ELLIPTIC CURVES: AN EFFICIENT AND SECURE ENCRYPTION SCHEME IN MODERN CRYPTOGRAPHY

Shubham Agarwal[1], Dr. Anand Singh Uniyal[2]

[1,2] *Department of Mathematics, M.B. (Govt.) P.G. College, Haldwani, Uttarakhand, (India)*

ABSTRACT

There are many drawbacks in current encryption algorithms in terms of security, real-time performance, etc, and researchers are continuously formulating various algorithms to overcome the same. Among them, the elliptic curve cryptography (ECC) is evolving as an important cryptography, and shows a promise to be an alternative of RSA. Small size, high security and other features characterize ECC. The aim of this paper is to develop a basis for utilizing efficient encryption schemes in cryptography with low computing power and resources. Elliptic Curve Cryptography (ECC) fits well for an efficient and secure encryption scheme. It is more efficient than the ubiquitous RSA based schemes because ECC utilizes smaller key sizes for equivalent security.

Keywords : Cryptography, Elliptic Curve Cryptography (ECC), RSA, Security.

I INTRODUCTION

The Concise Oxford Dictionary (2006) defines cryptography as the art of writing or solving codes. This definition may be historically accurate, but it does not capture the essence of modern cryptography. First, it focuses solely on the problem of secret communication. This is evidenced by the fact that the definition specifies "codes", elsewhere defined as "a system of pre-arranged signals, especially used to ensure secrecy in transmitting messages".
Second, the definition refers to cryptography as an art form. Indeed, until the 20th century, cryptography was an art. Constructing good codes, or breaking existing ones, relied on creativity and personal skill. There was very little theory that could be relied upon and there was not even a well-defined notion of what constitutes a good code [1].

II MODERN CRYPTOGRAPHY

In the late 20th century, the picture of cryptography radically changed. A rich theory emerged, enabling the rigorous study of cryptography as a science. Furthermore, the field of cryptography now encompasses much more than secret communication, including message authentication, digital signatures, protocols for exchanging secret keys, authentication protocols, electronic auctions and elections, and digital cash. In fact, modern cryptography can be said

to be concerned with problems that may arise in any distributed computation that may come under internal or external attack. Without attempting to provide a perfect definition of modern cryptography, we would say that it is the scientific study of techniques for securing digital information, transactions, and distributed computations. Another very important difference between classical cryptography (say, before the 1980s) and modern cryptography relates to who uses it. Historically, the major consumers of cryptography were military and intelligence organizations. Today, however, cryptography is everywhere! Security mechanisms that rely on cryptography are an integral part of almost any computer system. Users (often unknowingly) rely on cryptography every time they access a secured website. Cryptographic methods are used to enforce access control in multi-user operating systems, and to prevent thieves from extracting trade secrets from stolen laptops. Software protection methods employ encryption, authentication, and other tools to prevent copying. The list goes on and on.

In short, cryptography has gone from an art form that dealt with secret communication for the military to a science that helps to secure systems for ordinary people all across the globe. This also means that cryptography is becoming a more and more central topic within computer science.

2.1 Cryptographic System

A cryptographic system is any computer system that involves cryptography. Such systems include for instance, a system for secure electronic mail which might include methods for digital signatures, cryptographic hash functions, key management techniques, and so on. Cryptographic systems are made up of cryptographic primitives, and are usually rather complex. Because of this, breaking a cryptosystem is not restricted to breaking the underlying cryptographic algorithms- usually it is far easier to break the system as a whole.

2.2 Cryptanalysis

Cryptanalysis refers to the art and science of analyzing information systems in order to study the hidden aspects of the systems [2]. Cryptanalysis is used to breach cryptographic security systems and gain access to the contents of encrypted messages, even if the cryptographic key is unknown.

