Securing MMS with High Performance Elliptic Curve Cryptography

Prof. B. N. Jagdale
Assistant Professor,
IT Department, MIT COE,
Pune, India.

Prof.R.K.Bedi
Assistant Professor,
Comp Department, MIT COE,
Pune, India.

Sharmishta Desai
PG Student,
IT Department, MIT COE,
Pune, India.

ABSTRACT

Multimedia Messaging Service (MMS) is a new standard in mobile messaging. Like SMS, MMS is a way to send a message from one mobile to another. MMS can include not just text, but also sound, images and video. For making MMS secure, steganography can be used with it. Without having privacy of data there is no meaning of doing communication using extremely high end technologies like SMS or MMS. This can be achieved by using steganography, which is the process of hiding secret information inside some carrier. SMS and MMS are can be used as carrier for hiding information on mobile devices. For insisting more security, encrypted data will be hidden inside MMS. As mobile devices have less memory and less processing power, we cannot use computation intensive encryption algorithms like AES, DES, and RSA. Elliptic Curve Cryptography (ECC) is emerging as an attractive alternative to traditional public-key cryptosystems. ECC offers equivalent security with smaller key sizes resulting in faster computations, lower power consumption, as well as memory and bandwidth savings. In my paper, I have proposed a method of encrypting text with ECC and then hiding encrypted text in MMS. SMS are limited to 160 character messages while MMS has no size limit. Biggest use of MMS is likely to be for companies for sending MMS messages to subscribers, enquirers or customers or for banks for sending secret information like PINS/Passwords etc. The computational burden of ECC can be minimized by executing ECC with multiple threads.

Keywords

Elliptic Curve Cryptography (ECC), Digital Signature Algorithm (DSA), Diffiee-Hellman (DH), POSIX (Portable Operating System for UNIX), Multimedia Messaging Service (MMS)

1. INTRODUCTION

Steganography is a technology of hiding messages inside some harmless carriers to shelter the communication so that the outsiders may not discover the existence of information in the carrier. This is the major distinction between steganography and other methods of hidden exchange of information. For example, in cryptography method, people become aware of the existence of information by observing coded information, although they will be unable to comprehend the information. However, in steganography, nobody will understand the existence of information in the resources. MMS stenography is a combination of image and text steganography.

Elliptic Curve Cryptography (ECC) is emerging as an attractive public-key cryptosystem for mobile/wireless environments. Compared to traditional cryptosystems like RSA, DH, ECC offers equivalent security with smaller key sizes, which results in faster computations; lower power consumption, as well as memory and bandwidth savings. This is especially useful for mobile devices which are typically limited in terms of their CPU, power and network connectivity. This paper proposes a method of combing steganography with elliptic curve cryptography so that no one can recognize existence of information. If anyone able to recognize the existence of information, he/she cannot read that information as it is in encrypted form. As mobile devices have less memory and processing power, I have proposed a method of creating multiple threads for performing multiple ECC operations. Multithreading allows parallel implementation of ECC.

2. ELLIPTIC CURVE THEORY

Elliptic Curve Cryptography (ECC) 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 where as 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 [7].

An elliptic curve over a field K is a curve defined by an equation of the form

$y2 = x3 + ax + b$,

where $a, b \in K$ and $4a3 + 27b2 \neq 0$.

The condition that $(4a3 + 27b2) \neq 0$ implies that the curve has no "singular points", which will be essential for the applications we have in mind. 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.

An elliptic curve over a prime field is defined as follows,

$y2 \bmod p=(x3+ax+b) \bmod p$

Here, p is any random large prime number.

3. ELLIPTIC CURVE CRYPTOGRAPHY

Elliptic curve cryptography starts with generating points of a curve [3].

3.1 Generate points of a curve:-

```
Algorithm gen_points(a,b,p)
{
x=0
while(x<p)
{
find res=(x3+ax+b)
Find different values of y2 whose mod with p is equal to res
Find square root of y;
Finally, all values of (x, y) gives different points on elliptic curve.
}
}
```

3.2 Generate keys of a user

Suppose there are two users A and B.

Following algorithm is used for generating keys.

```
Algorithm Generate_keys()
{
User A will select any random number KA as a private key.
Select generator point G(point having small x and y coordinates) from the curve points.
To generate public key kAp multiply KA with G using point_mult() algo.
Follow above steps to generate keys(kB, kBp) for user B.
}
```

3.3 Point Multiplication in ECC

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

```
Algorithm Point_mult()
{
For doubling a point(2p) use following fomulae
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, we 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
```

3.4 Encrypting Text

```
Algorithm Encrypt_Text()
{
Convert character in a text 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}
}
```

3.5 Decrypting Text

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

4. PARALLEL IMPLEMENTATION OF ECC

Threads use and exist within process resources, and are able to be scheduled by the operating system and run as independent entities largely because they duplicate only the bare essential resources that enable them to exist as executable code.

The program performance increases when it contains multiple threads. Threads can be used to achieve parallelism.

ECC contains many parallel operations which can be executed separately by separate threads.

