International Journal of Computer and Communication Technology

Volume 1 | Issue 4 Article 2

October 2010

Elliptic Curve Cryptosystem for Email Encryption

Abhijit Mitra
Computer Science & Engineering, Calcutta Institute of Engineering & Management, West Bengal University of Technology 24/ 1A, Chandi Ghosh Road, Kolkata – 700040, India, abhijit.system@gmail.com

Saikat Chakrabarty
Computer Science & Engineering, Calcutta Institute of Engineering & Management, West Bengal University of Technology 24/ 1A, Chandi Ghosh Road, Kolkata – 700040, India, saikat.chakrabarty@yahoo.com

Poojarini Mitra
Computer Science & Engineering, Calcutta Institute of Engineering & Management, West Bengal University of Technology 24/ 1A, Chandi Ghosh Road, Kolkata – 700040, India, poojarini.mitra@gmail.com

Elliptic Curve Cryptosystem for Email Encryption

Abhijit Mitra [1], Saikat Chakrabarty [2], Poojarini Mitra [3]

Computer Science & Engineering, Calcutta Institute of Engineering & Management,
West Bengal University of Technology
24/ 1A, Chandi Ghosh Road, Kolkata – 700040, India
abhijit.system@gmail.com 1
saikat.chakrabarty@yahoo.com 2
poojarini.mitra@gmail.com 3

Abstract–

The idea of information security lead to the evolution of cryptography. In other words, cryptography is the science of keeping information secure. It involves encryption and decryption of messages. The core of cryptography lies in the keys involved in encryption and decryption and maintaining the secrecy of the keys. Another important factor is the key strength, i.e. the difficulty in breaking the key and retrieving the plain text. There are various cryptographic algorithms. In this project we use Elliptic Curve Cryptography (ECC) over Galois field. This system has been proven to be stronger than known algorithms like RSA, DSA, etc. Our aim is to build an efficient elliptic curve cryptosystem for secure transmission or exchange of confidential emails over a public network.

Keywords– **Email encryption, Cryptography, ECC, Encryption, Decryption, Plaintext, Ciphertext, Secret key, Private key, Public key, Symmetric key cryptography, Asymmetric key cryptography, Elliptic curve, Galois field.**

I. INTRODUCTION

Cryptography is the art of achieving security by encoding messages to make them non-readable. It allows secure transmission of confidential information over insecure channels. It also allows secure storage of sensitive data on any computer. Cryptography, in addition to providing confidentiality, also provides authentication, integrity and non-repudiation of data. [1]

Based on the key, cryptosystems can be classified into two categories: S*ymmetric* and A*symmetric*. In Symmetric or Secret Key Cryptosystems, we use the same key for both encryption as well as decryption. Whereas Asymmetric or Public key cryptosystems use two different keys. One is used for encryption while the other key is used for decryption. One of the keys is made public while the other key is kept private.

Elliptic Curve Cryptography (ECC) is an approach to public key cryptography based on the algebraic structure of elliptic curves over finite fields. The use of elliptic curves in cryptography was suggested independently by Neal Koblitz and Victor S. Miller in 1985.

II. REVIEW WORK ON ELLIPTIC CURVE CRYPTOGRAPHY

Overview

The mathematical operation of ECC is defined over the elliptic curve $y^2=x^3+ax+b$, where $4a^3+27b^2 \neq 0$. Each value of 'a' and 'b' gives a different elliptic curve. All points (x,y) which satisfy the above equation plus a point at infinity lie on the elliptic curve. Fig. 1 below shows an elliptic curve. [3]

