EXTENDING DATA SECURITY TO AVOID DATA FRAUDS ON MOBILE DEVICES

[1] Ansar Ahemad Shaikh, [2] Nilesh S Vani

[1]PG Student, [2]Assistant Professor
Computer Engg Department,
GF's Godavari College of Engineering, Jalgaon (India)

ABSTRACT

Mobile devices are storage of lot of sensitive business as well as personal data. If mobile is stolen then the security of whole data will be lost. It causes damage to intellectual property of company as well as of individual. Traditionally, public key cryptography algorithm like RSA is used for providing security. As RSA impose computational burden in terms of execution time, memory, power and bandwidth, RSA can be replaced with Elliptic Curve Cryptography (ECC). Mobile devices are constrained in terms of power, memory and bandwidth etc. In our paper, We have implemented ECC over binary field (2^m) and proved its results over RSA for mobile devices which are constrained in terms of power, memory and bandwidth etc. ECC is used for encryption of mobile's phone book contents, email addresses stored SMS contents and also scheduler information set in a mobile. We have reduced memory consumption by dynamically generating points of elliptic curve. As we are not encrypting text character by character, Size of cipher text gets reduced. Also cryptanalytic attack is avoided by avoiding repetition of patters.

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

I. INTRODUCTION

MOBILE applications are gaining unwanted attention from criminals and intruders [15][16]. The public key algorithms like RSA, DH, and ECC etc are available for providing security to such applications. Traditionally security was provided with cryptography algorithm like RSA. But RSA impose more burdens in terms of execution time and memory consumption. For devices like sensor, mobile or smart card it is very costly in terms of memory and power to use RSA[6][7].ECC can be combined with digital watermarking as well as with Steganography for best security with privacy of data[5][13]. Emerging financial and banking mobile applications demand fast as well as more secure transactions [15][16]. Many authors have explained in literature how ECC is more useful than RSA.

In this paper, We have implemented ECC and proved with results how it is better than RSA because it takes less time, less key size, less power and less memory. This paper also focuses on implementing elliptic curve cryptography over binary field (2m) and using it for encryption of mobiles crucial data like phonebook contents, email addresses stored, SMS contents and also scheduler information set in a mobile. Finally ECC performance is compared with RSA. The implementation is checked on mobile operating system like Android.

In our paper, Section 2 gives theory of using elliptic curves for cryptography. Section 3 explains the work done with ECC in literature. Section 4 explains method used for implementation of ECC. In section 5, experimental results are given. And in section 6 I have concluded the paper.

II. ELLIPTIC CURVE THEORY

Elliptic Curve Cryptography (ECC) is a public key cryptography proposed by two scientists Kobitz and Miller [1][2]. The security of ECC is lies on discrete logarithm problem. It is a full exponential algorithm

which is very difficult to break. ECC gives same level of security that of RSA with less key size for example 160 bits key of ECC is equivalent with 1024 bits key of RSA. Elliptic Curve cryptography is based on elliptic curve equation. Any elliptic curve over binary field is given by equation[3]

y2+xy=x3+ax+b ……………………………….(I)

Here a, b are constants .User can change values of a and b to get different elliptic curve equations. These a and b should satisfy following constraint,

16a2+47b3 should not be equal to zero.

Any elliptic curve over prime field is given by equation [3]

y2=x3+ax+bmod p…………………………..(II)

Here a, b and p are constants. p is a prime number which is used for limiting number of points on the curve. More is the value of p, more points will be generated and more will be the security.

How this elliptic curve is used for cryptography is explained in section 4. Table 1 gives comparison between ECC and RSA which is taken from cert.com website [17] [18].

TABLE 1

Key Sizes In Bits With Equivalent Security Levels

Times to break in MIPS years	ECC	DH/DSA/ RSA	RSA/ECC Key Size Ratio
104	106	512	5:1
108	132	768	6:1
1011	160	1024	7:1
1020	210	2048	10:1
1078	600	21000	35:1

III. RELATED WORK

In literature many authors have tried to exhibit the advantages of ECC over RSA. B.Muthukumar, Dr. S.Jeevananthar explained Design of an Efficient Elliptic Curve Cryptography Coprocessor [8].It has explained point doubling, point addition and point multiplication operation .It is a hardware device and in my paper software implementation ECC is given. Multithreading Elliptic Curve Cryptosystem implemented by Uma S.Kanniah and Azman Samsudin from Universiti Sains Malaysia. They have used two parallel mathematical algorithms, Karatsuba and Montgomery for elliptic curve point multiplication[9]. But these two algorithms are complex to implement and slower for large number multiplication. For multithreading we can use java threads to reduce the complexity.

