Multiplication Based Elliptic Curve Encryption Scheme with Optimized Scalar Multiplication (MECES)

V.S.Shankar Sriram
Dept. Of Information Technology
Birla Institute of Technology
Mesra,Ranchi,India-835215

S.Dinesh
Dept. Of Computer Science & Engg.
Birla Institute of Technology
Mesra,Ranchi,India-835215

G.Sahoo
Dept. Of Information Technology
Birla Institute of Technology
Mesra,Ranchi,India-835215

ABSTRACT
Elliptic Curve Cryptography (ECC) fits well for an efficient and secure encryption scheme. It is efficient than the ubiquitous RSA based schemes because ECC utilizes smaller key sizes for equivalent security. This feature of ECC enables it to be applied to Wireless networks where there are constraints related to memory and computational power. The goal of this research is to develop an efficient method for Scalar Multiplication and to develop simple and efficient encryption scheme. In this paper we have compared the security of Elliptic curve AES (ECAES) with the Encryption scheme proposed by us. A comparative study of ECC with RSA is made in terms of key size, computational power, size of data files and encrypted files.

Categories and Subject Descriptors
D.4.6 [**Security and Protection**]: Access controls, Authentication, Cryptographic controls, Information flow controls

General Terms
Security

Keywords
Elliptic Curve Cryptography, Security

1. INTRODUCTION
Elliptic Curve Cryptography (ECC), proposed independently in 1985 by Neal Koblitz [1] and Victor Miller [2], has been used in cryptographic algorithms for a variety of security purposes such as key exchange and digital signatures. Compared to traditional integer-based public-key algorithms, ECC algorithms can achieve the same level of security with much shorter keys. For example, 160-bit Elliptic-curve Digital Signature Algorithm (ECDSA) has a security level equivalent to 1024-bit of Digital Signature Algorithm (DSA). Because of the shorter key length, ECC algorithms run faster, require less space, and consume less energy. These advantages make ECC a better choice of public-key cryptography, especially in resource constrained systems such as wireless and mobile devices for pervasive computing.

There are three families of public key algorithms that have considerable significance in current data security practice. They are integer factorization, discrete logarithm, and elliptic curve-based schemes. Integer factorization-based schemes such as RSA and Discrete Logarithm-based schemes such as Diffie-Hellman [13] provide intuitive ways of implementation. However, both methods admit of sub-exponential time for cryptanalysis [14]. Solving an ECDLP (Elliptic curve Discrete Logarithm Problem) takes full exponential time using pollard-rho method.

The most extended encryption/decryption scheme based on ECC is ECIES, proposed by Abdullah, Bellare and Rogaway in 1998 [2], being a variant of the El-Gamal scheme. ECIES can be found at ANSI X9.63, ISO/EC 15946-3 and IEEE 1363 standards. Comparison of ECC and RSA shows that security of ECC grows exponentially in its parameters, whereas the security of RSA grows only sub exponentially in its parameters from [5][6][7][8][9][10].

Since shorter key lengths translate into faster handshaking protocols, ECC is becoming increasingly important for wireless communications. [Source: Hank van Tilborg, NAW, 2001]. In wireless conditions, the equipment's resources like power, memory and computational capacity are limited. So the encryption scheme employed in it must consume limited resources. Most popular algorithms like RSA do not satisfy this. A comparison between ECC and RSA is shown in Table 1[1],[6]. It is clearly evident that ECC based Encryption/Decryption schemes are efficient in achieving security of applications that run on memory constrained

Table 1-Comparison of Key Sizes of RSA/DSA with ECC of Equal Security

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	132	6:1
10^{11}	1024	160	7:1
10^{20}	2048	210	10:1
10^{78}	21,000	600	35:1

Measured performance of public-key algorithms [4].

Table 2-Speed Up Ratio of ECC over RSA

	ECC-160	RSA-1024	ECC-224	RSA-2048
Operations/ sec	271.3	114.3	195.5	17.8
Speed up	2.4:1		11:1	

Doubling an RSA or an ECC key size multiplies the processing time by 8. The Application domain, that motivated our research, was the implementation of an Encryption Scheme that can utilize the Keys generated using Elliptic Curves, and also that offers higher level of security with simple operations. In this paper, we have proposed an Encryption Scheme based on Multiplication with the same El-Gamal Scheme of key computation. Besides constructing an Efficient Encryption Scheme, we have also suggested a Scalar Multiplication scheme, which has reduced number of Point Additions. Though multiple base Scalar Multiplications have already been proposed[15], we have presented a single base Scalar Multiplication scheme, where the base will be power of 2 and we have also provided an expression for choosing an optimal base for having least number of point Additions.

