Security using identity based cryptography – Key Generation

Dr. Purvi Ramanuj [#1]

[#]*Department of Information Technology, Gujarat technological University*

[1]purviramanuj@yahoo.com

***Abstract*— In today's world, lots of transactions are carried out online and huge data is getting transmitted via various applications using computer networks. Security of data has become essential as these data are prone to many attacks. So, secure and efficient key management scheme and security of messages are crucial. Many security solutions are already available and vast research efforts are being carried out worldwide. Elliptic curve cryptography has been proven a highly secure method for network security. Biometrics is blended with identity-based cryptography for enhanced security and efficiency. A comprehensive key management scheme is presented as a robust security solution.**

Keywords— Identity based Security, Key management, Key generation, Key Revocation, Fingerprint, ECC

I. INTRODUCTION

Nowadays in the era of computers and online transactions, security of data transmitted is the main concern of people. Today whole world is moving towards the internet and there should be a strong mechanism which protects the data of the transaction. This data may be in terms of payment information or secret information of the organization or secret information of nation. Security is important whether it is a small organization applying to a tender or a common man making payment of his electricity bill, an ATM giving money to legitimate users or a nation is building economic or military policy or even in the war zone as everything everywhere is now on the internet.

To gain Security of data or information very strong way is to encrypt the data.

1.1 Security Challenges

- Most of the systems are using channels which are shared and mode of data transfer is broadcasting. Also, the environment in which it operates is not secure.
- Network nodes are physically not secured hence there exist always a chance that these nodes are attacked by a malicious node. In such situation, a distributed architecture has the upper edge.
- Trust among nodes poses a concern as nodes are roaming and topology changes very frequently. Also, a node may have more than one membership with different domains. This demands for adaptive security mechanism.
- As nodes are leaving and joining network frequently, the size of a network varies. Also, there are networks which are large in size. The security solution should be efficiently working with a large range of network size.
- Because of roaming nature of nodes, they are fueled by battery power. Also, a small size of nodes makes them restricted to computational power.

There exist many algorithms for encryption and decryption. Using this algorithm and key applying on the algorithm, security can be ascertained.

1.2 Purpose of Cryptography

Confidentiality, Availability, and Integrity – CAI are three major security goals – popularly known as CAI Triad (3).

All algorithm of cryptography depend on the piece of information called key. As algorithms are known to all, security highly depends on the key used for encryption and decryption. The key is a variable which changes its value frequently and is employed using an appropriate algorithm on a message which can be a string or block. On application, the key will transform such data into some complex – not an easily understandable form which is called encrypted data. Also, a key can recover the original data from an encrypted data which is known as decryption. Research is going on to generate key easily and securely. After generation of a key, it is distributed to each node.

Key management plays a pivotal role in providing security to data communication in the network.

1.3 Types of key management

As shown in Figure 1, cryptography can be divided into following types

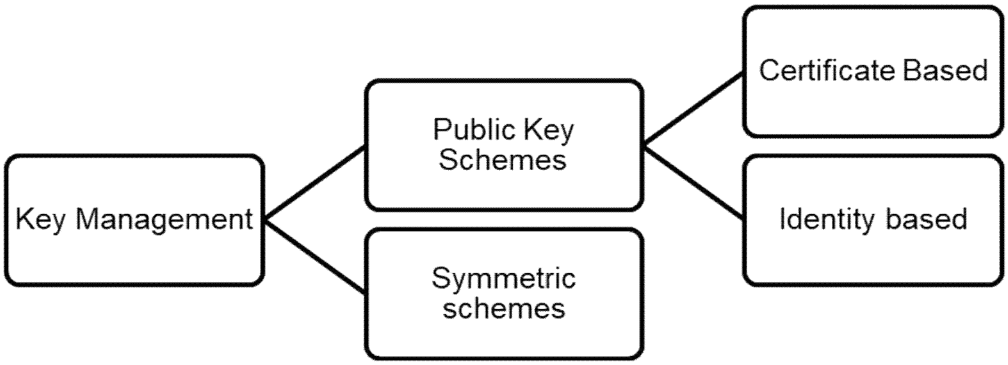

Fig 1 Types of Cryptography

1.3.1 Symmetric Key Cryptography:

In this kind of cryptography, a single key is employed by both sender and receiver. It is a shared secret between nodes in communication. The sender node should have prior knowledge of key before sharing information. Sender node encrypts data and produces ciphertext from the message using the secret key. Such data is received at receiver's end and it recovers original message again by using a same secret key. It is less complex and faster but the nodes must trust each other and share the secret efficiently and securely. In these schemes, to ensure security, keys are changed frequently.

