An approach to improve performance in Security using elliptic curve cryptography over RSA

Dileep Kumar Agarwal, Sudhir Rathi
Department of Computer Science, Sobhasaria Engineering college, Sikar, Rajasthan, India

Abstract— *Explosion of networks and the huge amount of data transmitted along with securing data content is becoming more important concern. Data encryption is widely used to ensure security in open networks such as the Internet. Development of cryptography research and computer technology, the capabilities of cryptosystems such as RSA and Diffie-Hellman are inadequate due to the requirement of large number of bits. The cryptosystem based on Elliptic Curve Cryptography is recently gained a lot of attention in industry. ECC Encryption and decryption methods can only encrypt and decrypt a point on the curve and not messages. The Encoding (converting message to a point) and Decoding (converting a point to a message) are important functions in encryption and decryption in ECC. This paper investigates the improvement of performance in security of text, images, audio based application using Elliptic Curve Cryptography (ECC) algorithm and evaluate the effectiveness of ECC over RSA algorithm. For the same level of security for best currently known attacks, elliptic curve based systems can be implemented with smaller parameters, leading to significant performance advantages. In the present study the performance advantages of elliptic curve systems are highlighted by comparing their performance with RSA based systems. The elliptic curves considered in the study are defined over Zp, called the prime curves and all the mathematical operations of ECC is defined over the elliptic curve* $y2 = x3 + Ax + B$, *where* $4A3 + 27B2 \neq 0$. *'A' and 'B' are called curve parameters and different values of these parameters define different elliptic curve. The public key here is a point (x, y) lying on 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. 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.*

Keywords— **ECC, RSA, DLP, ECDLP**

I. INTRODUCTION

Elliptic curves (EC) were suggested for cryptography by Victor Miller [1] and Neal Koblitz [2] in 1985 in the form of Elliptic Curve Cryptography (ECC). ECC follows Public Key encryption technique and the security provided is based on the hardness of Discrete Logarithm Problem (DLP) called Elliptic Curve Discrete Logarithm Problem (ECDLP). Given that $kP = Q$, where P, Q are the points on an elliptic curve and k is a scalar. According to ECDLP, if k is significantly large then it is unviable to calculate k when the values of P and Q are given. Here k is the discrete logarithm of Q, having base P. Similar to other Public Key encryption techniques, the security level of ECC also depends on the sizes of the keys used[8,9].

TABLE I, illustrates the key sizes used in ECC and RSA cryptosystems, where each row corresponds to the same level of security[3,4].

TABLE I. THE EQUIVALENT PUBLIC KEY SIZES FOR ECC AND RSA ACCORDING TO NIST

ECC key size(Bits)	RSA Key size(Bits)	Key Size Ratio
163	1024	1:6
256	3072	1:12
384	7680	1:20
512	15360	1:30

The advantage of ECC is that the inverse operation gets difficult to compute at a rapid phase, compared to the inverse operations in RSA for increase in the key size. Commercially 1024 bit RSA and 160 bit ECC systems are shown as offering nearly same security status. The smaller key size causes faster cryptographic computations and makes smaller software or smaller chip deployment possible. Thus ECC has a great potential to be used in environments with limited resources.

II. ECC OVERVIEW

The security provided by ECC is based on the elliptic curve parameters and the basic functionalities are discussed in this section.

A. ECC Basics

An elliptic curve E can be defined over a finite field *Fp* or finite field F_2m [6,11];

1) Elliptic Curve defined over Prime field E(Fp): Fp consists of integers modulo *p*, and having the integers in the range [0, *p*-1], where *p* is a large prime number. Elliptic curve over the prime field *Fp* is represented as:

$y^2 \bmod p \equiv x^3 + ax + b \bmod p$, where

$$4a^3 + 27b^2 \bmod p \neq 0.$$

The domain parameters for Elliptic curve over *Fp* can be represented as; *p, a, b, G, n* and *h.*

Where *p* is a large prime number, *a* and *b* are curve parameters, *G* is a point on the elliptic curve called a generator point (x_G, y_G), *n* is the order of the elliptic curve and *h* is the cofactor defined as;

h = (number of points on elliptic curve *E(Fp)*) /*n*.

