International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 7, July - 2014

Analysis of Elliptic Curve Cryptography for Mobile Banking

Monali N. Shetty
Assistant Professor
Department of Computer Science
Fr. Conceicao Rodrigues College of Engineering
Mumbai, India

Tejas Puranik
B. E. Student
Department of Computer Science
Fr. Conceicao Rodrigues College of Engineering
Mumbai, India

Swati Jaybhaye
B.E. Student
Department of Computer Science
Fr. Conceicao Rodrigues College of Engineering
Mumbai, India

Swapnali Ghosalkar
B. E. Student
Department of Computer Science
Fr. Conceicao Rodrigues College of Engineering
Mumbai, India

***Abstract* - The tremendous increase in the use of mobile and wireless devices with limitations on power, bandwidth and low security postulates a new generation of Public Key Cryptography (PKC) schemes. We state Elliptic curve cryptography as a PKC scheme which is capable of fulfilling those requirements. Our paper examines the use of Elliptic Curve Cryptography (ECC) in such a constrained environment along with the other two aspects of ECC, namely its security and efficiency. In the paper, the performance of ECC is evaluated by comparing its different methods of implementation to find out the most efficient solution for mobile environment considering the constraints of battery life, processing power, memory, speed, bandwidth etc. The efficient method is then tested for mobile payment application. ECC encryption and decryption is implemented and tested on user module to check whether it is capable of handling all constraints and providing high security. The implementation is divided into two parts first, design of API for ECC (Elliptic Curve Cryptography) which generates shared secret key required for secure communication and performs encryption, decryption and secondly, mobile application which allows user to perform mobile banking with the help of ECC.**

Keywords: Elliptic Curve Cryptography, Analysis of ECC, Mobile Banking

I. INTRODUCTION

Elliptic Curve Cryptography is a public key Cryptography. Every user taking part in public key cryptography will take a pair of keys, a public key and a private key. The private key is known to only the authorised user whereas public keys are distributed to all users participating in communication. In ECC we will use some predefined constants which are known as 'Domain Parameters'. ECC is based on properties of a particular type of equation created from the mathematical group derived from points where the line intersects the axes. To generate the next point successive addition is performed but it is very difficult to find what number was used, even if you know the original point and the result. The heart of ECC is discrete logarithm problem that can be stated as "it should be very hard to find a value k such that Q=KP where P and Q are known'. But 'it should be relatively easy to find Q where k and P are known' P, Q are points on the elliptic curve. Equations based on elliptic curves have a characteristic that is very valuable for cryptography purposes: they are relatively easy to perform, and extremely difficult to reverse.ECC has various applications ,some of the major fields are virtual currency Bitcoin, Secure Shell(SSH) protocol , Transport Layer Security (TLS) protocol , Physical smart cards like Austrian e-ID.The reason behind choosing ECC for mobile payments is it overcomes the constraints like shorter key size, smaller signature length, low calculation , fast operation and high security working.

II. ELLIPIC CURVE CRYPTOGRAPHY

A. ECC Algorithm

An Elliptic curve equation is of the form

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

or

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

or

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

where **x** and **y** are variables, **a** and **b** are constants. However, these values are not necessarily real numbers; instead they

may be values from any field.

There are different methods of ECC implementation, the most common is ECC implementation over real numbers, but it has several problems-they cannot be stored precisely in a computer memory and it is not possible to predict the amount of storage required by them. Hence, we are considering an alternative approach of ECC implementation i.e. over finite field.

B. ECC implementation over finite field

In the cryptographic schemes, elliptic curves over two finite fields are mostly used.

i.Prime field Fp , where p is a prime

Elliptic Curve equation:

$$y^2 \bmod p = x^3 + ax + b \bmod p \quad (4)$$

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

Point Addition
If $P=(x_1, y_1)$, $Q=(x_2, y_2)$,
$R(x_3, y_3) = P+Q$ can be computed as
$x_3 = \lambda^2 - x_1 - x_2 \bmod p$
$y_3 = \lambda(x_1 - x3) - y1 \bmod p$
$\lambda = (y_2 - y_1)(x_2 - x_1)$ (5)

Point doubling
If $P = Q$
$x_3 = \lambda^2 - x_1 - x_2 \bmod p$
$y_3 = \lambda(x_1 - x3) - y1 \bmod p$
$\lambda = (3*x_1^2 + a)/2y_1$ (6)

ii.Binary field F2m, where m is a positive integer
Elliptic Curve equation:

$$y^2 + xy = x^3 + ax^2 + b \quad (7)$$

where $b \neq 0$

Here the elements of the finite field are integers of length at most m bits.