2. PRELIMINARIES

In this section, we take a look at the basic concepts of Elliptic curves.

Let F_p be a prime finite field so that p is an odd prime number, and let a,b € F_p .Then a,b satisfy

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

Then an elliptic curve $E(F_p)$ over F_p defined by the parameters a,b € F_p consists of the set of solutions or points P= (x,y) for x,y € F_p to the equation:

$$E:\ y^2 \pmod p = (x^3+ax+b) \pmod p$$

together with an extra point O called the point at infinity. The equation

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

is called the defining equation of $E(F_p)$. For a given point P= (x_P,y_P), x_P is called the x-coordinate of P,and y_P is called the y-coordinate of P.

The number of points on $E(F_p)$ is denoted by # $E(F_p)$.The Hasse Theorem states that

$$\mathbf{p+1-2\sqrt{p} \ \le\ \#\ E(F_p)\ \le\ p+1+2\sqrt{p}}$$

2.1 Notations

The Notations given below are used throughout the paper:

F_p : Prime field for the prime number p.

E : A non-super singular Elliptic Curve with a set of points (x,y) in the prime field F_p.

a,b : Curve Parameters satisfying the equation

n : Bit size of the Keys.

P : Base point.

k : A large prime of the order of p and k<p. This is the Private Key.

Q : The Public Key computed from k & P.

SK : Session Key.

r : Randomly chosen number of Bit Size n.

R : An Elliptic Curve point satisfying the equation of the curve and it is computed as R=rP.

D : Session Key Computed at Decryption End

2.2 Elliptic Curve Arithmetic

2.2.1 Adding two distinct points P and Q

Consider the point $P(x_p,y_p)$ that belongs to the Elliptic Curve E.The negative of the point P is -P(x,-y).Take another point $Q(x_q,y_q)$.P+Q=R , where $R=(x_r,y_r)$,

where x_r,y_r are computed as follows

$$S=(y_p - y_r)/(x_p - x_r)$$

$$x_r=(S^2-x_p-x_q) \pmod p$$

$$y_r=(-y_p+S*(x_p-x_r)) \pmod p$$

2.2.2 Doubling the point P

Given P(x,y) and if y is not zero ,then we calculate R=2P as follows.

$$S=((3x^2+a)/2y_p) \pmod p$$

$$X_r=(S^2-2x_p) \pmod p$$

$$Y_r=(-y_p+ S*(x_p-x_r)) \pmod p$$

2.3 Scalar Multiplication

We use the following Algorithm for faster Field Multiplications. Let P(x, y) be a point that belongs to the Elliptic Curve E..We wish to multiply this with a large prime , **k**. Let n be the bit-length of the **k**. Then, **k** can be represented as follows:

$$K=k_{n-1}*2^{n-1}+k_{n-2}*2^{n-2}+k_{n-3}*2^{n-3}+\ldots+k_1*2+k_0 \text{ with } k^{n-1}=1.$$

Algorithm :

1. Set Q←P
2. For i=n-2 to 0 do

   Q←2Q

   If k_i=1 then

   Q←Q+P

   End if

   End for
3. Return Q

The Running time for this Algorithm is $\log_2 n$, where n is the bit length of the prime **k**.

2.4 Proposed Scalar Multiplication

For performing Scalar Multiplication over a point P with a number k, with base 2 would make n-Point Doublings and j-Point

Additions, where **j** is the number of set bits in the bitwise representation of the number **k.** At the worst case this method requires n-1 doublings and n-1 additions.

In this section we have proposed a Scalar Multiplication scheme that will reduce the No. of Point Additions for the same No. of Point doublings. Here, we represent the Number k, with another base instead of the Binary representation (Base 2 representation).

Now, we represent the Number k, with the new Base b, where $\mathbf{b=2^g}$, where **g** is an integer.

$$\text{So, } k= r_{n-1}*b^{n-1}+r_{n-2}*b^{n-2}+r_{n-3}*b^{n-3}+\ldots+r_1*b+r_0$$

Where r_i's are the coefficients in the New Base. So here, the values of r_i can take values that vary from 0 to 2^g-1.Now the Algorithm is rewritten as follows:

Algorithm:

1. Precompute **uP** where u ranges from 2 to 2^g-1
2. Assign $Q \leftarrow r_{n-1}P$
3. For i=n-2 to 0 do.

   $Q \leftarrow bQ$ $(b=2^g)$.
   If ($r_i \neq 0$)
   $Q \leftarrow Q+r_iP$
   End If
   End for
4. Return Q.

