DOI: http://dx.doi.org/10.26483/ijarcs.v8i7.4465

ISSN No. 0976-5697

Volume 8, No. 7, July – August 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

SECURE HYBRID ENCRYPTION USING ECC IN CLOUDS

Shaik Danisha
Department of CSE
JNTUA college of Engineering, Ananthapuramu
Andhra Pradesh, India.

C. Shoba Bindu
JNTUA college of Engineering, Ananthapuramu
Andhra Pradesh, India.

P. Dileep Kumar Reddy
JNTUA college of Engineering, Ananthapuramu
Andhra Pradesh, India.

Abstract: The evolution of the cloud computing furnishes a better way for the data storage. ABE with delegation is accustomed for accessing the data and keep it confidential. But this scheme faces some limitations like the cloud servers can replace the cipher text with some malignant intent. They may also cheat the eligible users as unauthorized ones. The cloud servers even face external threats like the security attacks. So, to protect against the security attacks and to preserve the data integrity in the cloud, a hybrid encryption with VD scheme is proposed by Xu et al. Even this scheme faces the challenges like having larger keys, more computational time and vulnerability to man-in-the-middle attacks. A secure Hybrid Encryption using ECC in clouds is proposed to overcome these problems. The application of Elliptic Curve Cryptography and its algorithms in the proposed work provides a greater security because of the Elliptic Curve Discrete Logarithm Problem. The proposed work ensures greater security with the aid of a smaller key size. It also ensures less computational time, less memory and communication bandwidth.

Keywords:Elliptic Curve Cryptography, Elliptic Curve Diffie-Hellman, Elliptic Curve Integrated Encryption Scheme, Elliptic curve Digital Signature Algorithm, Verifiable Delegation(VD), Hybrid Encryption.

I. INTRODUCTION

Cloud Computing provides a better way of the data storage. Cloud servers store a larger amount of shared data which the authorized users can access. It allows the users to deploy the data. Thus, helping the end user to relieve the task of storage of data and cost of using hardware or software. The users who are authorized can share and access the data which is mounted in the clouds. Whilst there are tremendous benefits of using the clouds, the major concerns focuses on providing security against the attacks like CPA, CCA[1]. Preserving the data integrity is also a major task.

Users with confined computing power, delegate a part of decryption to cloud servers. Providing the delegation is favourable, but it suffers from problems like tampered results, responding the appropriate users as unauthorized ones due to some malicious intended cloud servers [1]. A circuit Cipher-text Policy Attribute-Based Hybrid Encryption with Verifiable Delegation by Xu et al.[1] overcomes the problems. Still, this scheme faces the issues of having larger keys, more computation time. The representation of circuits with the attributes to form a cipher text policy is a harder task. The Diffie-Hellman protocol used for Key exchange can be vulnerable to man-in-the-middle attacks as it doesn't authenticate the users.

A Secure Hybrid Encryption Using ECC in Clouds focuses on overcoming the issues. The proposed work uses the variants of ECC algorithms. Thus ensuring a greater security with the benefit of smaller keys, providing Perfect Forward Secrecy and less consumption of time and memory.

The rest of the paper is organized as follows: section 2 describes the problem statement, section 3 describes proposed work, section 4 analyses the security of the system, section 5 describes the efficiency analysis and section 6 gives a brief conclusion.

II. RELATED WORK

The cloud servers can handle and calculate numerous amounts of data according to the user's needs. With the increasing needs of data, the data must be accessed in a simplified way. The attribute based encryption is accustomed to keep the data confidential. Attribute-Based Encryption is classified into 2 variants: Key-Policy ABE and Cipher-text Policy ABE.

A. CP-ABE for circuits

The access structure of the required data is included in the cipher-text. The data owner holds and defines the access policy. The key issuer includes the attributes for the users in the private key[1]&[2]. So, each cipher-text composes of the access policy and the attributes of the data in the private key. An end user decrypts the data iff he satisfies the attributes in the private key.

The general circuits are employed to represent the access structure. The attributes of the access policy are represented in the circuits as leaf nodes using AND, OR, NOT gates[1]. An end user can access the data if his attributes matches with those in the policy. For example, if the data owner uploads a file giving the access policy as ("Student" AND "M.Tech" OR "B.Tech" AND "Grade=A"), then the end-user should satisfy the above policy so as to decrypt the data.

The difficulty in this scheme is implementing the policies over the circuits. Even though it provides a strong mechanism of access control, it has some practical issues.

As for decrypting with the circuits, the one who partially decrypts gets to know easily about the decryption value. Moreover, it is susceptible of backtracking attack.

B. Hybrid VD-CPABE

The major issue in the clouds is that of a malicious cloud server. The users who have low computing power usually outsource a part of decryption to the cloud server. But, the malicious intended cloud servers make misuse of this for their own benefits. To overcome this, Xu et al's Hybrid VD-CPABE scheme is used. The hybrid encryption system guarantees that the untrustworthy cloud server doesn't know anything regarding the encrypted message. Here, the hybrid encryption is specified for 2 main reasons: (i) The circuit for ABE is a bit encrypted one. (ii) To authenticate the cipher-text which is delegated to the cloud servers to decrypt[1].