Point Addition
If $P=(x_1, y_1)$, $Q=(x_2, y_2)$,
$R(x_3, y_3) = P+Q$ can be computed as
$\lambda = (y_2+y_1)/(x_2+x_1)$
$x_3 = (\lambda^2) + \lambda + x_1 + x_2 + a$
$y_3 = \lambda(x_1 + x_3) + x_3 + y_1$ (8)

Point doubling
$P = (x_P, y_P)$, then $R = 2P = (x_R, y_R)$
$x_R = (\lambda * \lambda) + \lambda + a$
$y_P = x_R^2 (\lambda + 1) x_R$
where $\lambda = x_P + y_P / x_P$ (9)

C. Selecting an Appropriate Elliptic Curve

Conditions to be satisfied:

- $E(Fq)$ should be divisible by a sufficiently large prime, in order to resist against the Pollard ρ-attack.
- $E(Fq)$ should not to be equal to q, to avoid the Semaev-Smart-Satoh-Araki attack.
- To resist the MOV reduction attack, n should not divide qk-1 for all $1 \leq k \leq 30$

D. *Key Generation using Elliptic Curve Diffie Hellman Scheme (ECDH):*

1. Ephemeral key pair generation for Alice
   Select a private key $n_A \in [1, n-1]$
   Calculate public key $Q_A = n_A G$
2. Ephemeral key pair generation for Bob
   Select a private key $n_B \in [1, n-1]$
   Calculate public key $Q_B = n_B G$
3. Excange of Q_A and Q_B
4. Shared Key Computation
   $K = n_A Q_B$, $K = n_B Q_A$

K will be consistent as $K = n_A Q_B = n_A n_B P = n_B Q_A$

E. *Encryption*

Alice selects **P**, a point on the curve, as her plaintext, **P**. She then calculates a pair of points on the text as ciphertexts: C1=K * G, C2= P + K * Q_B .We may wonder how an arbitrary plaintext can be a point on the elliptic curve. This is one of the challenging issues in the use of the elliptic curve for simulation. Alice needs to use an algorithm to find a one-to-one correspondence between a block of text and the points on the curve.

Representation of a message to a point:
ECC cryptosystem deals with the points lying within the defined elliptic curve to perform operations such as key generation, encryption and decryption. Hence, prior to ECC encryption the plaintext input should be mapped to Elliptic curve points. Our implementation involves mapping of every plaintext character to the random point on the curve. One to one correspondence between block of text and curve points is maintained with the help of ASCII values of the plaintext characters.

Generating cipher text:
For every plaintext character, the corresponding curve point **P** is given to ECC encryption module and pair of points C1=K * G, C2= P + K * Q_B is calculated as cipher text. After generating the pair of points for entire plaintext block i.e. one by one for all the mapped plaintext points, the cipher text block of points is considered as C1 followed by C2 part of all the cipher text pairs. C1 is transmitted only once at the beginning as the resulting value of C1 will be same for all (C1, C2) pairs i.e. C1=K * G.

F. *Decryption*

Multiply the C1 by n_B i.e. (K G) *n_B and subtract the value from the second part of the received CT(C2) i.e.
P + K * Q_B - (K G) *n_B = P + K * $n_B G$ - (K G) *n_B = P(10)

After subtracting we get the original plain text point and the process is repeated for entire cipher text block. Then, the decrypted points are decoded to plaintext character, one by one to get the corresponding plain text message.

III. IMPLEMENTATION OF ALGORITHM

The example of ECC algorithm is explained with the snapshots of GUI developed for analysis of ECC.