In addition to mathematical analysis of cryptographic algorithms, cryptanalysis also includes the study of side-channel attacks that do not target weaknesses in the cryptographic algorithms themselves, but instead exploit weaknesses in their implementation. Even though the goal has been the same, the methods and techniques of cryptanalysis have changed drastically through the history of cryptography, adapting to increasing cryptographic complexity, ranging from the pen-and-paper methods of the past, through machines like Bombes and Colossus computers at Bletchley Park in World War II, to the mathematically advanced computerized schemes of the present. Methods for breaking modern cryptosystems often involve solving carefully constructed problems in pure mathematics, the best-known being integer factorization.

III ELLIPTIC CURVES

In mathematics, an elliptic curve (EC) is a smooth, projective algebraic curve of genus one, on which there is a specified point O. An elliptic curve is in fact an abelian variety – i.e., it has a multiplication defined algebraically, with respect to which it is a (necessarily commutative) group – and O serves as the identity element. Often the curve itself, without O specified, is called an elliptic curve. Any elliptic curve can be written as a plane algebraic curve defined by an equation of the form:

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

where a and b are real numbers. This type of equation is called a Weierstrass equation. The definition of elliptic curve also requires that the curve be non-singular; i.e., its graph has no cusps or self-intersections. The point O is actually the "point at infinity" in the projective plane.

If $y^2 = P(x)$, where P is any polynomial of degree three in x with no repeated roots, then we obtain a nonsingular plane curve of genus one, which is thus also an elliptic curve. If P has degree four and is squarefree this equation again describes a plane curve of genus one; however, it has no natural choice of identity element. More generally, any algebraic curve of genus one, for example from the intersection of two quadric surfaces embedded in three-dimensional projective space, is called an elliptic curve, provided that it has at least one rational point.

Elliptic curves are especially important in number theory, and constitute a major area of current research; for example, they were used in the proof, by Andrew Wiles, of Fermat's Last Theorem. They also find applications in elliptic curve cryptography (ECC) and integer factorization.

3.1 Elliptic Curve Cryptography

Elliptic curve cryptography (ECC) is an approach to public-key cryptography based on the algebraic structure of elliptic curves over finite fields. Elliptic curves are also used in several integer factorization algorithms that have applications in cryptography, such as Lenstra elliptic curve factorization.

The use of elliptic curves in cryptography was suggested independently by Neal Koblitz [3] and Victor S. Miller [4] in 1985. Elliptic curve cryptography algorithms entered wide use in 2004 to 2005. The algorithm was approved by NIST in 2006. In 2013, the New York Times revealed that Dual Elliptic Curve Deterministic Random Bit Generation had been included as a NIST national standard due to the influence of NSA, which had included a deliberate weakness in the algorithm [5].

Public-key cryptography is based on the intractability of certain mathematical problems. Early public-key systems are secure assuming that it is difficult to factor a large integer composed of two or more large prime factors. For elliptic-curve-based protocols, it is assumed that finding the discrete logarithm of a random elliptic curve element with respect to a publicly known base point is infeasible –this is the "elliptic curve discrete logarithm problem" or ECDLP. The entire security of ECC depends on the ability to compute a point multiplication and the inability to compute the multiplicand given the original and product points. The size of the elliptic curve determines the difficulty of the

problem. For current cryptographic purposes, an elliptic curve is a plane curve which consists of the points satisfying the equation, $y^2 = x^3 + ax + b$ along with a distinguished point at infinity.

3.2 Elliptic Curve Cryptographic Algorithm

Elliptic Curve Cryptography (ECC) is an alternative mechanism for implementing public-key cryptography. The equation of an elliptic curve is given as:

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

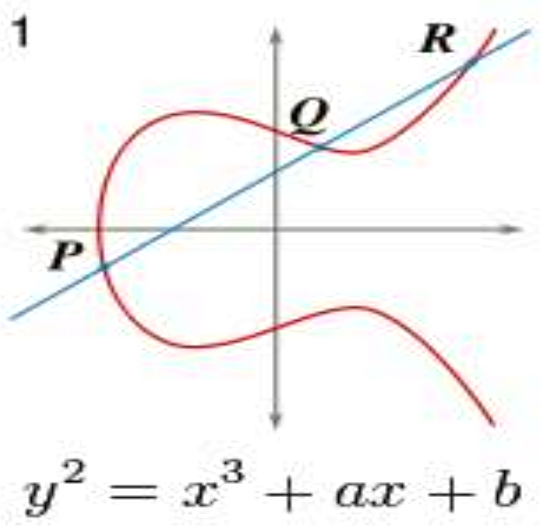

Fig. 1 Elliptic curve

3.2.1 Key Generation

Key generation is an important part where we have to generate both public key and private key. The sender will be encrypting the message with receiver's public key and the receiver will decrypt its private key.