1.3.2 Asymmetric key cryptography:

To overcome the problem of pre-sharing of secret key among communication parties, asymmetric key cryptography was introduced in the 20th century. In this type of cryptography, two different keys are employed for data encryption and decryption. These keys are mathematically related to some function or operation but on the other hand are not possible to derive private key by means of a public key. It is also known as public key cryptography because the public key is shared with all and the private key only is kept in safeguard. Sender node encrypts data using receiver's public key, which is known to all in the network. But such encrypted message can only be decrypted by a private key which is kept a secret with the receiving node. Also, a sender node can sign a message using its private key and same can be verified by the receiver node using sender node's public key.

One of the biggest challenges of public key cryptography is the user must have verified that the public key being used by him is actually belong to the intended user. This trust is provided by a third party by issuing a certificate for a user. This certificate ensures that a particular user is unique and the key indeed belongs to the same user.

1.3.3 Identity Based Cryptosystem

Shamir first presented the identity-based cryptosystem (5). Node's identity is used for generating node's public key. This identity can be an email id, IP address or biometric attribute. A trusted third party generates Private Key. As node's identity is used in key generation, it is known as ID-based cryptography (IBC).

Using an identity to generate cryptographic keys makes the cryptography simpler and there is no further need to maintain bulky certificates. New user entry is easier and in fact, the message can be encrypted for the new entrant before its actual entry in the network. It employs complex mathematics and many research efforts are employed worldwide.

Advantages of Identity-based cryptography

In this scheme, nodes are required to maintain only PKG – private key generator parameters. This makes IBC very simple and lightweight in terms of memory storage requirements.

The public key is naturally discoverable from node's ID. Key management is straightforward and more efficient. Data communication is more secure.

Disadvantage of Identity based cryptography

PKG is the central authority for generation of private keys of all nodes. So such centralized approach can be dangerous in today's networks. Also, the private keys generated by PKG need to be safely transmitted to concerned nodes. Privacy of node may be at stake once as the public key is directly derived from such identities.

II. PROPOSED SCHEME

2.1 Key generation

The fingerprint of each person is unique. This is the most important feature which is used in the proposed scheme. The scheme is designed to achieve a high level of security using fingerprint signature generated from individual finger impression so as to produce the domain parameters of the elliptic curve, or to produce private keys. The fingerprint is one of the most reliable and most used biometric features. The signature is unique and can not to be produced by another individual. Generally, in elliptic curve cryptography schemes, a random value of parameters a and b are chosen. In the proposed scheme, fingerprint data and private key generated using RSA are used respectively for these parameters.

There are two stages of the key generation process. This provides two level of security. The first stage uses RSA algorithm, which is one of best trapdoor functions used since last several years. RSA uses the mechanism of multiplication and factorization concept. As multiplication of prime numbers is very easy but factorization is very difficult, RSA was known to be best and easy algorithm for years.

The second level uses ECC - Elliptic curve cryptography, which is the best performing algorithm nowadays. An Elliptic curve cryptography includes a prime number as a maximum, a curve equation, a private key which is a random number, and a public key which is generated multiplying private key with a generator point G on the curve. Computing the private key from the public key in this kind of cryptosystem is called the elliptic curve discrete logarithm function. This turns out to be a Trapdoor Function.

2.2.1 Working Module

The operation of the proposed scheme is described in this section. The scheme has following two phases:

- Initialization
- Communication

In initialization phase Private Key Generator – PKG generates its own master private and public key using RSA algorithm. Also, fingerprint information is provided by each node to PKG. In communication phase, actual messages are transmitted between nodes.