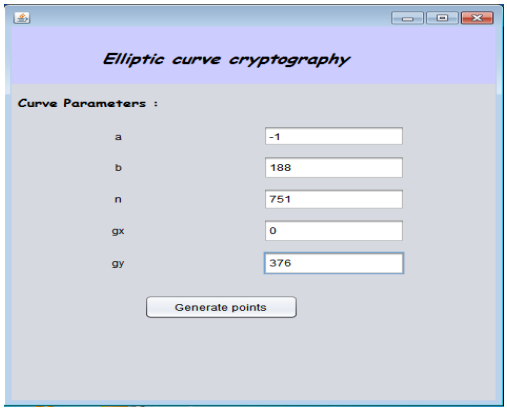

Fig.1. GUI of ECC

User will have to enter domain parameters such as a, b, n , g_x, g_y where g_x, g_y are co-ordinates of generator point, a and b are constants and n is the range of finite field.

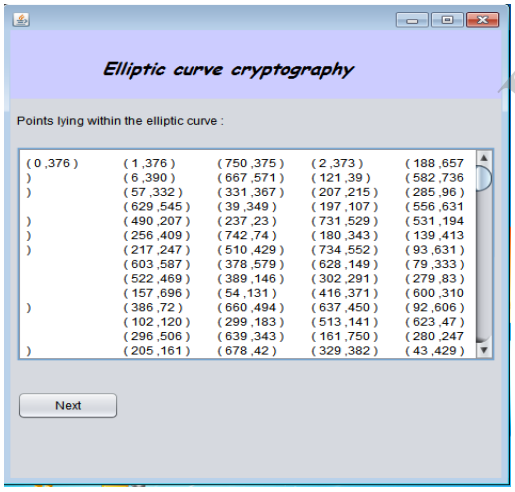

Fig. 2 . Finite points generation

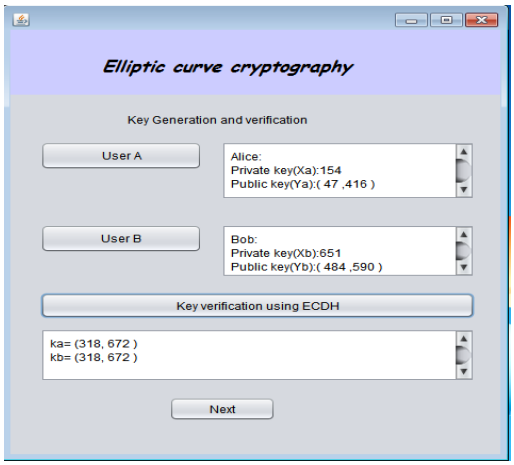

Fig. 3. Private and Public key generation and verification

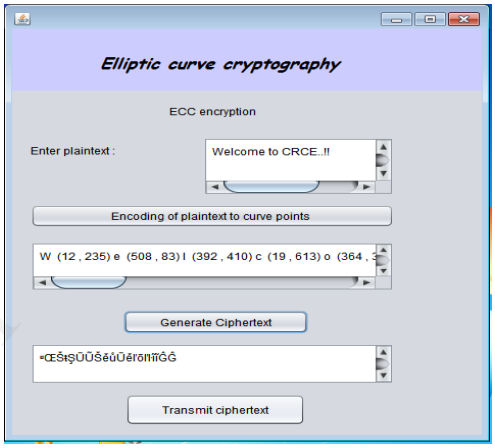

Fig. 4 . Encryption and Cipher text transmission

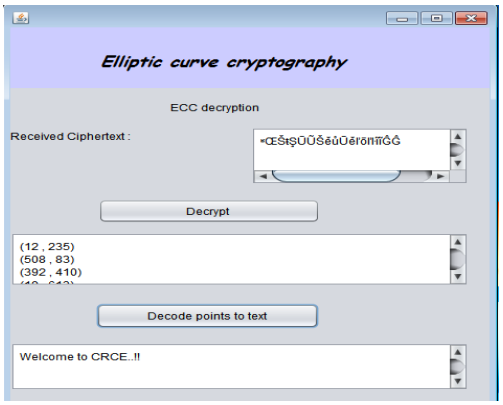

Fig. 5 . Decryption

IV. ANALYSIS OF ECC

A. *Analysis of different methods of ECC*

Over finite prime field:

It is suitable for s/w applications and for the processors having large multipliers for performing integer arithmetic. They do not need the extended bit-fiddling operations required by binary curves.

Over binary field:

Suitable for implementation of embedded systems and for h/w implementation as simply XOR and AND gates are needed to implement the whole system. Less no of logic gates as compared to prime field implementation are required.