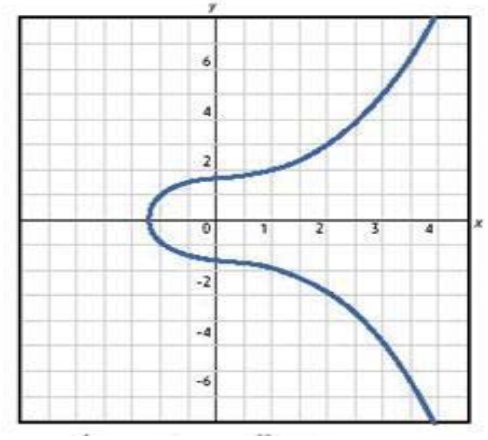

Fig. 1 An elliptic curve

The major advantage of ECC over RSA is that, it requires much shorter key lengths for ensuring the same level of security. For example, 160 bit key in ECC is considered to be as secured as 1024 bit key in RSA. The fastest known technique for taking the elliptic curve logarithm is known as the Pollard rho method. Here also ECC performs better than RSA. Moreover, security of ECC grows exponentially with its parameters while that of RSA grows sub-exponentially. The computational overhead of

ECC is $O(n^3)$, where *n* is the key length. Table 1 shows the comparison of key sizes of RSA and ECC for providing the same level of security. [6]

The disadvantage of ECC is that it involves much computation and hence is complex.

TABLE I
COMPARISON OF KEY SIZES OF RSA AND ECC

RSA key size (in bits)	ECC key size (in bits)
1024	160
2048	224
3072	256
7680	384
15360	512

A. Elliptic Curves over Galois Field

A finite field is a field with a finite field order (i.e., number of elements), also called a Galois field. A field of a finite number of elements is denoted by F_q or GF(q), where q is the number of elements[4]. Two types of finite field F_q are used :

- Finite field F_p with q = p, p is an odd prime which are called Prime Finite field.
- Finite field F_2^m with $q = 2^m$ for some m (positive integer) which are called Binary Finite field.

We are using ECC over Binary Finite field.

B. Elliptic Curves over Binary Finite field

The equation of the elliptic curve on binary field F_2^m is $y^2 + xy = x^3 + ax^2 + b$, where .

The set of points on $E(F_2^m)$ also include point O, which is the point at infinity and which is the identity element under addition. [3]

The domain parameters for elliptic curve over F_2^m are m, f(x), a, b, G, n and h.

m is an integer defined for finite field F_2^m.

f(x) is the irreducible polynomial of degree m used for elliptic curve operations.

a and b are the parameters defining the curve $y^2 + xy = x^3 + ax^2 + b$.

G is the generator point (x_G, y_G), a point on the elliptic curve chosen for cryptographic operations.

n is the order of the elliptic curve.

The scalar for point multiplication is chosen as a number between ***0*** and ***n – 1***.

h is the cofactor where ***h*** = $\#E(F_2^m)/n$. $\#E(F_2^m)$ is the number of points on an elliptic curve.

Point Addition : Consider two distinct point J and K such that

$J = (x_J, y_J)$ and $K = (x_K, y_K)$

Let $L = J + K$ where $L = (x_L, y_L)$, then

$x_L = s^2 + s + x_J + x_K + a$

$y_L = s(x_J + x_L) + x_L + y_J$

$s = \frac{y_J + y_K}{x_J + x_K}$, s is the slope of the line through J and K.

If $K = -J$ i.e $K = (x_J, x_J + y_J)$ then $J + K = O$,where O is the point at infinity.

If $K = J$ then $J + K = 2J$ then point doubling equations are used.

Also $J + K = K + J$.

Point Subtraction : Consider two points J and K such that $J = (x_J, y_J)$ and $K = (x_K, y_K)$.

Then $J - K = J + (-K)$ where $-K = (x_K, x_K + y_K)$

Point Doubling : Consider a point J such that

$J = (x_J, y_J)$, where $x_J \neq 0$

Let $L = 2J$ where $L = (x_L, y_L)$, then

$x_L = s^2 + s + a$

$y_L = x_J^2 + (s + 1) * x_L$

$s = x_J + \frac{y_J}{x_J}$, s is the tangent at point J and a is one of the parameters chosen with the elliptic curve.

