‘a’ and ‘b’, together with few more constants constitutes the domain parameter of ECC [4] . One main advantage of ECC is its small key size. A 160-bit key in ECC is considered to be as secured as 1024-bit key in RSA.

Discrete Logarithm Problem

The security of ECC depends on the difficulty of Elliptic Curve Discrete Logarithm Problem. Let P and Q be two points on an elliptic curve such that kP = Q, where k is a scalar. Given P and Q, it is computationally infeasible to obtain k, if k is sufficiently large. k is the discrete logarithm of Q to the base P. Hence the main operation involved in ECC is point multiplication, i.e. multiplication of a scalar k with any point P on the curve to obtain another point Q on the curve.

Elliptic Curves

The way that the elliptic curve operations are defined is what gives ECC its higher security at smaller key sizes [10]. An elliptic curve is defined in a standard, two dimensional x, *y* .Cartesian coordinate system are given below as equation(1) and equation(2):

$$y^2 = x^3 + ax + b \qquad (1)$$

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

There is a more focus on the elliptic curves defined over GF(2^N).The mathematic foundation of ECC is based on the above equations. A sample representation of elliptic curve is given in Figure 1.

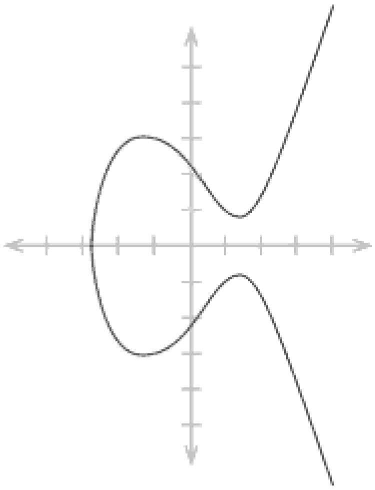

Figure1: An Elliptic curve

In elliptic curve cryptosystems, the elliptic curve is used to define the members of the set over which the group is calculated, as well as the operations between them which define how math works in the group. It is done as follows: imagine a graph labeled along both axes with the numbers of a large prime field.

III. ECC OPERATIONS

Some of the operations which can be performed on ECC are given below in detail:

1) Point Multiplication

The dominant operation in ECC cryptographic schemes is point multiplication. This is the operation which is the key to the use of elliptic curves for asymmetric cryptography---the critical operation which is itself fairly simple, but whose inverse (the elliptic curve discrete logarithm problem defined below) is very difficult [6]. ECC arranges itself so that when you wish to performance operation the cryptosystem should make easy encrypting a message with the public key, decrypting it with the private key the operation you are performing is point multiplication [7].

Point multiplication is simply calculating the value of *kP*, where *k* is an integer and *P* is a point on the elliptic curve defined in the prime field. In point multiplication, a point P on the elliptic curve is multiplied with a scalar k using elliptic curve equation to obtain another point Q on the same elliptic curve, i.e. kP=Q. Point multiplication is achieved by two basic elliptic curve operations.

- Point addition, adding two points J and K to obtain another point L i.e., L = J + K.

- Point doubling, adding a point J to itself to obtain another point L i.e. L = 2J.

2) Point addition

Point addition is the addition of two points J and K on an elliptic curve to obtain another point L on the same elliptic curve.

Consider two points J and K on an elliptic curve as shown in figure (a). If K ≠ -J then a line drawn through the points J and K will intersect the elliptic curve at exactly one more point –L. The reflection of the point –L with respect to x-axis gives the point L, which is the result of addition of points J and K. Thus on an elliptic curve L = J + K. If K = -J the line through this point intersect at a point at infinity O. Hence J + (-J) = O. This is shown in figure 2(b). O is the additive identity of the elliptic curve group. A negative of a point is the reflection of that point with respect to x-axis.