Select a number 'd' within the range of 'n', where n is called the modulus. Now, we can generate the public key:

$$Q = d * P \quad (3)$$

d = The random number that we have selected within the range of (1 to n-1). P is the point on the curve.

'Q' is the public key and 'd' is the private key.

3.2.2 Message Encryption

Let 'm' be the message that we are sending. We have to represent this message on the curve. This has in-depth implementation details. All the advance research on ECC is done by a company called certicom.

Consider *'m'* has the point *'M'* on the curve *'E'*. Randomly select 'k' from [1 to n-1].

Two cipher texts will be generated, let it be C_1 and C_2.

$$C_1 = k * P$$

$$C_2 = M + k * Q$$

C_1 and C_2 will be send.

3.2.3 Message Decryption

We have to get back the message 'm' that was send to us,

$$M = C_2 – d * C_1 \quad (4)$$

M is the original message that we have send.

How do we get back the message, $M = C_2 – d * C_1$

(Since, $C_2 = M + k * Q$ and $C_1 = k * P$)

Therefore, $M = C_2 – d * C_1 = (M + k * Q) – d * (k * P)$

$= M + k * d * P – d * k * P \quad (Q = d * P)$

$= M$ (Original Message)

IV RSA CRYPTOSYSTEM

The various observations just stated form the basis for the RSA public-key cryptosystem, which was invented at MIT in 1977 by Ronald Rivest, Adi Shamir and Leonard Adleman. The public key in this cryptosystem consists of the value n, which is called the modulus, and the value e, which is called the public exponent. The private key consists of the modulus n and the value d, which is called the private exponent.

An RSA public-key / private-key pair can be generated by the following steps [6]:

Step-1: Generate a pair of large, random primes p and q.

Step-2: Compute the modulus n as n = pq.

Step-3: Select an odd public exponent e between 3 and n-1 that is relatively prime to p-1 and q-1.

Step-4: Compute the private exponent d from e, p and q.

Step-5: Output (n, e) as the public key and (n, d) as the private key.

The encryption operation in the RSA cryptosystem is exponentiation to the e^{th} power modulo n:

$$c = \text{ENCRYPT}(m) = m^e \bmod n. \quad (5)$$

The input m is the message; the output c is the resulting ciphertext. In practice, the message m is typically some kind of appropriately formatted key to be shared. The actual message is encrypted with the shared key using a traditional encryption algorithm. This construction makes it possible to encrypt a message of any length with only one exponentiation. The decryption operation is exponentiation to the d^{th} power modulo n:

$$m = \text{DECRYPT}(c) = c^d \bmod n. \quad (6)$$

The relationship between the exponents e and d ensures that encryption and decryption are inverses, so that the decryption operation recovers the original message m. Without the private key (n, d) (or equivalently the prime factors p and q), it's difficult to recover m from c. Consequently, n and e can be made public without compromising security, which is the basic requirement for a public-key cryptosystem.

The fact that the encryption and decryption operations are inverses and operate on the same set of inputs also means that the operations can be employed in reverse order to obtain a digital signature scheme following Diffie and Hellman's model. A message can be digitally signed by applying the decryption operation to it, i.e., by exponentiating it to the d^{th} power:

$$s = \text{SIGN}(m) = m^d \bmod n. \quad (7)$$

The digital signature can then be verified by applying the encryption operation to it and comparing the result with and/or recovering the message:

$$m = \text{VERIFY}(s) = s^e \bmod n. \quad (8)$$

V COMPARISON OF ECC WITH RSA

The primary benefit promised by ECC is a smaller key size, reducing storage and transmission requirements, i.e. that an elliptic curve group could provide the same level of security afforded by an RSA-based system with a large modulus and correspondingly larger key. RSA takes sub-exponential time and ECC takes full exponential time.

