Implementation of Text based Cryptosystem using Elliptic Curve Cryptography

1. Problem definition

It is almost infeasible *Data* encryption is widely used to ensure security in open networks such as the internet. With the fast development of cryptography research and computer technology, the capabilities of cryptosystems such as of RSA and Diffie-Hellman are inadequate due the requirement of large number of bits. The cryptosystem based on Elliptic Curve Cryptography (ECC) is becoming the recent trend of public key cryptography. This paper presents the implementation of ECC by first transforming the message into an affine point on the Elliptic Curve (EC), over the finite field GF(p). In ECC we normally start with an affine point called Pm(x, y) which lies on the elliptic curve. In this paper we illustrate the process of encryption/decryption of a text message to attempt a brute force attack to break the cryptosystem using ECC.

2. **SOLUTIONS**

2.1 Existing Methodology

1. **RSA**

The RSA(Rivest-Shamir-Adleman) scheme is block cipher in which the plaintext and ciphertext are integers between 0 and n-1 for some n. A typical size of n is 1024 bits, or 309 decimal digits. Plaintext is encrypted in blocks, with each block

having a binary value less than some number n. The defence in RSA against brute force attack is to use a large key. However since calculations are involved in both key generation and in encryption and decryption, the larger the size of the key, the slower the system will run.

2. **Diffie-Hellman Key-Exchange**

The purpose of the algorithm is to enable two users to securely exchange a key that can be used for subsequent encryption of messages. The algorithm depends itself is limited to the exchange of secret values. Its effectiveness depends on the difficulty of computing discrete logarithms. The key exchange protocol is vulnerable to Man-in- the- Middle Attack.

2.2 Proposed Approach

This paper presents the implementation of ECC by first transforming the message into an affine point on the Elliptic Curve (EC), over the finite field GF(p). In ECC we normally start with an affine point called Pm(x,y) which lies on the elliptic curve. The paper illustrates the encryption and decryption of a text message.

Elliptical Curves

In general, cubic equations for elliptical curves take the following form:

$$\mathbf{y^2 = x^3 + ax + b.}$$

where x and y are points from GF(p), and a and b are integer modulo p satisfying,

$$\mathbf{4a^3+27b^2 \neq 0.}$$

GF(p): For a given prime, p, we define the finite field of order p, GF(p), as the set Z_p of integers {0,1,2,...,p-1} together with the arithmetic operations modulo p.

Also included in the definition of elliptical curve is a single element denoted by O and called the *point of infinity* or the *zero point*. An elliptical curve E consists of points (x , y) satisfying the above two equations. The set of points (x , y) are said to be affine points.

Addition of two points

For two points P= (x_p,y_p) and Q=(x_q,y_q), that are not the negatives of each other, the slope of the line *l* that joins them is :

$$\mathbf{S^2= [(y_q - y_p)/(x_q-x_p)]mod\ p}$$

We can express the sum R=P+Q as

$$\mathbf{x_r=(S^2 - x_p - x_p)mod\ p}$$
$$\mathbf{y_r= [-y_p + S(x_p-x_r)]mod\ p}$$

Doubling of a point

Let P= (x_p,y_p) be a point. For doubling P we use the following equations:

$$\mathbf{S=[(3xp2+a)/2yp]\ mod\ p}$$

Then R=2P,

$$\mathbf{x_r=(S^2 - 2x_p)\ mod\ p}$$

$$\mathbf{y_r=[S(x_p - x_r) - y_p]\ mod\ p}$$

Elliptical Curve Cryptography requires scalar multiplication. Suppose P(x,y) is a point. To determine kp, where k is any integer, we use both addition and doubling.

Base point: The smallest co- ordinate point (x ,y) that satisfies the EC. It is denoted by G.

Suppose A wants to send a message to B.

Let G be the base point and let Pm be another affine point satisfying the EC. Let k be any integer such that k<p, k is kept as secret. Calculate kG using additions and doublings. Select a private key n_b for B. The public key of B is

$$\mathbf{P_b=n_bG}.$$

Suppose A wants to send a character to B, then evaluate

$$\mathbf{P_{m1}= (ASCII\ value\ of\ character)P_m}$$

Now calculate kP_b.