The Prime field operations involve modular arithmetic consisting of the operations; addition, subtraction, multiplication, division, multiplicative inverse, and modulus. Prime field operations are more suitable in software implementations of ECC.

2) Elliptic Curve defined over Binary field E(F_2m): The elliptic curve on a binary field F_2m is represented as:

$$y^2 + xy = x^3 + ax^2 + b, \text{ where } b \neq 0.$$

The length of the integers of the finite field is limited to m bits. These numbers can be represented as binary polynomials with degree of $m - 1$.

The domain parameters for elliptic curve over F_2m are *m, f(x), a, b, G, n* and *h.* Where *f(x)* is the irreducible polynomial of degree *m*. *G* is a point on the elliptic curve called a generator point (x_G, y_G), *n* is the order of the elliptic curve and *h* is the cofactor where;

h = (number of points on elliptic curve *E*(F_2m)) /*n*.

The Binary field operations involve polynomial arithmetic which consists of the operations; addition, subtraction, multiplication, division, multiplicative inverse and finding irreducible polynomial. Binary field operations are more efficient in hardware implementation of ECC.

The domain parameters and other parameters must be mutually agreed upon by the two entities who wish to have a secure and trusted communication deploying ECC. The points, which lie on the elliptic curve are; a point at infinity and the points, which satisfy the Elliptic Curve equation.

B. ECC operations

ECC follows the group law and logarithm problem. From the ECDL problem it is evident that the major operation involved in ECC is point multiplication [5, 7, 8, 12].

i) Point Multiplication: Points *P* and *Q* lie on the elliptic curve such that *P* is multiplied with a scalar *k* to obtain the point *Q*.

$$kP = Q$$

The point multiplication operation involves series of point addition and point doubling operations. The point doubling and point addition methods are illustrated as follows:-

$$\text{If } k = 23, \text{ then } kP = 23 \cdot P$$
$$k = 2(2(2(2P) + P) + P) + P$$

The scalar which is used for point multiplication is chosen from the range [0, $n - 1$].

C. Performance of ECC

Its inverse operation gets harder, faster, against increasing key length than do the inverse operations in Diffie Hellman and RSA [3,4].

As security requirements become more stringent, and as processing power gets cheaper and more available, ECC becomes the more practical system for use. And as security requirements become more demanding, and processors become more powerful.

This keeps ECC implementations smaller and more efficient than other implementations. ECC can use a considerably shorter key and offer the same level of security as other asymmetric algorithms using much larger ones. Moreover, the gulf between ECC and its competitors in terms of key size required for a given level of security becomes dramatically more pronounced, at higher levels of security.

III. IMPLEMENTATION RESULTS

The Encoding and Decoding times are specific to the processor. The observations are recorded on a machine with 1GB RAM and 1.6 GHz processor speed on Win XP platform. The following table (table 1) shows CPU Times for Encoding and Decoding when implemented in MATLAB[7,8], and a character 'a' is being encrypted and decrypted for different domain parameters of elliptic curves.

p	a	b	CPU Times(Secs)	
			encoding	decoding
2011	9	7	0.0115	0.0045
4093	9	7	0.0139	0.0044
8191	10	17	0.0177	0.0072
16381	1	17	0.0328	0.0012
65521	7	29	0.0823	0.0036

The Execution time for encoding and decoding functions will not vary according to the value of a,b,p(domain parameters ECC).The execution time for encoding is different for different values of ECC domain parameters.

Now we consider an elliptic curve over a finite field associated with a prime number p = 2011 whose equation can be written as

$$\mathbf{y^2 \equiv x^3 + 4x + 20 \bmod 2011}$$

Where 4, 20 are two integers which satisfy $\mathbf{4(4^3) +27(20^2) \neq 0 \pmod{2011}}$.