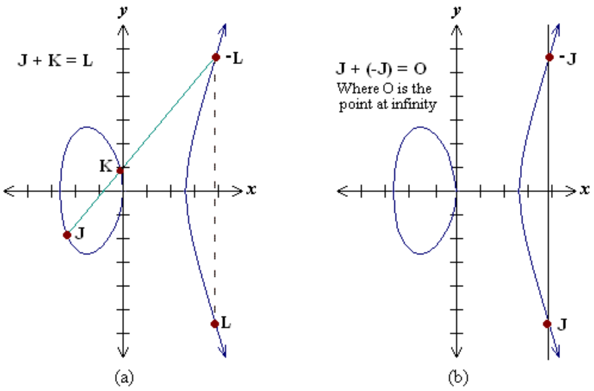

Figure 2: Addition of two points

2) **Point doubling**

Point doubling is the addition of a point J on the elliptic curve to itself to obtain another point L on the same elliptic curve. To double a point J to get L, i.e. to find L = 2J, consider a point J on an elliptic curve as shown in figure 3(a). If y coordinate of the point J is not zero then the tangent line at J will intersect the elliptic curve at exactly one more point –L. The reflection of the point –L with respect to x-axis gives the point L, which is the result of doubling the point J.

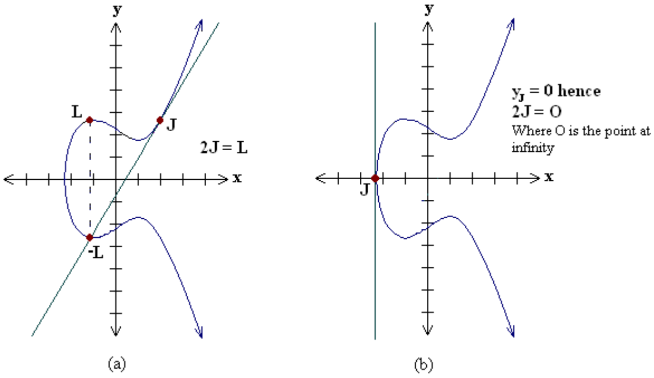

Figure3: Doubling of points

Thus L = 2J. If y coordinate of the point J is zero then the tangent at this point intersects at a point at infinity O. Hence 2J = O when yJ = 0. This is shown in figure 3(b).

IV. ADVANTAGES OF ECC

There are lot of differences between ECC and RSA.These differences become more and more pronounced as security levels increase (and, as a corollary, as hardware gets faster, and the recommended key sizes must be increased). A 384-bit ECC key matches a 7680-bit RSA key for security [3,5]. The smaller ECC keys mean the cryptographic operations that must be performed by the communicating devices can be squeezed into considerably smaller hardware, that software applications may complete cryptographic operations with fewer processor cycles, and operations can be performed that much faster, while still guaranteeing equivalent security. This means, in turn, less heat, less power consumption, less real estate consumed on the printed circuit board, and software applications that run more rapidly and make lower memory demands which lead in turn to more portable devices which run longer, and produce less heat.

Now, consider these three facets of the problem

(i) Firstly, the fact that the security and practicality of a given asymmetric cryptosystems relies upon the difference in difficulty between doing a given operation and its inverse.

(ii) Second, the fact that the difference in difficulty between the forward and the inverse operation in a given system is a function of the key length in use, due to the fact that the difficulty of the forward and the inverse operations increase as very different functions of the key length; the inverse operations get harder faster.

(iii) Third, the fact that as the longer key lengths are used to adjust the greater processing power are now available to attack the cryptosystem, even the ‘legitimate’ forward operations get harder, and require greater resources (chip space and/or processor time), though by a lesser degree than do the inverse operations.

If these three facets can be understand, one can easily grasp the advantages of ECC over other asymmetric cryptosystems. The key comparison of various algorithms such as ECC, RSA and Key Size Ratio is given in Table1.

Table1: Key Comparison of Algorithms

ECC KEY SIZES (Bits)	RSA KEY SIZES (Bits)	KEY SIZE RATIO (Bits)
163	1024	1:6
256	3072	1:12
384	7680	1:20
512	15360	1:30

V. CONCLUSION

In this paper, it is concluded that ECC is a stronger option than the RSA and discrete logarithm systems in the future. Thus, it can be said that ECC is an excellent choice for doing asymmetric cryptography in portable, necessarily constrained devices right now. As an example: a popular, recommended RSA key size for most applications is 2,048 bits. For equivalent security using ECC, a key of only 224 bits is needed. In short, to make a device with smaller band, make them run longer on the same battery, and produce less heat, the most elegant and most efficient asymmetric cryptosystem that scales for the future is ECC.

REFERENCES

[1] Darrel Hankerson, Julio Lopez Hernandez, Alfred Menezes, Software Implementation of Elliptic Curve Cryptography over Binary Fields, 2000.