Concurrent Algorithm For High-speed Point Multiplication in Elliptic Curve Cryptography implemented by Jun-Hong Chen, Ming-Der Shieh and Chien-Ming Wu,Taiwan employed the nonadjacent form of a binary sequence to reduce the number of 1's in an operand so as to decrease the total number of addition in ECC encryption/decryption[10]. But It need an extra memory space to store an intermediate point, but it can achieve 100% hardware utilization. Memory space usage is reduced in my implementation.

Hai Yan and Zhijie Jerry Shi has given software implementation of ECC over 8-bit processor[11]. They have explained implementation on different word size processors.

IV. PROPOSED METHOD

The proposed model is given below

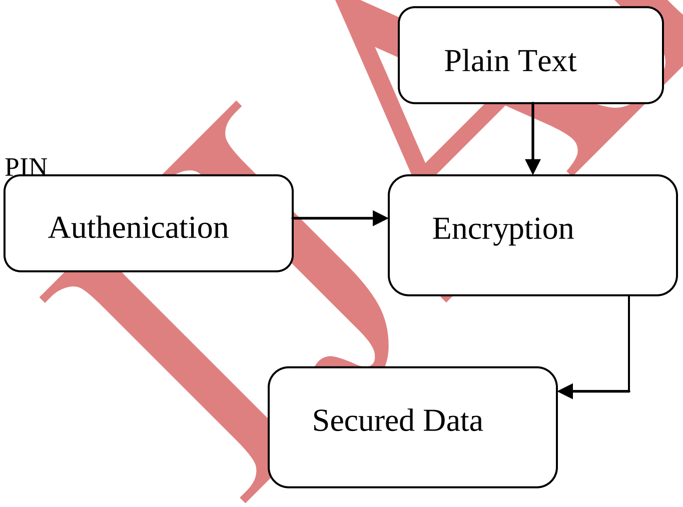

Figure. 1 Model for Mobile Security

In Fig 1, model for mobile security is shown. The user is authenticated with his PIN numbers. All the data on mobile internal memory as well as on storage card is stored in encrypted form. The data I am considering here includes Contact details, email addresses stored in mobile memory as well as SMS and scheduler contents.

For encryption as a key PIN number is used. Though mobile is stolen by somebody, then it is not possible to use or read data on that mobile without PIN. As mobile devices are resource constrained devices in terms of memory and power, RSA is not feasible for them. So, for encryption light weight elliptic curve cryptography is used.
In Fig 2. Development environment used for ECC implementation is given. ECC is implemented in Java with Eclipse IDE and executed on Android SDK emulator. The same jar and jad file can be executed on Windows Mobile also.

ECC Program

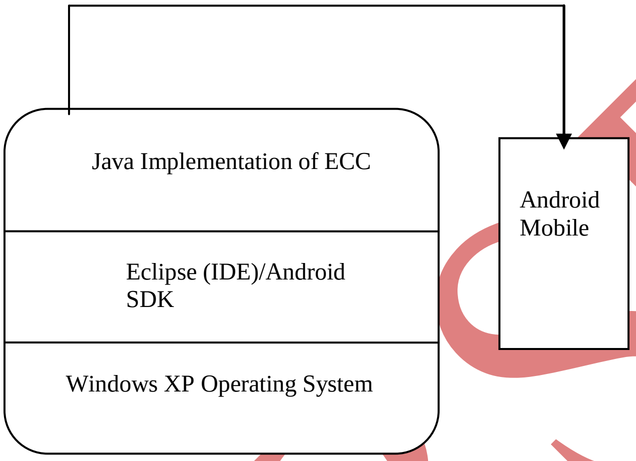
Java Implementation of ECC
Eclipse (IDE)/Android SDK
Windows XP Operating System
Android Mobile

Figure. 2 ECC Development environment for Mobile

In Elliptic Curve Cryptography, to convert plain text into cipher text, following steps are executed.

1. Convert Text into ASCII Format.
2. Generate points on Elliptic curve
3. Generate keys of Users
4. Encrypt Text

The detailed Elliptic Curve Cryptography Algorithm is given below. In this paper I modified the work proposed by [4] to ECC implementation with threads. Elliptic curve cryptography starts with generating points of a curve.