For computing Q in this key, an optimal value for g has been formulated, so that the Scalar Multiplication results in least No. of Additions, with the Precomputations taken into consideration. Let the Key-Size be **n**. The optimal value for g is ,

$$\mathbf{g=ceil(\log_2(n) / 2)}$$

This formula works only for values of n upto 1024.Generalizing the above algorithm,

m_b : denotes the total number of bit or bit length in the proposed base b.

Therefore, $\mathbf{m_b=n/g}$ $\mathbf{(b=2^g).}$
T(b) :Denotes the time complexity for computing the Public key with the New base, b , where $b=2^g$

A_t : Denotes Time Complexity for Point Addition Operation.

D_t : Denotes Time Complexity for Point Doubling Operation.

Then,

$$\mathbf{T(b)= (ceil(n/g) - 1)gD_t + (ceil(n/g) – 1) A_t+(2^g-1)A_t}$$

For Eg. Let the bit size be 128.

g=ceil(7/2)=4 => g=4

So No. of Additions = $((2^4-1)+128/4)=47$

If g is chosen to be 5, then

No. of Additions=ceil($(2^5-1)+128/5$)=57

Similarly, for n=512 , g=ceil(9/2)=5

So , No. of Additions=ceil($(2^5-1)+512/5$)=134

If we randomly choose the base value to be 2^6 or g=6 , then

No. of Additions=ceil((2^6-1) +512/6)=149

So, from these we conclude that $\mathbf{g=ceil(\log_2(n) / 2)}$ **,**proves to compute a base that would yield least No. of Point Additions. Whereas, if we follow the method specified in section 2.3 will make j point Additions, where j is the No. of Set Bits in the Private Key,k.

3. PROPOSED ENCRYPTION AND DECRYPTION SCHEME

3.1 Mathematical Concept of Encryption and Decryption

Consider a Prime Number p, another number m and e where e<p and m<p.Then,

$$m*e \equiv c \ (mod\ p) \qquad (1)$$

where c is the remainder obtained by multiplying m & e and with respect to prime p.

We find number d , such that :

$$e*d \equiv 1 (mod\ p) \qquad (2)$$

The above operation is accomplished using Extended Euclidean Algorithm. Finally m is recovered as follows:

$$d*c \equiv m \ (mod\ p) \qquad (3)$$

So the actual operation

$$c=(m*e)\ mod\ (p)$$

$$d*c\ (mod\ p)= d*(m*e)=(m*1)=m\ (mod\ p)$$

m is recovered in this way. **d** is computed using Extended Euclidean Algorithm.

(1) is the Encryption Operation and (3) is the Decryption Operation.

3.2 Operations With the Keys

We compute SK as SK=rQ. SK has two components that are the x-co-ordinate and the y-co-ordinate. SK represents the session key that represents the Key, with which data is actually encrypted.

3.3 Encryption

Let the Bit Size of the key be n .The Block Size of the Message to be encrypted will be of size **n**. The Block of message is represented as an n-bit number by simply concatenating the Base-2 representation of each character in the message.

3.3.1 Step 1

We choose a message block m of Size n. The x-co-ordinate of SK becomes the Encryption Key and in this step, we accomplish the modular operation with respect to the Private-Key k. The remainder generated is tempc.

$$m * (SK.x) \equiv tempc \ (mod\ k)$$

3.3.2 Step 2

Here we multiply tempc generated in step1 with the corresponding y-co-ordinate of Session Key,SK with which Encryption was performed with it's x-co-ordinate in Step1. This operation is done with respect to the Field Prime p. The remainder obtained in this step forms the final Cipher Text. This is denoted by c.

$$tempc * (SK.y) \equiv c \ (mod\ p)$$

Reason for using k is that, we don't introduce a new variable into the Encryption process and so the overhead in transferring Secret components reduces, since the Private Key is directly involved in Key Generation and in Encryption, Decryption Processes.Finally, R is appended to the Cipher Text.

3.4 Decryption

We have used the El-Gamal Scheme for Session Key Generation. After receiving the secret-key, we perform scalar multiplication to compute D=kR, where R is retrieved from the Cipher Text, to which it was appended during encryption.

But R=rP.

So, D=krP.

During Encryption, we computed SK as SK=rQ, where Q is the public-key. But Q=kP.

So , SK=rkP=krP=D.

So, SK is recomputed as D here and the same is used for decryption.