B. *Security of ECC*

TABLE I. RSA AND ECC COMPARISON

TIME TO BREAK IN MIPS YEARS	RSA/DSA KEY SIZE	ECC KEY SIZE	RSA/ECC KEY SIZE RATIO
10^4	512	106	5:1
10^8	768	32	6:1
10^{11}	1024	163	7:1
10^{20}	2043	210	10:1
10^{78}	21000	600	35:1

The security of ECC depends upon how to calculate k when point is given in scalar multiplication. The security levels which is given by RSA can be provided by smaller keys of elliptic curve cryptosystem As compared to RSA, which offers 1024 bit security strength, ECC offers the same in 160 bit key length. Efficiency of ECC is depends upon factors such as computational outlay ,key size ,band width ,ECC provides higher-strength per- bit which include higher speeds, smaller power consumption, bandwidth reserves, storage efficiencies, and smaller certificates. For providing security mechanism will require fundamental basic security services such as authentication, confidentiality, non-repudiation and message integrity.

C. *Implementation of ECC cryptosystem for mobile banking*

We have integrated our code of ECC with j2me and it meets all the constraints required for mobile banking. For the first time registration user will have to visit bank's website where he will have to provide certain confidential information for getting his Mpin. Mpin is the security code which helps to provide authentication and confidentiality. At the same time keys required for encryption and decryption will be generated for the user.

The following snapshots give the overview of ECC equipped mobile banking module.

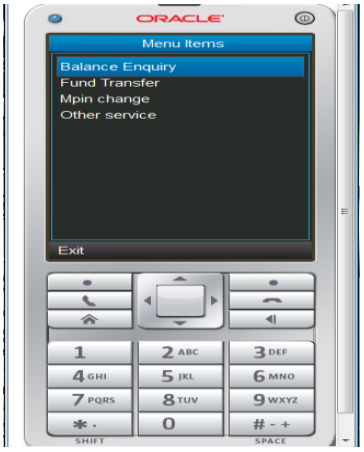

Fig. 6. M-Banking menu

Fig. 7. Mpin submission by user

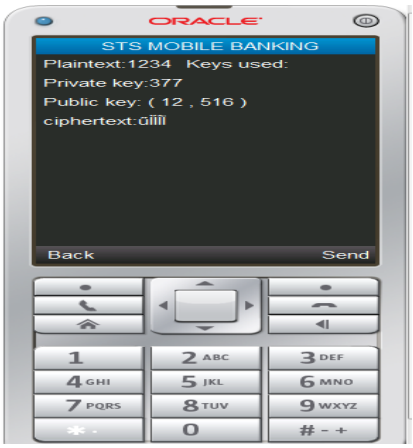

Fig. 8. CT generation for mpin

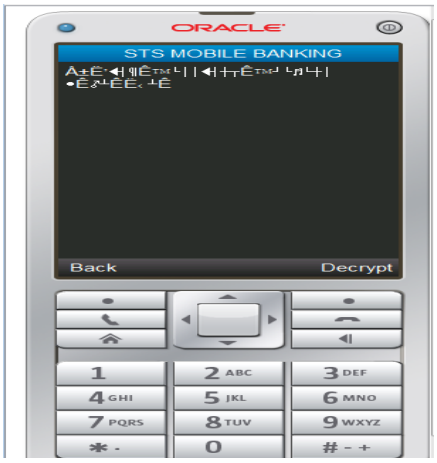

Fig. 9. Encrypetd server side response

Fig. 10. Actual (decrypted) response shown to the user

After deployment of the module, screens giving information about ECC parameters and encrypted responses will not be available to the user i.e. fig 8and 9 are not to be displayed to the user screen they are only needed from implementation point of view. Hence, the ECC equipped mobile banking approach is made transparent to the general user.

D. *Analysis over different key sizes*

Curve name: secp112r1

Plaintext: hello

Ciphertext: A ㅅㅁ ソ>ㅆ ㅈ &n卜 9P ㅆ vㅠ u

Ciphertext size: 20

Curve name: secp160r1

Plaintext: hello

Ciphertext: dㄴ 7Q사 |㉿jK朩㎜ ㅠ N"\ㅁ ㅅ사 & ツ 사 사

Ciphertext size: 26

Curve name: secp256r1

Plaintext: hello

Ciphertext: 〒ㅋ ㅅㅛ u6o卜t\: 卜 ㅁ Yㄴ £ 卜 X 5pa ₩* 카 8ㅛ ——————————|ㅆ 〒

Ciphertext size: 38

Result: As the key size increases the size of ciphertext also increases. So we can say that, as the key size increases the security also increases.