Phase 1: Initialization

Before starting a network, this is an initial phase. Any node from the participating nodes can work as private key generator known as PKG. Following steps are executed for generation of the public key and private key by PKG.

Step 1: PKG will generate the master public key and master private key of each node using RSA algorithm.

A. Choose two distinct prime numbers p and q.

B. Compute n = pq.

C. Compute $\varphi(n) = \varphi(p)\varphi(q) = (p-1)(q-1) = n - (p + q - 1)$, where φ is Euler's Totient function. This value is kept private.

D. Choose an integer e such that $1 < e < \varphi(n)$ and $\gcd(e, \varphi(n)) = 1$; i.e., e and $\varphi(n)$ are co-prime.

E. Solve for d given $d \cdot e \equiv 1 \pmod{\varphi(n)}$

'd' is master private key and 'e' is master public key.

Each node provides its fingerprint data to PKG.

Phase 2: Communication

Whenever any node wants to communicate with another node, it should have the public key of receiving node to encrypt the message. To get the Public key of the receiver, Sender node sends a request to PKG. The process can be defined in following 4 steps.

Step 1 PKG processes on the fingerprint of receiver node and generates identification key.

The following process is performed on the captured data of fingerprint:

- Histogram Equalization
- Binarization
- Thinning
- Minutia Generation
- Mean of Ridge ending & Bifurcation
- Identification key generation

Step 2 Elliptic curve is generated from input of identification key and master private key of receiver node.

Curve used is:

$$E: y^2 = x^3 + ax + b \text{……..[1]}$$

Public key is Q = d * P

d = a random number selected within the range of (1 to n-1).

P is a point on the curve.

Q is the public key and 'd' is the private key.

Domain parameters generation using fingerprint

Domain parameters define the elliptic curve. These parameters need to be selected carefully so as to make a resulted elliptic curve and hence ECDLP to be hard and resistant to all known attacks. As discussed in Chapter 4, Following are selected as domain parameters:

$$D = (q, FR, a, b, G, n, h) \text{ ……..[2]}$$

- q is the field size (q=p) p is prime.
- FR shows the method – field representation. FR as F_q a Prime Field is used

- a, b: Two coefficients which define the curve equation. Both a and b ϵ F_q. For example in the case of a prime field:

$$y^2 = x^3 + ax + b \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots[3]$$

Both these parameters are selected as under:

- a = Identity key derived from the fingerprint of a node as per detailed in chapter 5
- b = Master Private Key of a node
- G: The base point or the generator point having two field elements x_g and y_g in F_q. G has a prime order.
- n: The order of the base point. Such that $n.G = \mathcal{O}$
- h: co-factor, where h= #E(q) /n , where #E(q) is the number of points on the curve.

Such domain parameters are validated using Algorithm for domain parameters verification.
Step 3 Public key is sent to sender node and the private key is sent to the receiver node.
Step 4 Encryption using the public key of a receiver by a sender. The receiver decrypts the data using its own private key.

2.3 Details of Implementation
Using the generated parameters, an elliptic curve is formed which is used for further cryptographic operations. Like all public key cryptography schemes, in this scheme also a key pair of public key and a private key is generated. The private key is kept a secret with the node while the public key is sent to other nodes. These nodes on receiving the public key, validate the received key by running Algorithm on Public Key Validation.
One can perform various cryptographic operations using generated elliptic curve parameters and key pairs.

III. Simulation and Result

Above scheme is implemented in Java. Elliptic curve key generation is tested for 112 bit, 160 bit, and 256-bit input. Two types of input for encryption are considered: First, in the form of the text message and second, as a file of data to be encrypted.
Figure 2 shows the basic scheme for key generation. When node A wants to communicate with node B, the following steps will be executed:
Step 1: Node A sends the identity of node B to PKG.
Step 2: On receiving ID of B, PKG generates a Private key of B i.e. PR_b. From this generated private key, PKG generates Public Key of node B i.e. PU_b. PU_b will be sent to node A and PR_b will be sent to node B.
Step 3: On receiving PU_b, node A will encrypt the message using this key and will send to node B. Node B will decrypt this message using its private key PR_b