3.4.1 Step 1

The Size of the Block remains the same, n. Represent this block of data as a n-bit number, **c** and Multiply it with the corresponding Decryption key d_1 , where

$$d_1 * D.y \equiv 1 (mod\ p)$$

since this operation was performed last during Encryption Operation. This is done with respect to the Field Prime p. The remainder generated is tempc.

$$c * d_1 \equiv tempc \ (mod\ p)$$

3.4.2 Step 2

Here we multiply tempc generated in step1 with the corresponding decryption key for step2 ,d_2 , where

$$d_2 * D.x \equiv 1 (mod\ p)$$

This operation is done with respect to the Secret-Key k. The Remainder obtained is the original message. Thus the original m is recovered.If private Key, k<p

$$tempc * d_2 \equiv c \ (mod\ k)$$

If private Key, k>p

$$(tempc+p) * d_2 \equiv c \ (mod\ k)$$

The Remainder obtained is the original message. Thus the original m is recovered.

4. SECUTITY

4.1 Security of ECC

The best known attack for Elliptic Curve Cryptography is the Pollard-Rho Attack. For a given key size of n bits, the time taken to solve the ECDLP is given by the formula

$$t = \sqrt{\pi m / 2} \quad , \text{where } m = 2^n$$

So for a 128 bit key, the time taken solve the ECDLP on 10^4 Computers, processing at the Rate of 10^4 keys/s for each computer, it will take about 10^4 years to compute the Secret key k.

ECDLP was solved for a 109-bit Key with the help of 10000 computers and it took about 549 days to solve it on P-4 Computers using Pollard-Rho Method.[13]

Other methods such as Baby-Giant Step exit, but are not much efficient than the Pollard-Rho Attack.

4.2 Security of Encryption/Decryption Operations

If a hacker wishes to perform Cipher Text Only Attack (Methods other than ECDLP) with the public key components through the Encrypted data, then there are two possible ways to hack:

4.2.1 Using Factorization

p is our Field prime and let c be a block of data that a hacker wishes to hack. a,b,p are Numbers of the same bit-size w and p is a prime greater than a and b. Then,

$$a * b \equiv c \ (mod\ p) ,$$

Only c and p are known to the hacker. So the only way to hack this is to form a number of this form,i.e. **h=(p*x+c)** where x varies from 1 to 2^w. For every of h , formed in this way, has to be factorized into a & b respectively which is another tedious operation. .

For a 128-bit Encryption Scheme, the time complexity to crack this Scheme would require

$t=2^{128}$*(complexity to factorize every value of h)

Which shows that **$t > 2^{128}$** .

In spite of finding, the y-coordinate of Session Key, the hacker has to find the corresponding x-coordinate(In Elliptic Curve Arithmetic, same more than one point can have the same y-coordinate.) and also the Private Key. This is a very tedious task and such an approach will fail

4.2.2 Using Brute force Attack on Private Key

Other than the factorization scheme said above, a hacker can perform a Brute force attack on the private key, k. For every value of k, the hacker will have to find the co-ordinates of the Session Key. So the time complexity involved would be,

$t=2^{128}$*(complexity to compute SK for every value of k)

Thus, this Encryption Scheme proves to be **Secure & Efficient** than **AES,** that offers maximum security among the Private Key Cryptographic Algorithms ,and hence much **Secure & Efficient**

than **ECAES** in terms of the Encryption scheme . As a whole, it proves to be a secure Encryption scheme than RSA.

5. SIMULATION RESULTS

We compared the Modified version of Scalar Multiplication with the Scalar Multiplication performed with Base 2 for Bit Sizes of 128,160,192,256.Fig.1 represents this comparison.

SM in Fig. 1 denotes Scalar Multiplication and Mod SM denotes the proposed Scalar Multiplication. Time is expressed in µs.

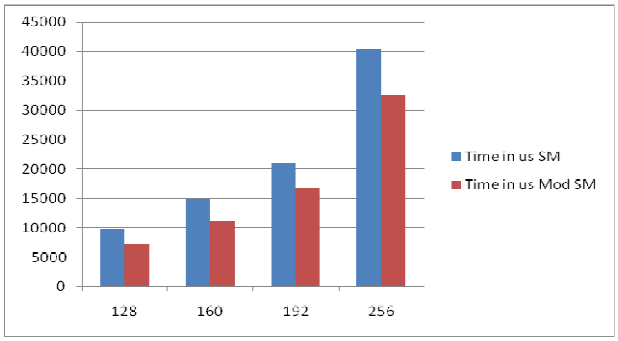

Figure 1. Comparison of Scalar Multiplication with base 2 and with base 2^g

Finally, we compared RSA-768 with MECES-128, since both offer equal security. We found that, MECES is far efficient than RSA in Encryption and Decryption, since RSA involves Exponentiation, whereas MECES involves only Multiplication. Fig. 2 shows the comparison between RSA and

MECES for time taken to encrypt 1 KB of data.