The encrypted message is $(kG, P_{m1}+kP_b)$

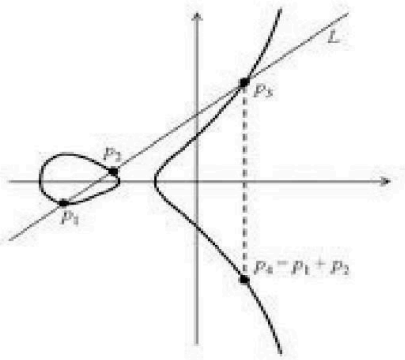
To decrypt the message B multiplies n_b with kG; $n_bkG=kP_b$ and thus gets Pm1 since

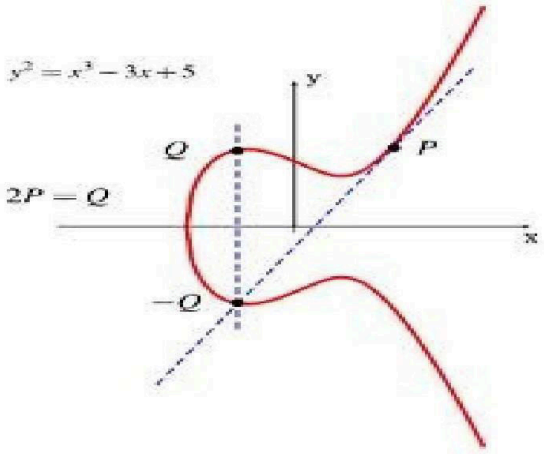
$$\mathbf{P_{m1}= P_{m1}+kP_b-n_bkG.}$$

$$\mathbf{(ASCII\ value\ of\ character)P_m=P_{m1}.}$$

Mathematical background

case 1 ***case 2***

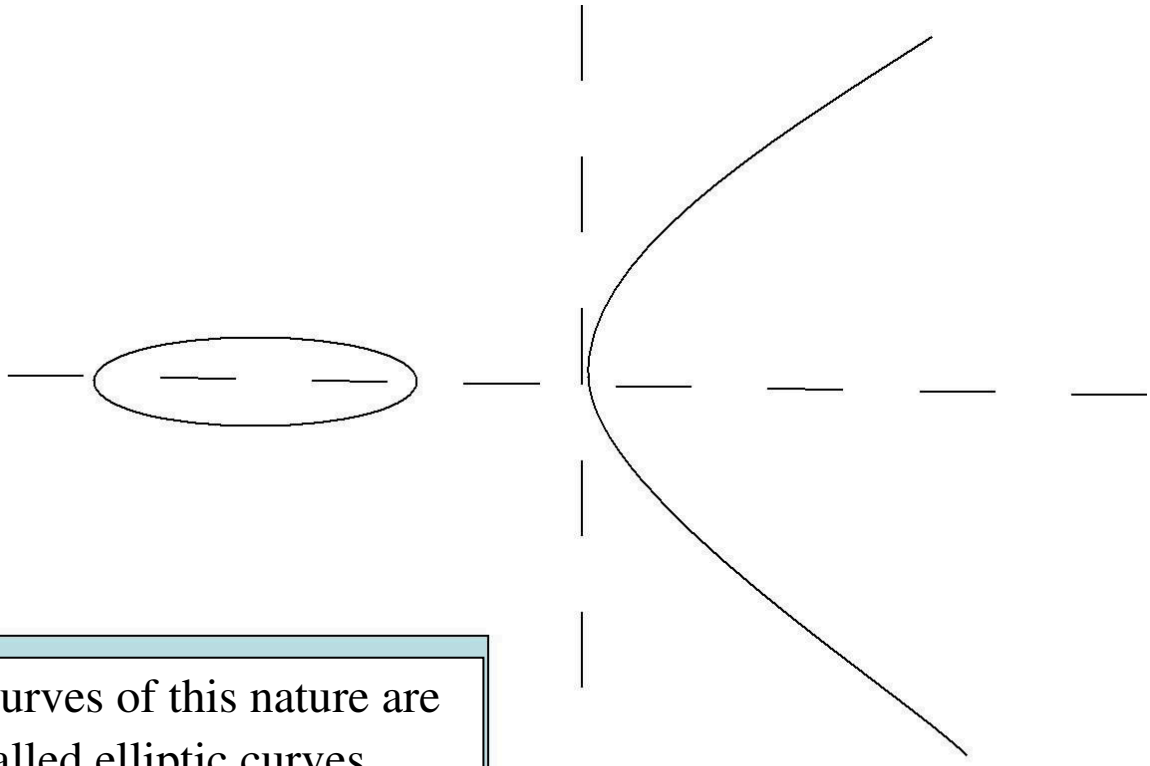




case 1: Addition operation.

case 2 : Doubling operation.