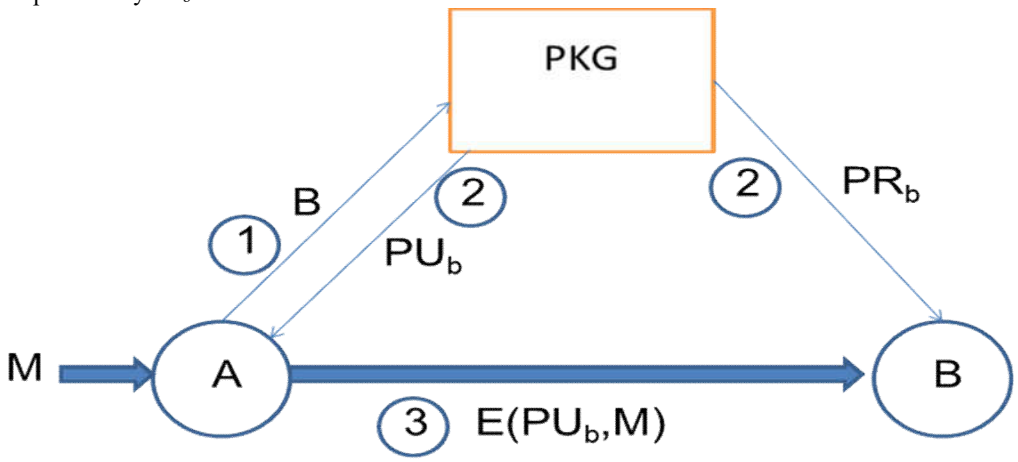

Fig 2 Key Generation Steps

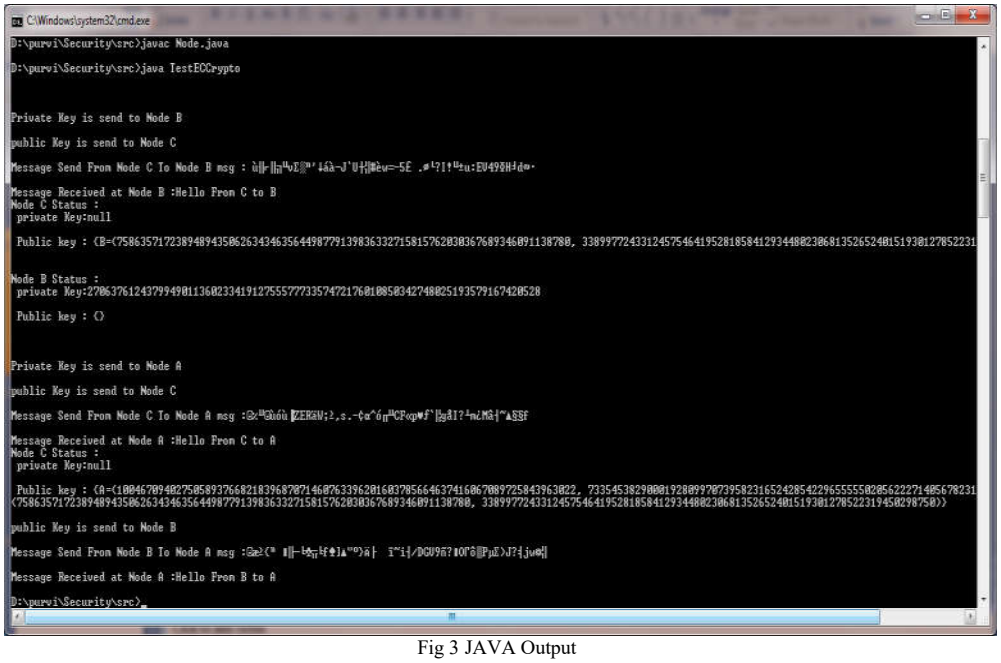

Fig 3 JAVA Output

Private Key is send to Node B

Public Key is send to Node C

Message Send From Node C To Node B msg : ,N+ª5|?6—Zíe $ÌØÏäÖà æ-
:'Õá¢KâÇ,_1ÇWôžëŽ¹