If $x_J = 0$ then $2J = O$, where O is the point at infinity. [1]

C. Encryption-Decryption Algorithm

Firstly, the persons involved in the secure transmission process should agree on domain parameters. Then, the receiver has to generate a public key and a private key. The public key has to be sent to the sender. The sender then encrypts the message using this public key. The encrypted message is sent over the public network. The receiver, after receiving the encrypted message, decrypts it using his private key. Thus, he retrieves the original message. Any intruder, even if he/she manages to get the message, cannot decrypt because he does not have the corresponding private key. [2]

- Input the values of domain parameters a and b
- Input the coordinates (x,y) of about 256 points or more
- Calculate the order of these points using the formulae for point addition

- Find out the highest order and the point having the highest order (G)
- Select a private key n (any positive integer less than the highest order)
- Generate the public key by computing $P_B=n*G$
- Send the public key to the sender and keep the private key confidential
- Map each character of the message (m) to be sent to a coordinate point (P_m) satisfying the equation of the elliptic curve with the pre-decided domain parameters
- Select a random positive integer k
- Encrypt P_m by computing $C_m=\{ k*G, P_m + k*P_B \}$
- Send the encrypted message C_m to the receiver
- To decrypt the message, compute n*kG
- Then subtract the above from $P_m + k*P_B$

III. PROPOSED WORK ON EMAIL ENCRYPTION

A. Overview

To establish a secure transmission of a confidential message over a public network we have to employ a very strong method of transmission in order to avoid potential intrusions. Hence, we propose the procedure mentioned below :

Suppose entity A wants to send an email to entity B. Also, let S_A be the mail server of A and S_B be the mail server of B.

- When A creates a mail account for the first time, S_A will generate a public key $Pub_A(s)$ and a private key $Pvt_A(s)$ for receiving messages from A.
- Similarly, A will also have to generate a public key $Pub_A(r)$ and a private key $Pvt_A(r)$ for receiving messages from S_A.
- Then, S_A sends its public key to A and A sends its public key to S_A.
- Whenever A wants to send an email (ie. A clicks the 'Compose Mail' button), the message gets encrypted using the public key $Pub_A(s)$ and sent to S_A.
- S_A receives the message and decrypts it using its private key $Pvt_A(s)$.
- Now, S_A will have to send this mail to S_B. Just like A and S_A, S_A and S_B too must also maintain a pair of public key and private key each.
- Say, S_A has $Pub_{SA}(r)$ and $Pvt_{SA}(r)$ as its public key and private key for S_B respectively. These will be used by S_A for receiving messages from S_B.
- S_B has $Pub_{SA}(s)$ and $Pvt_{SA}(s)$ as its public key and private key for S_A respectively. These will be used by S_B for receiving messages from S_A.
- S_A will send its public key $Pub_{SA}(r)$ to S_B and S_B will send its public key $Pub_{SA}(s)$ to S_A.
- S_A will now use the key $Pub_{SA}(s)$ to encrypt the message and then send it to S_B.
- S_B will decrypt the same using its private key $Pvt_{SA}(s)$.
- Now, S_B will have to send the email to B. Like before, B and S_B too had to generate a pair of private and public keys each when B first created its mail account under S_B.
- Suppose, the public key and private key of B, for receiving messages from S_B, are $Pub_B(r)$ and $Pvt_B(r)$ respectively.
- B had also sent its public key to S_B.
- S_B encrypts the message using this public key $Pub_B(r)$ and sends it to B.
- Finally, B decrypts the message to obtain the original message sent by A, by using its private key $Pvt_B(r)$.

In this way, a very secure transmission of a confidential message is possible. The figure given below describes the process.