Therefore, ECC offers same level of security with smaller key sizes. DATA size for RSA is smaller than ECC. Encrypted message is a function of key size and data size for both RSA and ECC. Since ECC key size is relatively smaller than RSA key size, encrypted message in ECC is smaller. As a result, computational power is smaller for ECC. The table shows the comparison of ECC with RSA in terms of Key length, Key generation, Signature generation and Signature verification performance.

Table Comparison of ECC with RSA [7]

Symmetric Algorithms	Key Size (bits)		Key Generation Performance (Time)(s)		Signature Generation Performance (Time)(s)		Signature Verification Performance (Time)(s)	
	ECC	RSA	ECC	RSA	ECC	RSA	ECC	RSA

80	163	1024	0.08	0.16	0.15	0.01	0.23	0.01
112	233	2240	0.18	7.47	0.34	0.15	0.51	0.01
128	283	3072	0.27	9.80	0.59	0.21	0.86	0.01
192	409	7680	0.64	133.90	1.18	1.53	1.80	0.01
256	571	15360	1.44	679.06	3.07	9.20	4.53	0.03

5.1 Key Length Comparison of ECC With RSA

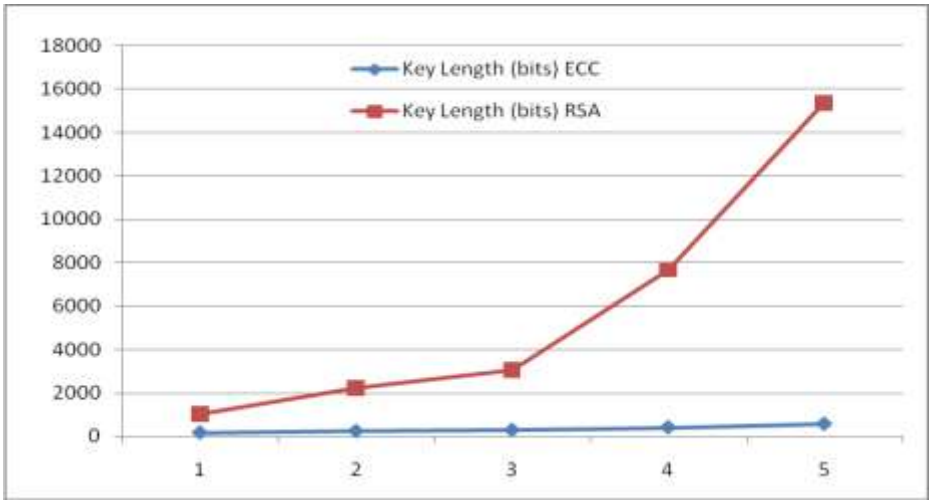

Fig. 2 Key length comparison of ECC with RSA

5.2 Key Generation Performance Comparison of ECC With RSA

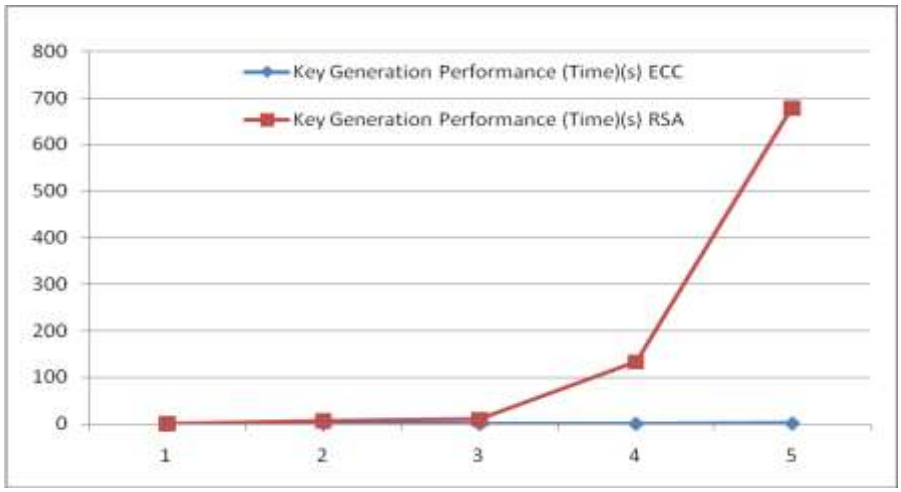

Fig. 3 Key generation performance comparison of ECC with RSA

5.3 Signature Generation Performance Comparison of ECC With RSA

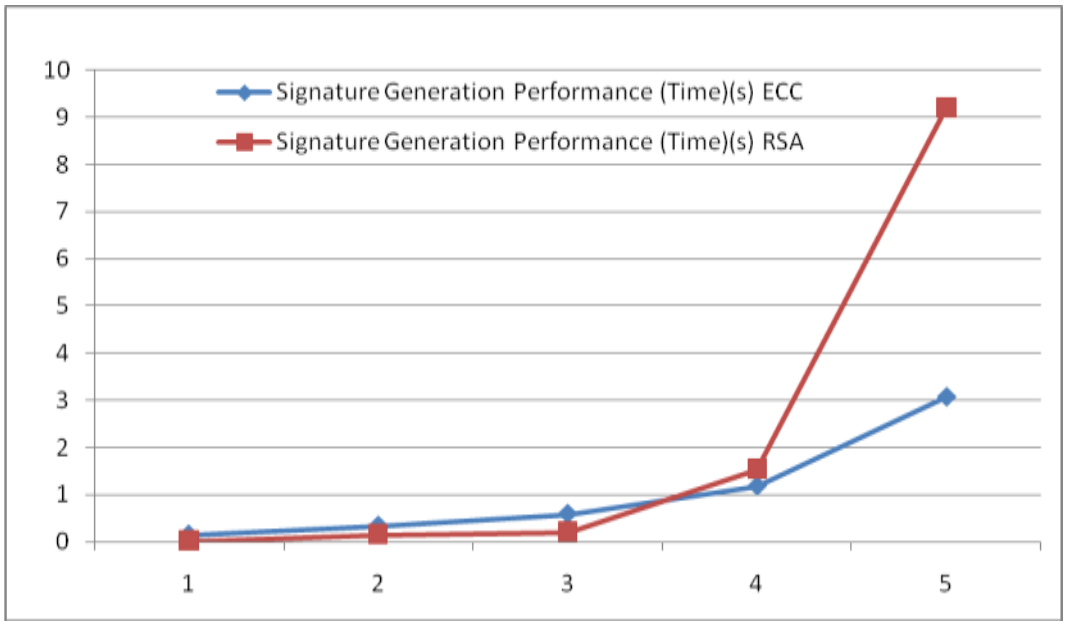

Fig. 4 Signature generation performance comparison of ECC with RSA

5.4 Signature Verification Performance Comparison of Ecc with RSA

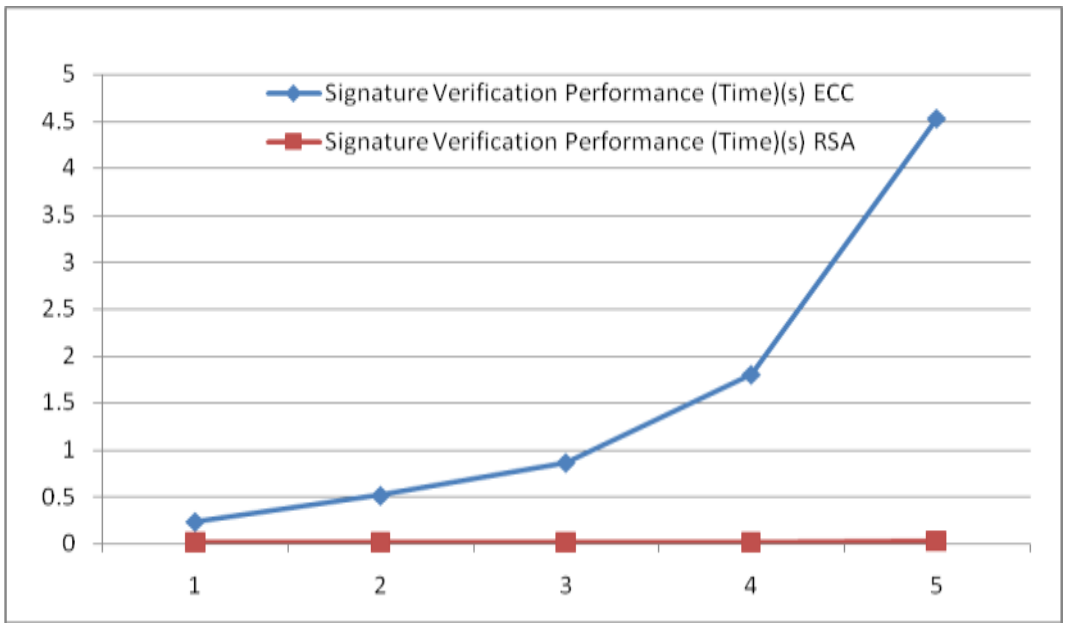