The basic EC operations are point addition and point doubling defined on the curve $\mathbf{E_{2011}\ (4,20)}$.

We developed under Matlab an interface in which we implemented the algorithm RSA as well as the ECC.

The Encryption and decryption methods in ECC are designed to encode and decode a point on the curve and not the entire message. During encryption, each character in the message has to be converted into ascii code then the ascii code into points of the form (x, y) and then the points have to be encoded by mapping each of them with each point on the elliptic curve and then the entire encoded points have to be converted back to ascii code.

Encryption & decryption text through ECC

enter message: myy
Cx =1776 for m
Cx =1935 for y
Cx =1278 for y
plain_text1 =myy
t = 0.0448 sec

Encryption & decryption text through RSA

enter message:myy
cipher_text =2567161 for m
cipher_text =230569 for y
cipher_text = 230569 for y
plain_text1 =myy
t =0.0305 sec

Important facts extracted from implementation of Encryption & decryption of text through RSA & ECC

1. Encryption & decryption in RSA is more faster than ECC
2. ECC assigns different ciphertext value to each same character of message & RSA assigns same cipher text value to same characters of message. ECC provides better security through assigning same character to different values of ciphertext

Encrypting & decrypting audio based file

- **Encryption & decryption algorithm**

(i) First we take audio file as an input X.

(ii) Convert each value of audio file X into unsigned 8.

(iii) Each value of audio file X, that is called message **m**, can be converted into the coordinate $\mathbf{(X_m,Y_m)}$ that are the point on elliptic curve.

$$\mathbf{X_m = m*s + J,\ J= 0,1,2,3\ldots}$$

$$\mathbf{Y_m = (x^3 +ax +b)^{1/2}}$$

Where m is message**'s'** is the random positive integer. $\mathbf{(X_m,Y_m)}$ is a square modulo **P**, where **P** is the prime no. and $\mathbf{P \geq s*m}$.

(iv) Encryption and decryption system require a point $\mathbf{(X_1, Y_1)}$ on curve. Now User **A** chooses a secret integer **K** and computes $\mathbf{Q = K(X_1, Y_1)}$ (using point doubling). User B's public key consists of $\mathbf{E_{2011}\ (4, 20)}$ and the points $\mathbf{(X_1, Y_1)}$ and **Q**, while the integer **K** is kept private. To encrypt and send message$\mathbf{(X_m,Y_m)}$ to user B, user A choose a random positive integer **'r'** and produce the cipher text consisting of the $\mathbf{M_1}$ & $\mathbf{M_2}$.

$\mathbf{M_1= r(X_1, Y_1)}$

$$\mathbf{M_2= (X_m,Y_m) + r.\ Q}$$

(v) Decrypt the cipher text using the method

$$\mathbf{M_2\text{-}KM_1 = (X_m,Y_m)}$$

We transform the audio file into $\mathbf{(X_m,Y_m)}$ coordinate that is the point of elliptic curve and then encrypted.

Audio file-I encrpted and decrypted through ECC

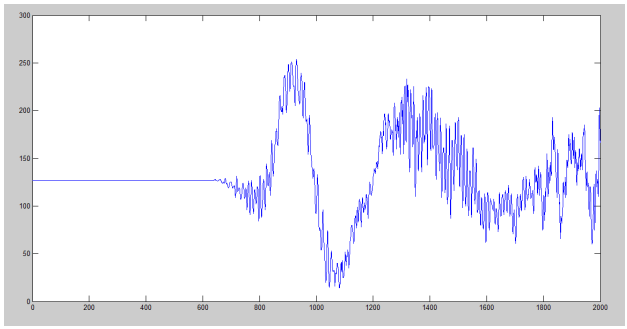

Fig. 1.1 Original audio file "loopy music.wav"