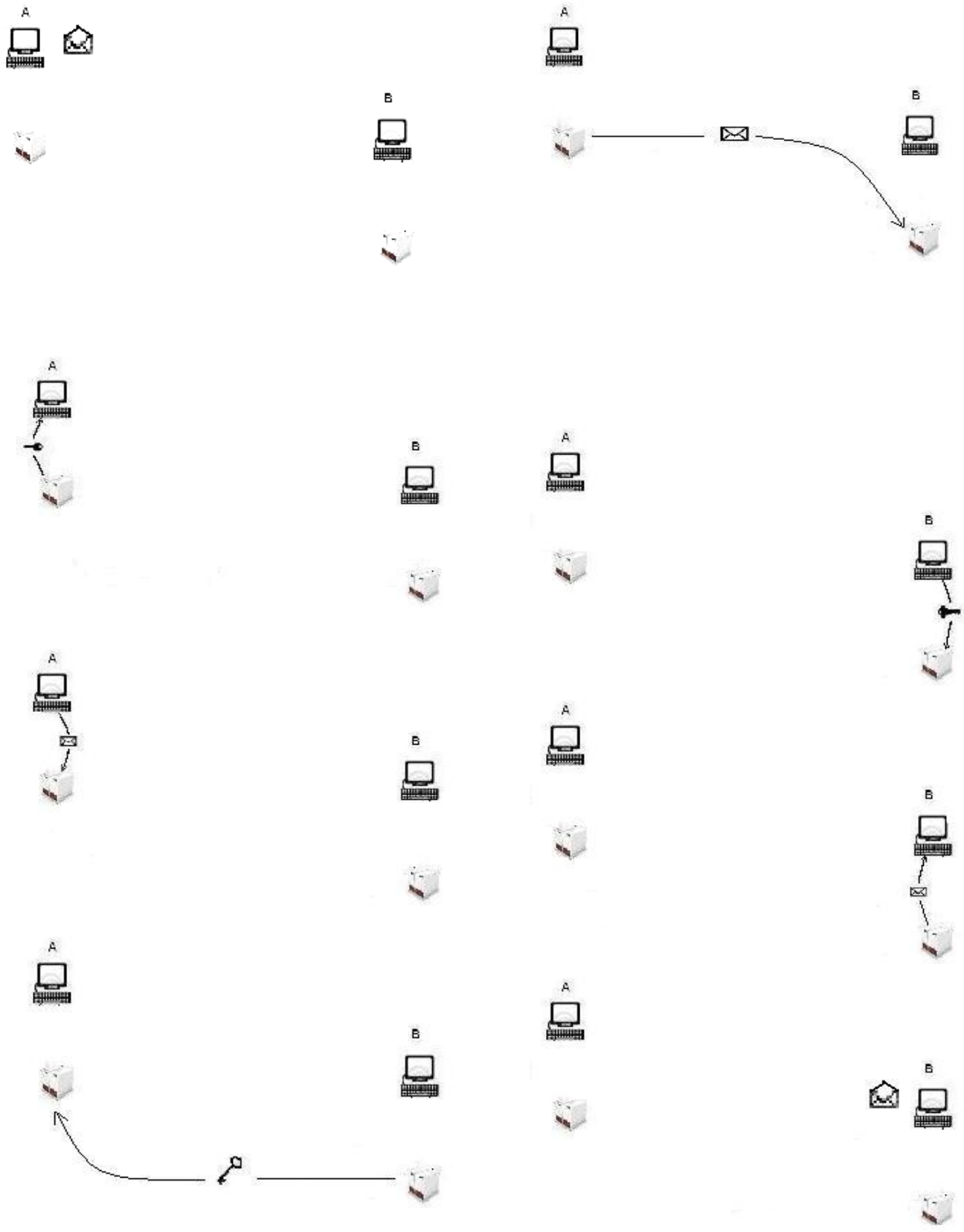

Fig 2. Email Encryption

B.
C.

D. Performance against Attacks

The proposed system is tolerant to a number of common network security attacks. [5]

- In case of eavesdropping, the original data cannot be retrieved because it is encrypted using elliptic curve cryptography.
- In case of data modification, the original data cannot be retrieved because it is encrypted and therefore the attacker cannot modify it. But arbitrary modification is possible.
- In case of identity spoofing, the attacker may engage in false conversation with a genuine user. This may be prevented by using digital signatures.
- In case of denial of service, this system may suffer since it does not have any suitable measure to prevent it.
- In case of man-in-the-middle attack, the system is safe because the data is encrypted and the attacker needs to know the private key of the receiver to retrieve the original data.
- In case of compromised-key attack, the system is quite safe because the encryption technique used here is elliptic curve cryptography and it is proven to provide high level of security. Cracking the key in ECC is very difficult and time-consuming.
- In case of sniffer attacks, again the system is safe because the attacker would be able to acquire only the encrypted data.
- In case of application-layer attack, the system may be vulnerable as the attacker may use various methods to compromise the security of the data stored. Only those attacks may be avoided where the attacker needs to use the data. The data may be rendered useless by encrypting it.

