Elliptic Curve Cryptosystem for Data Confidentiality in Cloud Environment

Abhijit Mitra, Amitesh Anand, Shyam Kumar Jha, Gourab Suhasaria

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
amitesh.unique@gmail.com

***Abstract*– In a cloud environment data may be stored at varied locations, both physically and geographically. It is not desirable to encrypt the same data multiple times while it is in transit from one location to another. In this paper, we introduce an approach for storing and retrieving data in a cloud securely, without the need for re-encryption.**

An important factor is the key strength, i.e. the difficulty in breaking the key and retrieving the plain text. In our scheme we have used 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 a strong and efficient elliptic curve cryptosystem for secure transmission or exchange of confidential data over a public network.

***Keywords*– Elliptic curve, Data Confidentiality, Cloud, Cloud Service Provider.**

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

A. 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]

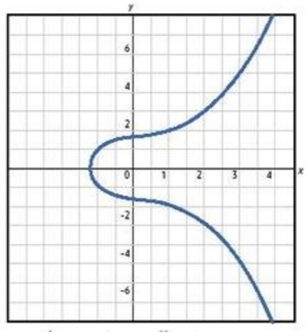

Fig. 1 An elliptic curve

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]

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

TABLE I
COMPARISON OF KEY SIZES OF RSA AND ECC

B. 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.

C. 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 $(a, b) \in F_2^m$.
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 $F_2{}^m$.

D. 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. CLOUD COMPUTING

A. Overview

Cloud computing refers to the provision of computational resources on demand via a computer network. It fundamentally allows for a functional separation between the resources used and the user's computer. The computing resources may or may not reside on the local network; for example in an internet-connected datacenter. Cloud architecture, the systems architecture of the software systems involved in th delivery of cloud computing, typically involves multiple cloud components communicating with each other over application programming interfaces. It is like having multiple programs, each doing one thing well and working together over universal interfaces. Complexity is controlled and the resulting systems are more manageable then their monolithic counterparts.

There are many issues involved in Cloud Environment, some of them are listed below:

- Non-Assertion
- Cloud Provider Espionage
- Data Lock in

Data confidentiality is a major issue, as the user's data may lie beyond its own premises. The issues can be dealt in specific details, by categorizing the threat sources:

- "**Co-Users**" of the Cloud Service
- Third-Party Cloud "**Service Provider**"
- On the "**Public Network**"

B. Current Practice for Data Confidentiality

Organization using the cloud services, stores its confidential encrypted data in the Cloud DataStore. When User wants to search any data from the Cloud, he encrypts its request using the same key given by the Organization. Now for example, User wants to search for data "A". He encrypts "A" to "X" and sends to Cloud Service Provider. "X" is searched in the Data Store. The requested data is found and sent back to the User, without any changes. The User decrypts "X" to get back "A".

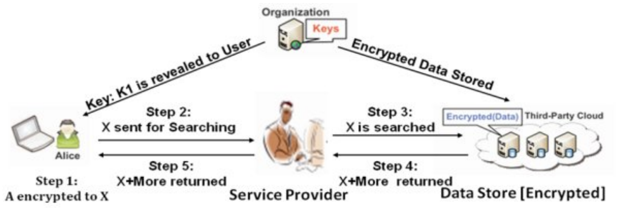

Fig. 2 – Current Practice

There lies a very subtle security issue:

- The "Key" used for data encryption in data store is not changed frequently, therefore the need for re-encryption arises for data-exchange.

- As Cloud works in a distributed environment so the following steps pose a serious security risk
  i) Step 2 & 3 are in the same format
  ii) Step 4 & 5 are in the same format
- If a third person listens to the repeated request-reply message exchange, the data confidentiality can be compromised.

We will be dealing with all the issues with a better approach in this paper, leading to improved security.

IV. PROPOSED WORK ON DATA-STORAGE AND DATA-RETRIVAL

A. *Overview*

ECC provides a way to protect the confidentiality of the sensitive data. In a cloud environment we would like to have data encryption policy which does not limit the functionality of the cloud applications. In this paper, we introduce an approach for storing and retrieving data in a cloud securely. The procedure is proposed below:

There are three entities involved in viz. i)User, ii)Organization and iii)Third party Cloud. The Third party cloud service is constituted of two entities, namely- i) Service Provider and ii) Data Store.

Encryption:

Before storing the data- P_m in the Cloud, it is encrypted as: $\{s*P_m + sG\}$ and the key $\{s\}$ and G (the fixed point, an encryption parameter) is stored in the organization.

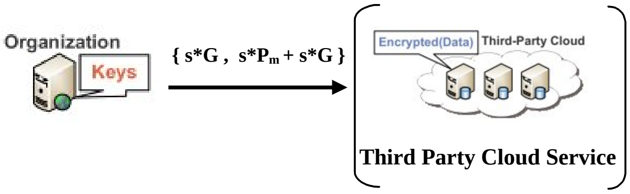

Fig. 3 – Encryption Phase

Encrypted data is stored in the Third-Party Cloud, along with Public-Key $\{s*G\}$ for all of the data encrypted with the same Private-key $\{s\}$.

Accessing the data:

Once data has been uploaded in the Third-Party Cloud along with its Public-Key, it is quite ready to be accessed by the legitimate user. After authentication of the user by the Organization, the Private-Key and the encryption parameter G, is sent to the User using Key-Exchange mechanisms.

Step -1: Encrypting the Data Request (sent to Service Provider)

The User now encrypts the data to be searched in the database, only partially.

- The User maps the Plain-text corresponding to the 256 points mapped to P_m.
- P_m is now multiplied with the Private-Key, i.e. $\{s*P_m\}$ and sent to the Cloud Service Provider.