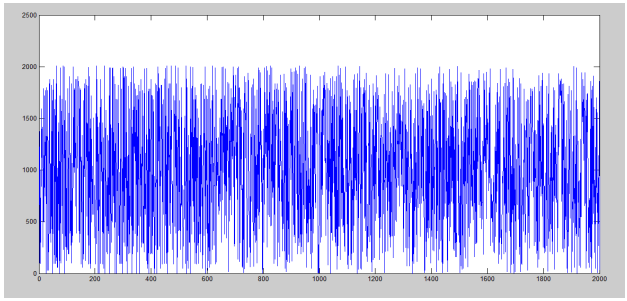

Fig 1.2 Encrypted audio file "loopy music.wav"

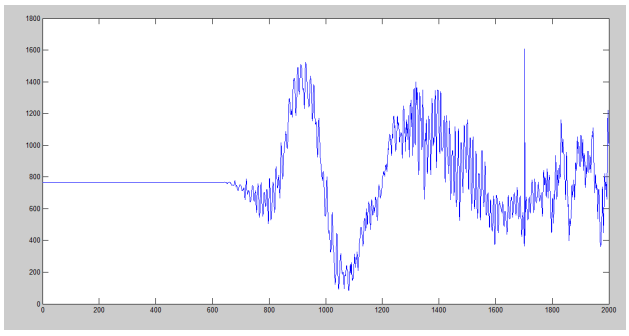

Fig. 1.3Decrypted audio file "loopy music.wav"

Audio file encrpted and decrypted through RSA

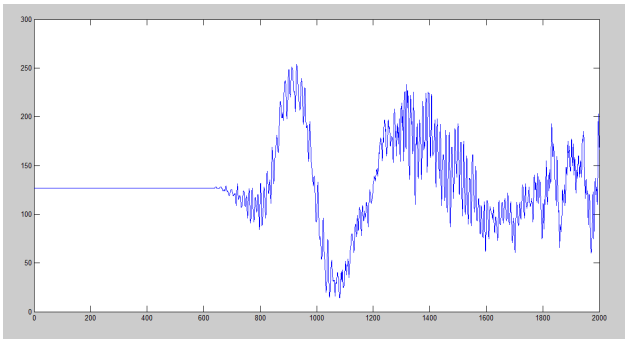

Fig 1.4 Original audio file "loopyMusic.wav"

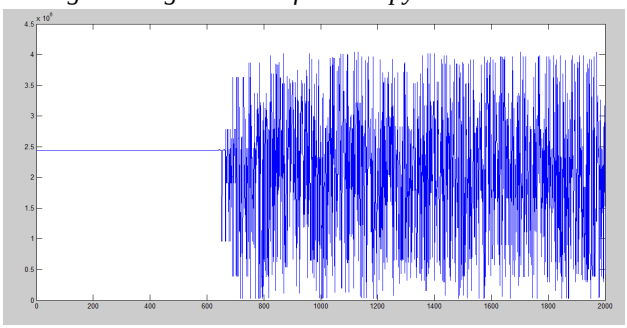

Fig 1.5 Encrpted audio file "loopy music.wav"

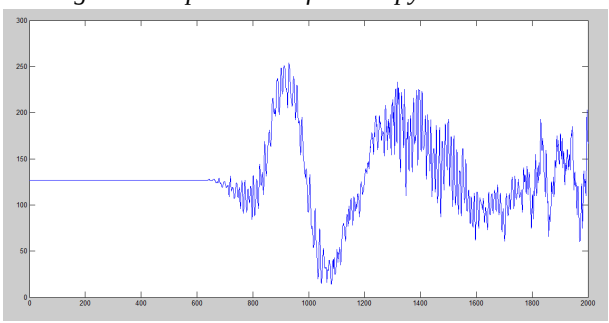

Fig 1.6 Decrpted audio file "loopy music.wav"