E. Advantages

The main advantage of this procedure is that it ensures very secure mode of transmission. Moreover, each transmission uses ECC and hence gives better security with small key sizes. In this system, uniformity in secure transmission is maintained as the same method is used in every transmission.

F. Disadvantages

This procedure is complex. Due to repetitive encryption and decryption, overhead increases. Also, the servers will have to maintain a huge amount of database for storing the keys of its users and other servers. In case of a server crash, the keys may be lost.

IV. CONCLUSIONS

Elliptic Curve Cryptography is about the design and analysis of mathematical techniques that enable secure communications in the presence of malicious adversaries. The principal reasons why elliptic curve cryptography gained so much popularity were its functionality, security and performance compared to other cryptographic algorithms. Another important reason is the small key size associated with this cryptographic algorithm. In particular, private-key operations (such as signature generation and decryption) and public-key operations (such as signature verification and encryption) for ECC are many times more efficient than RSA and Discrete Logarithm operations. Public-key operations for RSA are expected to be somewhat faster than for ECC if a small encryption exponent *e* is selected for RSA but here the security may suffer. The advantages offered by ECC can be important in environments where processing power, storage, bandwidth, or power consumption is constrained.

Design and implementation of elliptic curve cryptographic algorithms in a secure manner is a hard and challenging task. For efficient implementation of ECC, it is important for the point multiplication algorithm and the underlying field arithmetic to be efficient. Moreover, if the irreducible polynomial in binary field implementation is chosen to be trinomial or pentanomial the implementation of ECC can be made more efficient.

So, using ECC for email encryption will make it a very strong tool for secure transmission of confidential information over a vulnerable network. Although, this implementation of email encryption requires more time and storage space, it gives very high level of security. One may choose not to use this procedure to communicate public messages but for communicating private and confidential information, this procedure is very useful.

V. FUTURE ENHANCEMENTS

Some enhancements may be made on this system to acquire better performance :

- By using bigger domain parameters, security may be enhanced
- Digital signature may be used to authenticate the keys
- Replica servers and back-ups may be used to avoid loss of keys in case of server crashes
- Different techniques may be applied to preserve the private keys (For example, steganography)

ACKNOWLEDGEMENT

We wish to express our sincerest gratitude to our mentor for his critical suggestions, help, guidance and encouragement all through the project.

We convey our heartiest thanks to the Head of the Department and all the faculty members of Computer Science and Engineering, Calcutta Institute of Engineering and Management for their motivation and co-operation to build up this project. We are also immensely thankful to every individual who directly or indirectly contributed to this venture.

This project would not have been successful without their valuable help.

REFERENCES

[1] Darrel Hankerson, Alfred Menezes, Scott Vanstone, *Guide to Elliptic Curve Cryptography*, Springer-Verlag New York, Inc., pages 2, 81.

[2] William Stallings, *Cryptography and Network Security Principles and Practices*, Prentice Hall, 16 November 2005, pages 311,312.

[3] *Douglas Stinson,*"Cryptography: Theory and Practice"CRC Press LLC, ISBN**:** 0849385210**,** pages 32, 198-201.

[4] BRUCE SCHNEIER, "Applied cryptography", John Wiley & Sons, 1996, Second Edition, page 53-62.

[5] BRUCE SCHNEIER, *"E–Mail Security"* John Wiley & Sons, 1995,page 161-169

[6] N. Elkies. Elliptic and modular curves over finite fields and related computational issues. In Computational Perspectives on Number Theory, 21–76, 1998.