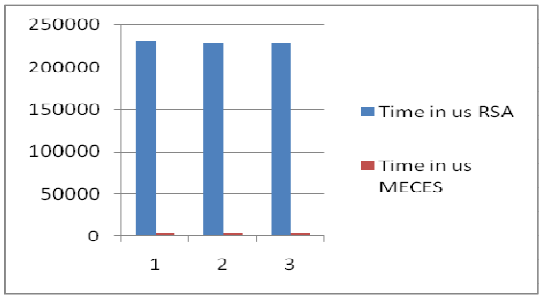

Figure 2 .Comparison of time taken to encrypt data , RSA & ECC .

Fig.3 compares the RSA and MECES for the time taken to decrypt 1KB of data, previously encrypted by itself.

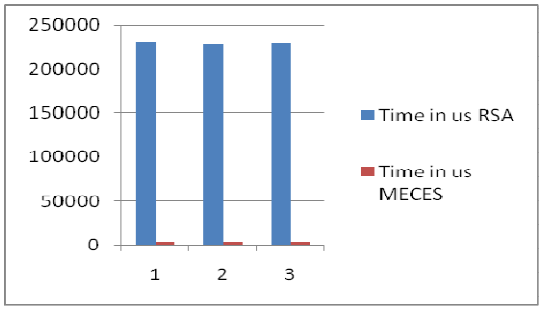

Figure 3 .Comparison of time taken to decrypt data , RSA & ECC .

6. CONCLUSION

1. Since the Encryption Scheme is secure, any hacker will be forced to solve the Elliptic Curve Discrete Logarithm Problem.

2. The Encryption Scheme is based on multiplication. So, it can be applied to constrained Systems like Wireless Devices, Mobile Phones etc.

3. Use of Multiplication, makes it FPGA implementable.

4. Since the time required to compute the keys is less, Timing based attacks on the Key will be difficult.

5. The Encryption Process involves two stages of Modular Multiplication. Hence Known Plain Text Attack will also be difficult, since the Number of unknown variables involved in the encryption are more. (Private Key and the Session Key Co-ordinates).

7. REFERENCES

[1] Jia Xiangyu,Wang Chao "The Application of Elliptic Curve Cryptosystem in Wireless Communication" 2005 IEEE International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications Proceedings.

[2] Abdullah, M., Bellare, M. and Rogaway, P, "DHAES:an encryption scheme based on the Diffie-Hellman problem". Contribution to IEEE P1363. 1998.

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

[4] Vipul Gupta,Douglas Stebila,Sheueling Chang Shantz ,"Integrating Elliptic Curve Cryptography into the Web's Security Infrastructure".

[5] G.V.S. Raju and Rehan Akbani." Elliptic Curve Cryptosystem and its Applications ".0-7803-7952-7/03/$17.00 0 2003 IEEE.

[6] Certicom White Papers ,"THE ELLIPTIC CURVE CRYPTOSYSTEM FOR SMART CARDS ".

[7] Guide to Elliptic Curve Cryptography - D. Hankerson.

[8] Bruce Schneier - Applied Cryptography, Second Edition - John Wiley & Sons [ISBN0471128457]

[9] MIT Press Handbook of Applied Cryptography.

[10] W. Stallings," Cryptography and Network Security", Prentice Hall, Second Edition,1998.

[11] Handbook of Elliptic and Hyperelliptic Curve Cryptography by Henri Cohen and Gerhard Frey.

[12] M.Aydos, B.Sunar, and C.K.Koc ,"An Elliptic Curve Cryptography based Authentication and Key Agreement Protocol for Wireless Communication" .2nd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, Dallas, Texas, October 30, 1998.

[13] D. M. Gordon, "A Survey of fast exponentiation methods," J. Algorithms, 27, 1998,pp. 129-146.

[14] Y. Han, P. Lmng, P. Tan, and J. Zhang, "Fast Algorithms for Elliptic Curve Cryptosystems over Binary Finite

Field," Advances in Cryptology- CRYPT0'99, LNCS 1716, pp.75-85.

[15] Daniel J. Bernstein, Peter Birkner, Tanja Lange, and Christian Peters, "Optimizing double-base elliptic-curve single-scalar multiplication" .