Audio file-II encrpted and decrypted through ECC

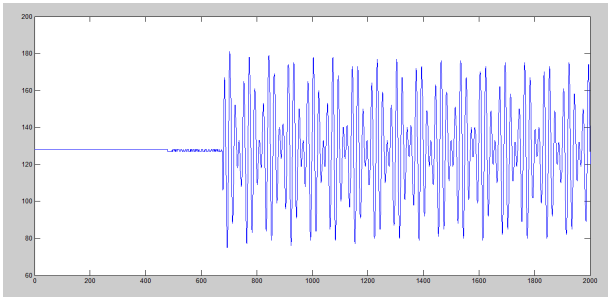

Fig 1.7 Original audio file "blip.wav"

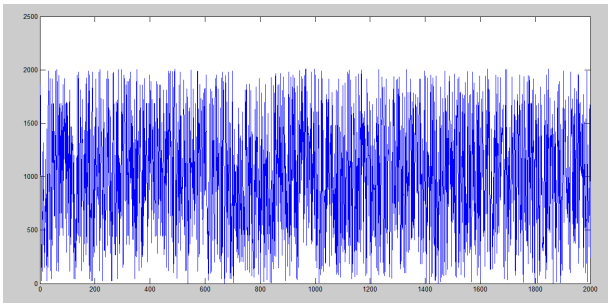

Fig 1.8 Encrypted audio file "blip.wav"

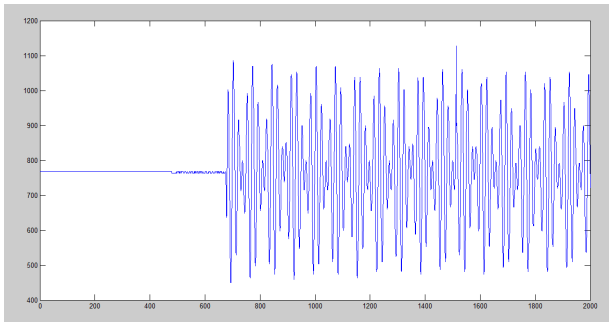

Fig 1.9 Decrypted audio file "blip.wav"

Audio file-II encrpted and decrypted through RSA

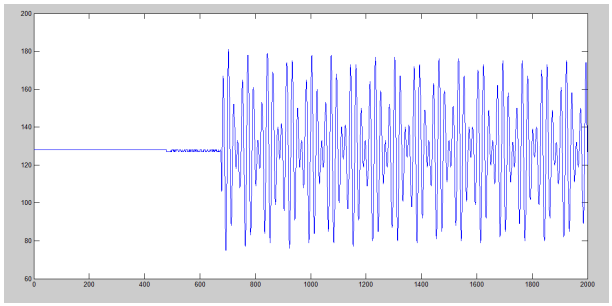

Fig 1.10 Original audio file"blip.wav"

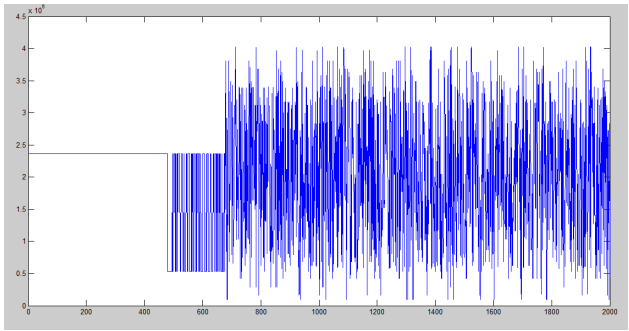

Fig 1.11 Encrypted audio file "blip.wav"

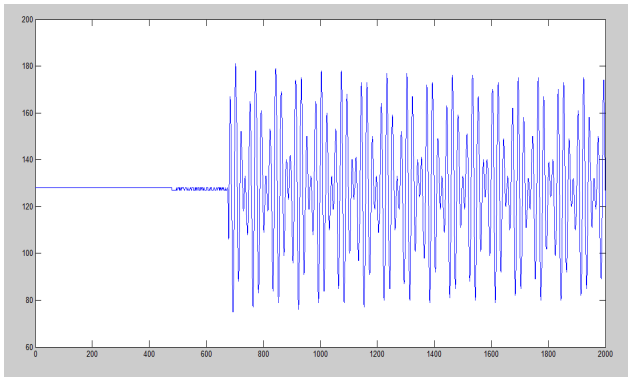

Fig 1.12 Decrypted audio file "blip.wav"