Elliptic curve



Curves of this nature are called elliptic curves

Definition of Elliptic curves

An elliptic curve over a field *K* is a nonsingular cubic curve in two variables, $f(x,y) = 0$ with a rational point (which may be a point at infinity).
The field *K* is usually taken to be the complex numbers, real numbers, rational numbers, algebraic extensions of rationals, p-adic numbers, or a finite field.
Elliptic curves groups for cryptography are examined with the underlying fields of F_p(where $p>3$ is a prime) and F2m (a binary representation with 2m elements).

General form of a EC

An *elliptic curve* is a plane curve defined by an equation of the form:

$$y^2=x^3+Ax+B \quad \text{..... (1)}$$

Points on the Elliptic Curve (EC)

Elliptic curve over a field L is composed of the points:

$$E(L)=\{\infty\}\cup\{ (x, y) \in L \times L \mid y^2+ \ldots = x^3 + \ldots \}$$

The point infinity(∞) is added to the above list for a special reason. Consider a line passing vertically through the curve. In such a case the line shall intersect the curve at only two points.

However as the curve is cubic in x, there should be three points on the curve satisfying the line .We consider the point at infinity (usually denoted by O) to be that third point.
O is considered to lie both at the top and bottom of the curve.

3. Solution Analysis

3.1 PROS

1. Shorter key length
   - Same level of security as RSA achieved at much smaller length.

2. Better Security
   - Secure because of ECDLP(elliptic curve discrete logarithmic problem.
   - Higher Security per key bit than RSA.

3. Higher Performance
   - Shorter key length ensures lesser power requirement suitable in wireless sensor applications and low power devices, reducing storage transmission requirements.
   - More computation per bit but overall lesser computational expense or complexity due to lesser no. of key bits.

3.2 CONS

1. Relatively Newer Field:
   - Idea prevails that all the aspects of the topic may not have been explored yet, possibly unknown vulnerabilities.
   - Doesn't have widespread usage.

2. Not perfect:
   - Attacks still exist that can solve ECC(112 bit key length has been publically broken).
   - Well known attacks are pollard's Rho attack (complexity O(square root (n))), pohlig's attack, Baby step, Giant step etc.

4. Implementation

4.1. S/W and Hardware requirements

S/W: C++ program used for implementation.

H/W: Communication channel and required hardware support for practical implementation.

4.2 Pseudo code

To do operations with EC points in order to encrypt and decrypt the points are to be generated first. The algorithm *'genPoints'* describes the process of generating the points for the given parameters 'a', 'b', and 'p'. Also the algorithm *'ECC'* describes the process of encryption and decryption on EC field.

Algorithm gen Points (a, b, p)
{
x-O;
While(x <p)
y^2 = (~ + ax + b) mod p;
if (y^2 is a perfect square in GF(p))
output(x, sqrt(y)) (x, -sqrt(y));
x = x+I;
}

Algorithm ECC

{

// *Key* Distribution

// *Let* UAand UBbe legitimate users

UA= {PA, nA} //*Key* pair for UA

UB= {PB,nB} //*Key* pair for UB

//*Send* the Public key of Ui, to UA

Send(PB,UA);

//*Send* the Public key of Ux to UB

Send (PA, UB);

// Encryption at A

Pml=aPm

// *la:* Ascii value of text

// *Pm:* random point on EC

PB=nB * G

// *G* is the base point of EC

*lin*B is the private key

CipherText={kG,Pml+k*PB

}

// Decryption at B

Let kG be the first point and

Pml + k*PB be the second point

nBkG = Ilg * first point;

Calculate Pml = Pml + kPB- nBkG;

Calculate the Pmvalue from Pml

using discrete logarithm

}

5 .Application Areas

(a) .Web Server:-

Here we are talking precisely by taking Apache web server as a benchmark, it was found that an Apache web server can handle 11%-31% more HTTPS requests per second when using ECC rather than RSA at short-term security levels.

(b) E-COMMERCE:-

There are card issue, card owner, business, bank, payment gateway and certificate in E-commerce transaction. We encrypt the payment code using ECC in order to protect the bank account and the transaction information is encrypted by ECC avoid to distortion

(c) Cellular Telephone:

To provide end to end security we are using Public key cryptography i.e., Elliptic curve cryptography. Secure Access Authentication in mobile communication is very crucial to protect information of the subscribers and avoid fraud. It has been mentioned in many literatures that a considerably smaller key size can be used for ECC compared to RSA.