[2] M. Brown, D. Hankerson, J. Lopez, A. Menezes, Software Implementation of the NIST Elliptic Curves Over Prime Fields, 2001.

[3] Certicom, Standards for Efficient Cryptography, SEC 1: Elliptic Curve Cryptography, Version 1.0, September 2000.

[4] Certicom, Standards for Efficient Cryptography, SEC 2: Recommended Elliptic Curve Domain Parameters, Version 1.0, September 2000

[5] Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone, Handbook of Applied Cryptography, CRC Press, 1996

[6] I. F. Blake, G. Seroussi, and N. P. Smart. Elliptic curves in cryptography, volume 265 of London Mathematical Society Lecture Note Series.Cambridge University Press, Cambridge, 2000. Reprint of the 1999 original.

[7] D. Shanks. Class number, a theory of factorization, and genera. In 1969 Number Theory Institute (Proc. Sympos. Pure Math., Vol. XX, State Univ. New York, Stony Brook, NY, 1969), pages 415 440. Amer. Math. Soc., Providence, RI, 1971.

[8] J. H. Silverman. The arithmetic of elliptic curves, volume 106 of Graduate Texts in Mathematics. Springer-Verlag, New York, 1986.

[9] W. Trappe and L.Washington. Introduction to cryptography with coding theory, (2nd ed.). Prentice Hall, Upper Saddle River, NJ, 2006.

[10] L. C. Washington. Elliptic curves. Number theory and cryptography, (2nd ed.). Chapman & Hall/CRC, New York, NY, 2008.

[11] A. S. Tanenbaum, "Modemn Operating Systems", Prentice Hall, 2003.[3] M. J. B. Robshaw, "Block Ciphers", Technical Report, RSA Laboratories, Number TR - 601, July 1994.

[12] H.M. Heys and S.E. Tavares, "On the Security of the CAST Encryption Algorithm," Proceedings of the Canadian Conference on Electrical and Computer Engineering, Halifax, Nova Scotia, Sep 1994, pp. 332-335.

[13] R. Rivest, “The encryption algorithm,” in Fast Software Encryption, ser. LNCS, vol. 1008. Springer-Verlag, 1995, pp. 86–96.

[14] M.A. Viredaz and D.A. Wallach, “Power Evaluation of a Handheld Computer: A Case Study,” WRL Research Report, 2001/1.

[15] P. Ruangchaijatupon, P. Krishnamurthy, "Encryption and Power Consumption in Wireless LANs-N,’’ The Third IEEE Workshop on Wireless LANs-September 27-28, 2001- Newton, Massachusetts.

[16] M. Matsui, “Linear Cryptanalysis Method for DES Cipher”, Advances an Cryptology: Proceedings of Eurocrypt ’93**,** Springer-Verlag, pp. 386-397, 1994.

Arun Kumar received B.Tech degree in Computer Science & Engineering from Kurukshetra University in 2007 and is persuing M.Tech. in CSE. Presently, he is working as Lecturer in Computer Engineering department in B.S.A. Institute of Technology & Management, Faridabad. His areas of interest is Cryptography.

Dr. S. S. Tyagi received B.Tech in Computer Science and Engineering from Nagpur University and M.E from BITS, Pilani and Ph.D in Computer Science from Kurukshetra University, Kururkshetra. Presently, he is working as Professor in Computer Engineering department in Manav Rachna International University, Faridabad. His areas of interests are Wireless Security, Mobile Ad hoc Networks and Wireless Mesh Networks.

Manisha Rana received B.Tech degree in Computer Science & Engineering from Maharshi Dayanand University in 2004 and M.Tech. in ECE from MDU in 2008. Presently, she is working as Assistant Professor in MRIU, Faridabad.

Neha Aggarwal received B.E. degree in Information technology with Hons. from Maharshi Dayanand University in 2007 and M.Tech. in IT from YMCA in 2010. Presently, she is working as Senior Lecturer in Computer Engineering department in B.S.A. Institute of Technology & Management, Faridabad. Her areas of interests are Data Mining, Search Engine & Web Mining.

Pawan Bhadana received B.E degree in Computer Science & Engineering from Maharshi Dayanand University in 2003 and M.Tech. in CSE from YMCA in 2008. Presently, he is working as Assistant Professor in Computer Engineering department in B.S.A. Institute of Technology & Management, Faridabad. His areas of interests are Networking and artificial intelligence.