Message Received at Node B :Hello From C to B

Node C Status : private Key:null
Public key :
{B=(75863571723894894350626343463564498779139836332715815762030367689346091138780,
338997724331245754641952818584129344802306813526524015193012785223194502987 50)}

Node B Status : private
Key:27063761243799490113602334191275557773357472176010850342748025193579167420528
Public key : {}

Fig 4 JAVA Program Output for Node C to B Communication

```
Private Key is send to Node A

public Key is send to Node C

Message Send From Node C To Node A msg :W/DJ #üqÕ{;?acÛºM
z‡{Ã¡¤}Þßµ€X4ð©3?Õ ¶ÝU³Ê£

Message Received at Node A :Hello From C to A

Node C Status : private Key:null
Public key :
{A=(100467094027505893766821839687071460763396201603785664637416067089725843963022,
733545382900019280997073958231652428542296555550205622271405678231203992003 89),
B=(75863571723894894350626343463564498779139836332715815762030367689346091138780,
338997724331245754641952818584129344802306813526524015193012785223194502 98750)}
```

Fig 5: JAVA Program Output for Node C to A Communication

File of the message as input and output:

The graphical user interface is created for ease of input. On click of Encrypt, File of the message is inputted to encrypt. Key generation and encryption are done. On click of Decrypt, the message will be decrypted and stored in a file.

We have selected plaintext data of 100 bytes, just for the practical purpose. Larger data size shall only result in a higher amount of time for encryption and decryption. All other operations are independent of data being encrypted.