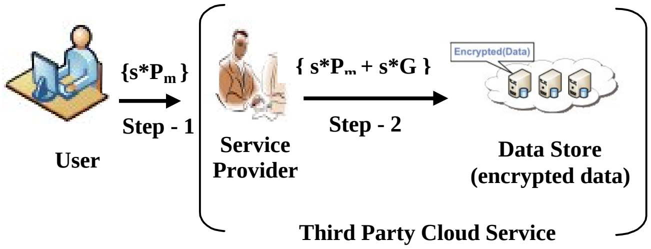

Fig. 4 – Request for Encrypted Data

Step -2: Receiving the Data Request and searching in Data Store.

The Cloud Service Provider, upon receiving the partially encrypted data $\{s*P_m\}$,

- encrypts it to the data corresponding to the data present in the Cloud, by adding the Public Key $\{s*G\}$ (which was stored by the Organization corresponding to the data) to the received request $\{s*P_m\}$.
- Searches the finally encrypted data $\{s*P_m + s*G\}$
- The encrypted data found in the data store is sent back to the user.

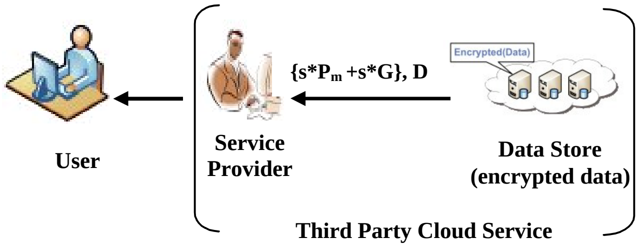

Fig. 5 – Data sent back to User after searching. Here D is the data requested by the user, in response to P_m .

Step - 3: Data Decryption in the User's System.

The User now decrypts the data received from the Third Party Cloud Service, by the following procedure:

- The $\mathbf{\{sG\}}$ is subtracted from the received cipher text $\mathbf{\{s*P_m + s*G\}}$
- Inverse-Modular of the private key is multiplied to $\mathbf{s*P_m * s^{-1}}$ to give $\mathbf{P_m}$
- The User maps the message $\mathbf{P_m}$ back to Plain-text

Here **D** is the additional data which is obtained in response to the searched data- $\mathbf{P_m}$, and can be decrypted similarly.

In this way, a very secure data can be searched and transmitted from an un-trusted Third Party Cloud Service. The figure given below describes the process.

As the data to be searched in the data store is in encrypted form, thus all the requests should ultimately be converted to the same form in which it has been encrypted using the particular key, this encryption scheme uses only a single key for both encryption and decryption. We can identify the data to be encrypted, we must choose how many keys to use for encryption, and the granularity of encryption. In the simplest case, we can encrypt all such data using a single key, and share the key with all users of the service.

In the other extreme, we could encrypt each data object with a different key. This increases robustness to key compromise, but drastically increases key management complexity. Thus we need to automatically infer the right granularity for data encryption that provides the best tradeoff between robustness and management complexity. The data can be partitioned into subsets, where each data subset is accessed by the same group of users. We can then encrypt each data subset using a different key, and distribute keys to groups of users that should have access (based on desired access control policies). [7]

Even if an intruder listens the public line for both request and reply, it is impossible to retrieve any information. The reply packet contains the request string, but no meaningful similarity can be found between them. Our method wards off the risk of data confidentiality both from a malicious co-user and the untrusted cloud, as no keys are stored in the cloud. Even if cloud stores **{sG},** there is no way to obtain the key **{s}.** [8]

Users are given decryption keys by the organization to provide users with transparent data access. Of course, these users must protect these keys from compromise.

B. Performance against Attacks

The proposed system is tolerant to a number of common computer and network security attacks and flaws. [5], [9]

- In case Cloud Provider Espionage, the confidential data lies unusable as it does not have the key.
- 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 cloud service provider. 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.

C. Advantages

The main advantage of this procedure is that it ensures very secure mode of transmission. In this system:

I] each transmission uses ECC and hence gives better security with small key sizes.

II] sniffing does not provide any clue as both the request and reply data are sent differently, making it all the more secure for any outside user to make any guess about the data.

III] the encryption overhead is shared by both User and Cloud Service Provider. So none of them is under immense pressure. It saves us from the need of re-encryption.

IV] if all the data is divided into subsets and unique key is assigned to each one of them which in turn provides more granularity for data confidentiality.

D. 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 the user groups and other servers. Searching encrypted data is a bit complex in nature.

V. 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. The advantages offered by ECC can be important in environments where processing power, storage, bandwidth, or power consumption is constrained.

ECC is based on discrete mathematics, so changing any one of the parameters even linearly gives a non-linear variation in result, which proves out to be suitable for making it a strong competitor for use in secure transmission of confidential information over a vulnerable network, especially in Cloud Computing Environment and saves us from the need of re-encryption. Although, this implementation requires more time and storage space, it gives very high level of security. So this procedure is very useful when highly confidential data is to be stored in the Cloud.

VI. 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] SCHNEIER BRUCE (1996), *Applied Cryotography,* 2nd ed. Usa: John Wiley & Sons, p53-62

[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.

[7] Krishna P. N. Puttaswamy, Christopher Kruegel, and Ben Y. Zhao, Silverline: Toward Data Confidentiality in Third-Party Clouds, Computer Science Department, UC Santa Barbara.

[8] Stephen S. Yau and Ho G. An., Confidentiality Protection in Cloud Computing Systems, Int J Software Informatics, Vol.4, No.4, December 2010, page 351–365.

[9] Armbrust, M., Fox, A., Griffith, R. et al. Above the Clouds: A Berkeley View of Cloud Computing. UCB/EECS-2009-28, EECS Department, University of California, Berkeley, 2009.