4.1 Generate points of a curve

```
Algorithm gen_points (a,b,p){
x=0
while (x<p){
Put values of a, b and x in equation y2+xy=x3+ax+b
Find roots of the equation y2+xy=x3+ax+b
//All values of(x,y) gives different points on elliptic curve.}}
```

4.2 Generate keys of a Mobile User

Suppose there are two users A and B. Following algorithm is used for generating keys.

```
    Algorithm Generate_keys_Mobile(){
Step 1:User A will select PIN number of mobile KA as a private key.
Step 2: Select generator point G from the curve points such that Point G is having small x and y coordinates.
Step 3: To generate public key kAp multiply KA with G using point_mult() algo.
```

4.3 Point Multiplication in ECC

To multiply any number K with point p(x,y) I repetitively apply point doubling and addition operations.

```
Algorithm Point_mult(){
For doubling a point(2p) use following formulae
S = [(3x 2 + a)/2yp] mod p
Then 2p has coordinates (XR, YR) given by:
XR = (S2 - 2 x) mod p
YR = [S (x - XR) - y] mod p
To determine 3P, I use  P +  2P, treating 2P=Q. Here P has coordinates (x,y), Q=2P has coordinates (XQ, yQ).
s=[(yQ-y)/(XQ-x)] / mod p
P+Q=-R
XR= (s2-x-XQ)  mod p
YR= (S(x-XR)-y) mod p}
```

4.4 Encryption on Mobile

1) Encryption of contact details, email addresses and scheduler contents

```
Algorithm Encrypt_Text(){
1. Access Contact details stored on mobile memory
2. Convert Contact details into its ASCII format
3. Select any point pm from generated points of a elliptic curve
4. Multiply ascii value with pm to get another point pm1 using Point_mult  algo
5. Cipher text will be {kG,pm1+k*kAp}
Repeat above steps for encryption of email addresses as well as scheduler contents stored on mobile memory
    }
```

2) Encryption of SMS contents

Whole SMS will not be encrypted .To reduce time of encryption, the confidential part of SMS only converted into encoded form.

```
    Algorithm Encrypt_ SMS ( ) {
Access SMS stored in Inbox Memory
Search sender's contact details in SMS using pattern search method.
Convert searched details into its ASCII format
Select any point pm from generated points of a elliptic curve
Multiply ASCII value with pm to get another point pm1 using Point_mult  algo
Cipher text will be {kG,pm1+k*kAp}
    }
```

4.5 Decrypting Text on mobile

```
Algorithm Decrypt_text_mobile (){
Take Cipher text will be {kG,pm1+k*kAp}
Calculate pm=pm1+k*kAp-kbkG}
```

The above implementation of ECC is shown below in Fig 3. First elliptic curve is selected by taking different values of a, b and m. Then points and keys are generated. For generation key users PIN number will is considered. Lastly using keys and points, data on mobile like contact numbers, SMS contents, Email Addresses and Scheduler information converted into cipher text using point multiplication method.

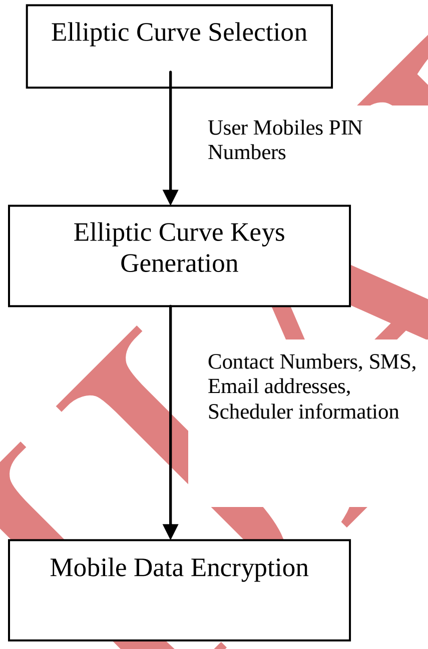
Elliptic Curve Selection
User Mobiles PIN Numbers
Elliptic Curve Keys Generation
Contact Numbers, SMS, Email addresses, Scheduler information
Mobile Data Encryption

Fig 3: Mobile Data Security Model

In above implementation of ECC, I have experimented some new things. After point generation of elliptic curve, instead of storing all points of the curve, in my implementation I have computed coordinates of the point whenever I need it. It saves the memory space required for ECC

We have converted whole plain text into ASCII value and then it is converted into point of a curve. The benefit of this method is avoiding repetition of cipher text block. So, cryptanalytic attack is not possible. Also size of cipher text is reduced resulting less storage space requirement which is beneficial for resource constrained devices like mobile.

V EXPERIMENTAL RESULTS

ECC implementation is done on GF(2m).ECC algorithm is implemented in Java. It is tested on Android which is one of the popular smart phone Operating system. In Fig 5 and Fig 6, ECC is compared with RSA by varying key size and data size respectively.In Fig 5 ,We can see RSA's excution time is less than ECC when key size is small. When key size become greater than 512 bits then RSA execution time increases than ECC. Security is increased by increasing number of bits in key size.