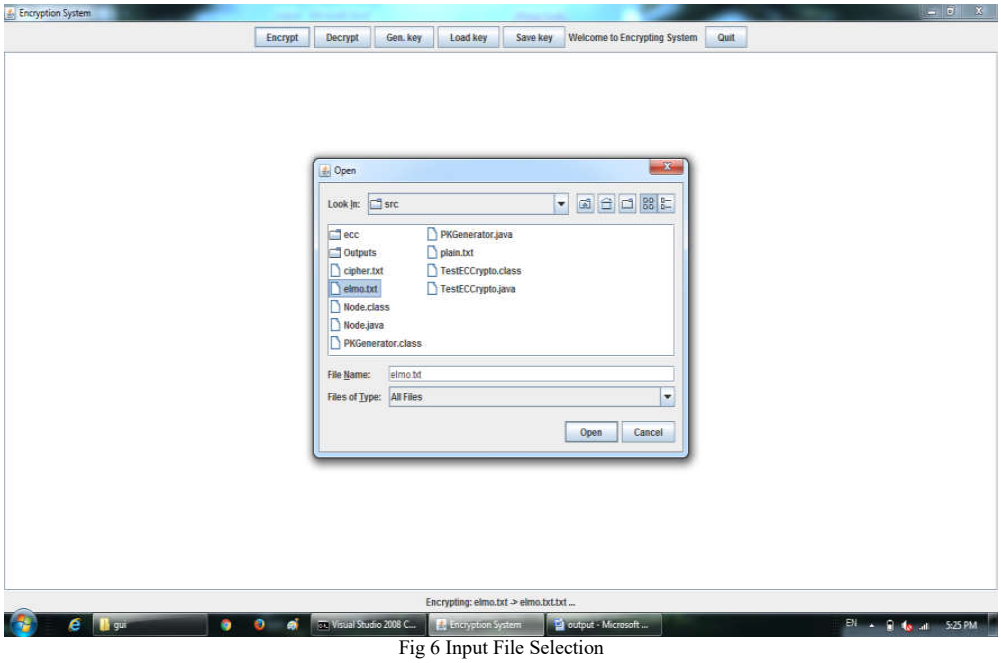

Fig 6 Input File Selection

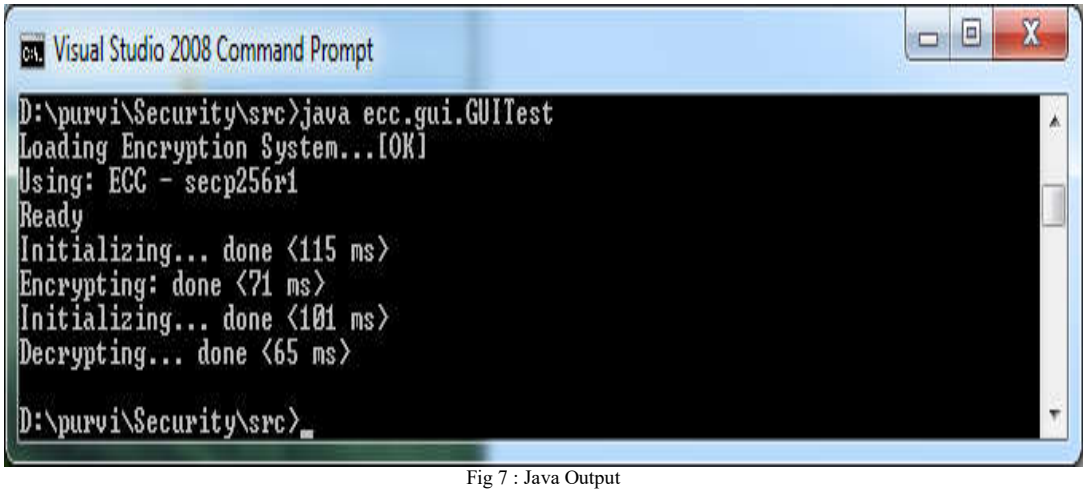

Fig 7 : Java Output

Time elapsed at each stage in the entire process is measured and recorded as under

Phase of Network	Process	Time*		
		112 Bit ECC	160 Bit ECC	256 Bit ECC
Initialization of Network	Elliptic curve Generation	63	78	109
	Key pair generation time	5	5	6
Communication in Network running mode	Encryption of Message	46	58	71
	Decryption of Message	41	44	65

* Time in Milliseconds Message size: 100 bytes

Table 1 Time Analysis of Proposed Scheme in JAVA

Plaintext data of 100 bytes is selected, just for the practical purpose. Larger data size shall only result in a higher amount of time for encryption and decryption. All other operations are independent of data being encrypted.

IV. CONCLUSION

In this paper generation of a key pair with robust security was intended. It is demonstrated that two levels security can be created using RSA and ECC by employing user's fingerprint details as an identity. Various cryptographic operations like encryption, decryption, digital signature etc. can be performed using ECDSA and ECIES algorithms using such generated keys.

Use of fingerprint, to produce key pair consisting of public and private keys enhances the intended security. Fingerprint and Private Key of a node are used to generate a unique curve. Trapdoor functionality of private key and uniqueness of fingerprint helps in providing a robust security solution. There is no algorithm to calculate or predict the fingerprint pattern. In this way, a two-level security is provided by the present scheme.

Major accomplishments of the paper are listed under:

- A security scheme with key generation is defined
- Proposed scheme provides a secured identity-based scheme which includes key generation using the identity of user and modification in Key revocation scheme.
- The identity-based scheme has the advantage of no certificate management. Processing power, storage space requirement, and communication bandwidth are much lower compared to the asymmetric key based scheme.
- Two levels of security are provided by employing RSA parameters and user's identification for generation of secure ECC curve.

REFERENCES

[1] De Loach, J. W. (2000). Enterprise-wide Risk Management: Strategies for linking risk and opportunity. London: Financial Times/Prentice Hall

[2] S. William, "Cryptography and Network Security: Principles and Practice", 2nd edition, Prentice-Hall, Inc., 1999

[3] Jason Andress, The Basics of Information Security, Second Edition, United States of America: Elsevier; 2013, 5p

[4] A. Menezes, P. van Oorschot and S. Vanstone "Handbook of Applied Cryptography" © 1997 by CRC Press, Inc

[5] Chen, L., et al.: Certification of public keys within an identity based system. In: Chan, A.H., Gligor, V.D. (eds.) ISC 2002. LNCS, vol. 2433, pp. 322–333. Springer, Heidelberg (2002)