E. *Analysis for unique mapping of plaintext characters*

Curve name: secp112r1

Plaintext: hello

Ciphertext: T G=ㅆ X;㉿|ㅂ +ㅜ {ㅆ 刂夕 A

Curve name: secp112r1

Plaintext: hello

Ciphertext: t ∃]⁻ 亻- ㅜ ?ㅙ {시 ソ. ㅜ #L¥¥

Curve name: secp160r1

Plaintext: hello

Ciphertext: I□ニ}4 ㅐ _□¥bW㎜ ㅌㅋ _ w□s¯ㅂ ¢,ㅠ

Curve name: secp160r1

Plaintext: hello

Ciphertext: dㄴ 7Q사 |㉿jK朩㎜ ㅠ N"\ㅁ ㅅ사 & ツ사 사

Curve name: secp256r1

Plaintext: hello

Ciphertext: 〒ㅋ ㅅㅛ u6o卜t\: 卜 ㅁ Yㄴ £ 卜 X 5pa ₩*카 8ㅛ ——————————|ㅆ

Curve name: secp256r1

Plaintext: hello

Ciphertext: ㅁ 刂ㅙ ㅚ _cR카 f_ㅓ □ㅈㅓ ツㅐ ㅁ 刂 cR카 f_ㅓ □ㅈㅓ ツㅐ R<ツㅁ ₩O X,¯¯w

Result: Each time different ciphertext is generated, even though plaintext and key size remains same. So it is clear that each character is uniquely mapped to different points at fferent times.

V. CONCLUSION

Our Paper studied the different methods of implementing the ECC. We implemented API for ECC which performed key generation, encryption and decryption. This API is then integrated with j2me to check whether ECC is suitable for mobile banking. As the security of the proposed system is very hard, it is very clear that the proposed Mobile Banking using ECC will dominate banking sector in India. It has been mentioned in many literatures that a considerably smaller key size can be used for ECC compared to RSA. Also mathematical calculations required by elliptic curve cryptosystem are easier, hence, require a low calculation power.

REFERENCES

[1] T K Mohanta, R K Samantaray, R P Panda." Public Key Cryptography for mobile payment". *Researcher 2013;5(5):9*-13]. (ISSN: 1553-9865).

[2] Prof.Avinash Wadhe, Miss Namrata A.Sable "Mobile SMS Banking Security Using Elliptic Curve Cryptosystem In Binary Field" International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622

[3] Vorugunti Chandra Sekhar1, Mrudula Sarvabhatla. "A Secure Account Based Mobile Payment Protocol with Public Key Cryptography" published in "ACEEE International Journal Network Security 3, 1 (2012)

[4] G.N.Purohit, Asmita Singh Rawat "Efficient Implementation of Arithmetic Operations in ECC over Binary Fields" published in International Journal of Computer Applications (0975 – 8887) Volume 6– No.2, September 2010

[5] Darrel Hankerson1, Julio Lopez Hernandez, and Alfred Menezes "Software Implementation of Elliptic Curve Cryptography over Binary Fields" published in Springer

[6] Sattar J Aboud "Public Key Cryptography for mobile payment" Information Technology Advisor,Iraqi Council of Representatives,Iraq-Baghdad

[7] Ajay Kakkar, M. L. Singh, P.K. Bansal "Comparison of Various Encryption Algorithms and Techniques for Secured Data Communication in Multinode Network" International Journal of Engineering and Technology Volume 2 No. 1, January, 2012

[8] "Study of Indian Banks Websites for Cyber Crime Safety Mechanism"– (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 2, No.10, 2011

[9]. Stefan Tillich and Johann Großsch¨adl „A Survey of Public-Key Cryptography on J2ME-Enabled Mobile Devices"

[10] Shivani Agarwal[1]*, Mitesh Khapra[1], Bernard Menezes[1] and Nirav Uchat[1] "Security Issues in Mobile Payment Systems" computer society of India