IV. IMPLEMENTATION ANALYSIS

In our comparison and analysis of the ciphertext, we compare the strength of the ECC & RSA, initially we have taken an audio file and then compare the results, we have found that RSA based algorithm comes up with same ciphertext values for the same frequency level as shown in fig.(1.5,1.11) however, when ECC has been operated on same audio file then it have found that for the same frequency level there will be different ciphertext values as because of random number generation for the same frequency level in the ECC algorithm. Based on above experiments, it can be concluded that ECC able to provides higher level of security in comparison to RSA.

V. CONCLUSION

ECC is a very encouraging and new field to work in order to find a more cost efficient method to perform encryption for portable devices and to secure image, audio, video transmission over internet. Elliptic curves are believed to provide good security with smaller key sizes, something that is very useful in many applications. The encryption and decryption of the text & audio are demonstrated. Also the work is extended to the image applications. Smaller key sizes may result in faster execution timings for the schemes, which is beneficial to systems where real time performance is a critical factor. We have estimated levels of security for RSA and ECC systems. This comparison illustrates the appeal of elliptic curve cryptography especially for applications that have high security.

From the implementation of RSA and ECC algorithms we now can conclude that operations in RSA are comparatively faster than ECC. In RSA key generation and encryption are faster whereas decryption is slower. On the other hand in ECC key generation and encryption are slower whereas the decryption is faster. From this conclusion RSA is faster but it is said that security wise ECC is stronger than RSA.

REFERENCES

[1] M.Prabu, Dr.R.Shanmugalakshmi, “A Comparative and Overview Analysis of Elliptic Curve Cryptography Over Finite Fields”, 2009 International Conference on Information and Multimedia Technology.

[2] R.K.Pateriya, Shreeja Vasudevan, “Elliptic Curve Cryptography in Constrained Environments: A review”, 2011 International Conference on Communication Systems and Network Technologies.

[3] Padma Bh, D.Chandravathi , P.Prapoorna Roja,”Encoding And Decoding of a Message in the Implementation of Elliptic Curve Cryptography using Koblitz’s Method” ,International Journal on Computer Science and Engineering Vol. 02, No. 05, 2010, 1904-1907.

[4] Zhi Li, John Higgins, Mark Clement,”Performance of Finite Field Arithmetic in an Elliptic Curve Cryptosystem” ,Advances in Cryptology-CRYPTO’99, LNCS 1716, pp.75-85

[5] ALFRED MENEZES, SCOTT VANSTONE,”The State of Elliptic Curve Cryptography” ,2000 Kluwer Academic Publishers, Boston

[6] Julian Lehmann ,”Fast Elliptic Curve Operation Costs and Comparison of Sole Inversion Precomputation Schemes” ,Proceedings of the 13th International World Wide Web Conference, ACM Press, 2006, pp. 202 -203.

[7] Nicholas Jansma, Brandon Arrendondo, ”Performance Comparison of Elliptic Curve and RSA Digital Signatures” ,IEEE 2009.

[8] Malek Jakob Kakish ,” authenticated and secure el-gamal cryptosystem over elliptic curves”, IJRRAS 10 (volume 2) ,February 2012.

[9] N. Koblitz, *“Introduction to Elliptic Curves and Modular Forms”*, 2nd edition, Springer-Verlag (1993).

[10] Williams Stallings, “Cryptography and Network Security”, Prentice Hall, 4th Edition, 2006.

[11] Darrel Hankerson, Alfred Menezes and Scott Vanstone, “Guide to Elliptic Curve Cryptography”, Springer-Verlag New York, 2004.

[12] Hankerson D, Menezes A, Vanstone S, “Guide to Elliptic Curve Crytography”, New York, USA:LNCS, Spring-Verlag, 2004.