Fig. 5 Signature verification performance comparison of ECC with RSA

From the charts it is clear that ECC requires significantly smaller key size to afford the same level of security as RSA. Key generation for ECC outperforms RSA at all key lengths, and is especially apparent as the key length increases. Since ECC does not have to devote resources to the computationally intensive generation of prime numbers, ECC can create the private/public key pair in superior speed to RSA comparable lengths. ECC key generation time grows linearly with key size, while RSA grows exponentially. RSA key generation is significantly slower than ECC key generation for RSA key of sizes 1024 bits and greater. When the key size is less, the time to generate the signature in ECC is comparatively more than RSA but as the key length increases the key generation time taken by ECC is much less than that of RSA. Signature verification is where RSA pulls ahead of ECC in performance. The time to verify a message signed in RSA is negligible for the key lengths used, and does not even show a difference until you go from 7680 to 15360 bits. ECC lags behind in performance in every key length, showing nearly linear growth for increasing key sizes.

VI CONCLUSION

Elliptic curve cryptography offers the highest strength-per-key-bit of any known public-key system of first generation techniques like RSA. With less bits required to give the same security, ECC has fared favorably compared to RSA. While ECC may be comparatively difficult to understand for the layman, it is yet an important technology that has enormous potential to grow in the future.

VII FUTURE SCOPE

Many devices are constrained devices that have small and limited storage and computational power, for constrained devices ECC can be applied.

- For wireless communication devices like PDA's multimedia cellular phones ECC can apply.
- It can be used for security of smart cards, wireless sensor networks, wireless mesh and network detection.
- Web servers that need to handle many encryption sessions.
- Any kind of application where security is needed for our current cryptosystem.

REFERENCES

[1] Jonathan Katz , Yehuda Lindell , Introduction to Modern Cryptography: Principles and Protocols, CRC Press, 2008,page-3

[2] "Cryptanalysis/Signals Analysis". Nsa.gov. 2009-01-15. Retrieved 2013-04-15.

[3] Koblitz, N. (1987). "Elliptic curve cryptosystems". Mathematics of Computation 48 (177): 203–209. JSTOR 2007884.

[4] Miller, V. (1985). "Use of elliptic curves in cryptography". CRYPTO 85: 417–426. doi:10.1007/3-540-39799-X_31.

[5] Perlroth, Nicole (September 10, 2013). "Government Announces Steps to Restore Confidence on Encryption Standards". The New York Times. Retrieved September 11, 2013.

[6] B. Kaliski, “The Mathematics of the RSA Public-Key Cryptosystem”, RSA Laboratories.

[7] N. Jansma and B. Arrendondo,”Performance Comparison of Elliptic Curve and RSA Digital Signatures”, 2004.

[8] A. Lenstra, and E. Verheul, "Selecting Cryptographic Key Sizes", Journal of Cryptology 14 (2001) 255-293.