Separate threads can be called for generating subset of points of a curve as given below.

```
Point.x=0;
Point.p=17;
pthread_create (&thread1, NULL, gen_points (void*) point);
pthread_create (&thread2, NULL, Generate keys (void*) point);
Point.x=18;
Point.p=37;
pthread_create (&thread3, NULL, gen_points (void*) point);
```

After creating thread1, call gen_keys () method with second thread. Also the point multiplication and text encryption can be done with multiple threads .It increases the performance of ECC.

As we are not creating separate processes, the address space will be shared and the switching time from one process address space to another process address space is minimized. Less time is required for terminating and creating a thread than process. Communication overhead is less between threads than processes. So multiple processors can be used efficiently.

5. MMS STEGANOGRAPHY

MMS is a combination of text and image. We can hide part of data in image and part of data in text [6]. Image may be a colored or black and white image. LSB algorithm can be used for hiding data inside an image. LSB is a simple method used for hiding data inside 24bit images [8]. Each pixel of an image can be represented by three colors Red, Green and Blue. In 24bit image, there can be 2^24 colors i.e. 16.7 million colors.

5.1 Hiding information in Image

Divide MMS image into 3 X 3 blocks.

Hide each bit of text into least significant bit of each pixel.

Select pixel for hiding data using password which will be shared by sender and receiver.

Suppose that we have three adjacent pixels (nine bytes). Its RGB encoding is as follows:

10010101	00001100	11001001
10010111	00001110	11001011
10011111	00010000	11001011

Now suppose we want to "hide" the following 9 bits of data: 101101101. If we overlay these 9 bits over the LSB of the 9 bytes above, we get the following (where bits in bold have been changed).

1001010**1**	0000110**0**	1100100**1**
1001011**1**	0000111**0**	1100101**1**
1001111**1**	0001000**0**	1100101**1**

Only half of the least significant bits are actually changed.

So it does not make observable change in the image.

5.2. Extracting Information from image

Divide picture into 3 X 3 blocks.

Use the same password used by sender for selecting pixels.

Extract the data bits from the least significant bits of selected pixels.

5.3 Hiding Information in Text

For Hiding data in Text part of MMS use message acronyms as follows. A list of acronyms will be created and shared by sender and receiver.

For hiding data bits, list of acronyms is searched. For hiding ‘1’, full word will be passed and for hiding ‘0’ acronym will be used. Acronyms may be like ‘u’ for you or ‘4u’ for for you.

Some acronyms are given below:

ASAP-	As soon as possible
L8-	Late
2 L8-	Too Late
Engg.-	Engineering
Plz-	Please
TMW-	Tomorrow
Deptt. -	Department

6. PROPOSED METHOD

Before hiding message inside a MMS, encrypt a message with ECC with a method explained in section 3. For executing ECC use multithreading as explained in section 4.

As multithreading is used, we can execute ECC parallel which results in effective use of mobile’s limited resources like memory and processing power.

Then hide encrypted message in MMS using method explained in section 5. Hide some part of message in image part of MMS and remaining part in text part of MMS.

If by some way unauthorized person successfully extract the message from MMS .He/she will not read that message because it is encrypted with sender’s private key using ECC. Mobile devices have constrained environment. So ECC can be efficiently used as it takes less memory and less power with small key sizes.

My method imposes double security of data by encrypting it and then hiding its each bit in MMS.

7. DISCUSSION AND CONCLUSION

Banking or financial applications need to transfer information like PINS or passwords secretly or companies send MMS messages to subscribers or customers. MMS is a low cost and less bandwidth communication technique. MMS has no size limit .So we can hide any size of data inside MMS. For insisting more security, encrypted data can be hidden inside MMS.

As mobile devices have less memory and processing power, ECC can be used for encrypting messages on mobile. If ECC is implemented with multiple threads, then memory and time required will get reduced. If the message encrypted with ECC is hidden inside an MMS then security of that message will be increased.

ECC is useful not only in resource constrained environment like mobile, pager or smart card devices which have limited memory, limited processing capability and limited backup but also on powerful computers because it provides strong security with smaller key sizes. Crypto++ is an ECC library which provides cryptographic functions. POSIX library can be used for creating multiple threads. Stepic is python image steganography library. These two libraries can be used for combining ECC with steganography.

8. REFERENCES

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

[2] N.Koblitz, Elliptic Curve Cryptosystems, Mathematics of Computation, volA8, 1987, pp.203 -209.

[3] Implementation of Text based Cryptosystem using Elliptic Curve Cryptography s. Maria Celestin Vigila1 , K. Muneeswaran'

[4] Data Hiding in Binary Image for Authentication and Annotation Min Wu, Member, IEEE, and Bede Liu, Fellow, IEEE

[5] A Novel Scheme of Data Hiding in Binary Images IEEE, International Conference on Computational Intelligence and Multimedia Applications 2007

[6] Steganography in MMS, Mohammad Shirali-Shahreza, Sharif University of Technology, Tehran, IRAN.

[7] Elliptic Curve Cryptography, an Implementation Guide, Anoop MS.

[8] Image Based Steganography Using LSB Insertion Technique, M. S. Sutaone. IEEE.