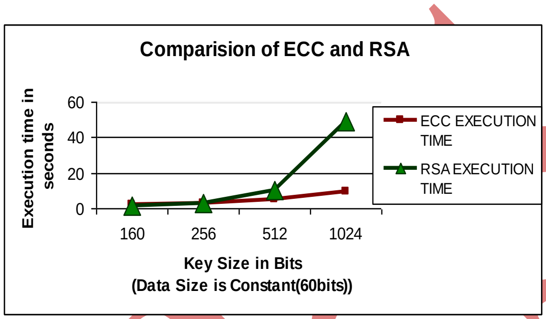

Fig. 5 Performance Comparison of ECC with RSA by varying Key Size on Android 2.1

In Fig.6, RSA's execution time is compared with ECC by varying Data Size.There is no greater effect on RSA if I increase Data size while ECC performance is exponential.

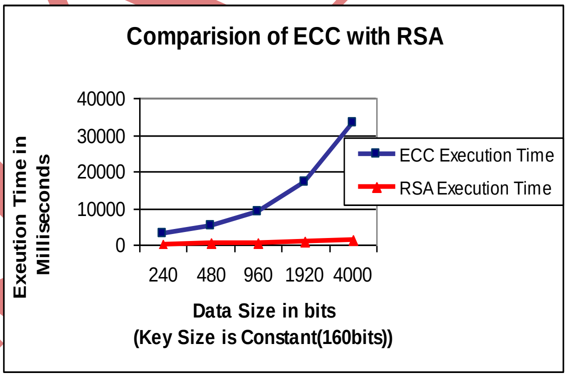

Fig.6. Performance Comparison of ECC with RSA by varying Data Size on Android 2.1

VI. DISCUSSION AND CONCLUSION

In this paper, I have implemented ECC in java for Android 2.1 SDK emulator. Machine configurations are Intel Core2 Duo CPU, 1.18GHz, 0.99GB RAM. Eclipse IDE is used for developing ECC in java. It is found that ECC takes less execution time than RSA when key size becomes greater than 512 bits. For comparison Equivalent key size of RSA and ECC is maintained as given in Table-I. In this ECC implementation, I have encrypted contact list ,SMS contents, Email Addresses .As scheduler information contains data related user's meetings which is crucial from business perspective, I encrypted scheduler information set by user I have reduced storage size required for storing points of elliptic curve by dynamically generating them. I avoided cryptanalytic attack by eliminating repetition of cipher text patterns. I reduced size of cipher text, so the storage required for this will be saved. Also the bandwidth required for transferring cipher text over the air will be reduced. As all data on mobile is stored in encoded form, data theft by intruder is not possible.

REFERENCES

[1] N. Koblitz, "Elliptic curve cryptosystems", Mathematics of Computation, vol. 48, pp. 203-209, 1987.

[2] V. Miller, "Uses of elliptic curves in cryptography",Advances in Cryptology: proceedings of Crypto'85, pp. 417-426, 1986.

[3] Hai Yan and Zhijie Jerry Sh, "Studying Software Implementations of Elliptic Curve Cryptography", IEEE 2006.

[4] MariaCelestin Vigila1, K. Muneeswaran's, "Implementation of Text based Cryptosystem using Elliptic Curve Cryptography", IEEE 2009.

[5] Hongbin Kong, Zhengquan Zeng, Lijun Yan, Jicheng Yang, Shaowen Yao,Nuoya Sheng, "Combine Elliptic Curve Cryptography with Digital Watermark for OWL Based Ontology Encryption", IEEE 2009.

[6] M.Aydos, T.Yanik and C.K.Kog, "High-speed implementation of an ECC based wireless authentication protocol on an ARM microprocessor,“ lEEE Proc Commun.,Vol. 148, No.5, pp. 273-279, October 2001.

[7] Kristin Lauter, "The Advantages of Elliptic Cryptography for Wireless Security", IEEE Wireless Communications, pp. 62- 67, Feb. 2006.

[8] B.Muthukumar, Dr. S.Jeevanantharr "Design of an Efficient Elliptic Curve Cryptography Coprocessor", IEEE 2009.

[9] Uma S.Kanniah and Azman Samsudin "Multithreading Elliptic Curve Cryptosystem",IEEE 2007.

[10] Jun-Hong Chen, Ming-Der Shieh and Chien-Ming Wu,Taiwan "Concurrent Algorithm For High-speed Point Multiplication In Elliptic Curve Cryptography", IEEE 2005.

[11] Hai Yan and Zhijie Jerry Shi "Software implementation of ECC over 8-bit processor", IEEE 2006.

[12] Tohari Ahmad1, Jiankun Hu2, Song Han "An Efficient Mobile Voting System Security Scheme based on Elliptic Curve Cryptography", IEEE 2009.

[13] Prof. B N Jagdale, Prof.R.K.Bedi and Sharmishta Desai, "Securing MMS with High Performance Elliptic Curve Cryptography", International Journal of Computer Applications 8(7):17–20, October 2010