(d) Improved authentication and key arguments using elliptic curve cryptography.

(e) Wireless security.

6. Future Enhancements

Security of Wireless Sensor Networks (WSNs) is a very crucial factor when deployed for reconnaissance in sensitive areas. The existing symmetric encryption scheme provides a good amount of security, but maintenance of keys is difficult. When asymmetric schemes are used, maintenance of keys is easier, but they provide a lesser degree of security when compared to symmetric encryption schemes. In order to cope up with these shortcomings, we propose to use an improved version of the hybrid encryption scheme, which is a combination of Advanced Encryption Standard (AES) and Elliptical Curve Cryptography (ECC) with cross encrypted keys for secure key exchange and node authentication and hybrid encryption for enhanced cipher-text security. In case of transmission in WSNs, Statistical Cooperative Diversity based on Alamouti code is the most commonly used transmission scheme. However, for an arbitrary number of sensors, Alamouti code limits the BER performance and energy is not distributed equally, thereby creating the energy hole problem which leads to early dysfunction of the sensors and may eventually lead to dysfunction of the Wireless Sensor Network (WSN). Extended Cooperative Space-time Block Codes (ECBSTBCs), which are obtained from Alamouti code, have the same characteristics of Alamouti code with the energy being distributed equally among the active sensors. With these factors in mind, we propose to use ECBSTBC as the transmission scheme. The improved hybrid scheme is ideal for ECBSTBC based WSN due to the speed of operation and higher degree of security that it offers.

7. Conclusion

As per the the paper it is concluded that, a text based Elliptic Curve Cryptosystem is implemented. Each character in the message is represented by its ASCII value. Each of these ASCII value is transformed into an affine point on the EC, by using a starting point called P_m. Transformation of the plaintext ASCII value by using an affine point is one of the contributions of this work. The purpose of

this transformation is two fold. Firstly a single digit ASCII integer of the character is converted into a set of co-ordinates to fit the EC. Secondly the transformation introduces non-linearity in the character thereby completely camouflaging its identity. This transformed character of the message is encrypted by the ECC technique. Decryption of ECC encrypted message is itself quite a formidable task, unless we have knowledge about the private key 'nB' ,the secret integer 'k' and the affine point P_{ml}.

The attractiveness of ECC, compared to RSA, is that it appears to offer better security for a smaller key size, thereby reducing processing overhead. The benefits of this higher-strength per-bit include higher speeds, lower power consumption, bandwidth savings, storage efficiencies, and smaller certificates. These advantages are particularly beneficial in applications where bandwidths, processing capacity, power availability or storage are constrained.

8. REFERENCES:

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

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

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

[4] Ray C. C. Cheng, Nicolas Jean-baptiste, Wayne Luk, and Peter Y. K Cheung,"Customizable Elliptic Curve Cryptosystems" , *IEEE Trans. On VLSI Systems,* vol. 13, no. 9, pp. 1048-1059, Sep. 2005.

[5] Alessandro Cilardo, Luigi Coppolino, Nicola Mazzocca, and Luigi Romano,"Elliptic Curve Cryptography Engineering", *Proceedings of the IEEE,* Vol. 94, no. 2, pp. 395 - 406, Feb. 2006.

[6] C. 1. McIvor, M. McLoone, and 1. V. McCanny, "Hardware elliptic curve cryptographic processor over GF(p)," *IEEE Trans. Circuits Syst.l* Reg. Papers, vol. 53, no. 9, pp. 1946-1957, Sep. 2006.

[7] Gang Chen, Guoqiang Bai, and Hongyi Chen, " A High-Performance Elliptic Curve Cryptographic Processor for General Curves Over GF(p) Based on a Systolic Arithmetic Unit" , *IEEE Trans. Circuits Syst. - 11:* Express Briefs, vol. 54, no. 5, pp. 412- 416, May. 2007.

[8] Standard specifications for public key cryptography, *IEEE standard,*pI363,2000.

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

A report by,

Manish Kumar Yadav (M120383CA)
Arvind Singh (M120352CA)
C.Haritulsi (M120381CA)
Aayush Kumar (M120382CA)
Jayshankar Yadav (M120387CA)
Haripriya (M120440CA)