Hybrid Encryption is a public cryptosystem comprising of the efficacy of the symmetric key cryptographic system. This can be constructed using Key Encapsulation Mechanism (KEM) and the Data Encapsulation Mechanism (DEM)[2]. KEM is a public-key cryptosystem. The KEM consists of the CP-ABE for the circuits. DEM is a symmetric-key cryptosystem. The DEM consists of symmetric encryption with encrypt-then-MAC mechanism. Each KEM takes a random Group element and encrypts it. Then it maps it through some key derivation functions to form a symmetric key 'dk' and a one-time verified key 'vk'.With the help of 'dk ' and 'vk', data is encrypted and verified. These two parts form the Hybrid VD-CPABE scheme[1]. It proves to be reliable against the selective chosen plain-text attacks (IND-CPA) under k-multilinear Decisional Diffie-Hellman assumption using Cipher-text policy attribute based encryption (CP-ABE) for circuits. It proves to be secure upon the selective chosen cipher-text attacks (IND-CCA) using Authenticated Encryption. This also provides verifiable delegation using the encrypt-then-MAC mechanism.

This scheme faces some limitations due to the techniques used in it. First of all, this includes large keys to encrypt a larger message. The size of the key increases with that of the data. Thus, implementing such large values can be more complex. Second, the Diffie-Hellman is prone to man-in-the middle attack as it doesn't authenticate the users. Third, it consumes greater amount of time to compute.

III. PROPOSED WORK

The emerging expansion of the information systems has made revelatory advances in the cryptographic systems to provide the data confidentiality and integrity. Elliptic Curve Cryptographic (ECC) technique is emerging as an alluring public-key cryptosystem. Unlike familiar cryptosystems like RSA, ECC ensures same level of security with smaller keys. Thus, it provides faster computation, less memory and bandwidth savings.

The proposed work uses Secure Hybrid Encryption using ECC to overcome the limitations of Xu et al. scheme. ECC is a public-cryptosystem defined over finite fields on the basis of algebraic structures of the elliptic curves. The Elliptic curve cryptography is defined on the supposition that the elliptic curve discrete logarithm problem (ECDLP) is very difficult. ECDLP is determining the integer k, given a rational point P on the elliptic curve E and the value of 'k*P'[4][5]. Elliptic curve cryptosystems rely on the hardness of solving the ECDLP.

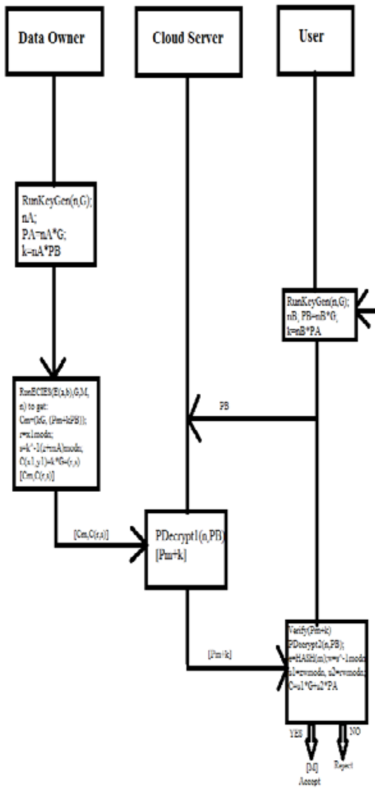
Data Owner
Cloud Server
User
Authority
SetUp(p,a,b,G,n,h)
RunKeyGen(n,G); nA; PA=nA*G; k=nA*PB
RunKeyGen(n,G); nB, PB=nB*G; k=nB*PA
PB
RunECIES(E(a,b),G,M,n) to get: Cm=(kG, (Pm+kPB)); r=x1modn; s=k^-1(z+rnA)modn; C(x1,y1)=k*G=(r,s) [Cm,C(r,s)]
[Cm,C(r,s)]
PDecrypt1(n,PB) [Pm+k]
[Pm+k]
Verify(Pm+k) PDecrypt2(n,PB); z=HASH(m);w=s^-1modn u1=zwmodn, u2=rwmodn; C=u1*G+u2*PA
YES
NO
[M] Accept
Reject

Fig1: Architecture of Secure Hybrid ECC scheme

The proposed work consists of the following 5 phases:

A. *SetUp phase*

To work with ECC, all the users must agree upon the elements defining the curve. These are called as "Domain Parameters"[6]. They are 6 distinct values : (p, a, b, G, n, h) defined as:

p : Prime numbers defining the field in which curve operates

a, b: Integer co-efficients of the curve

G : Generator

n : Order of G (Curve Generator Point)

h : Co-factor of curve

Authority ensures that all the users agree on the domain parameters.

B. *Key Generation Phase*

After agreeing on the domain parameters, a shared secret key is produced using Elliptic Curve Diffie-Hellman(ECDH) to form a secure communication channel. ECDH is the elliptic curve analog of the Diffie-Hellman key exchange. It is an anonymous key agreement protocol used to form a shared secret through an unsecured channel. This shared secret is shared across two parties who have elliptic curve public-private key pair[4]. The sender will encrypt the messages using public key of receiver and the receiver will decrypt it using his/her private key.

The shared secret between the data owner and the user is formed as follows[6][8]:

(i) A selects an integer ' nA<n '. This will be A's private key. Then, he generates public key as "PA = nA * G ".

(ii) B similarly selects private key ‘nB’ and computes public key “PB“.

(iii) Now, A generates the secret key as:

“ K = nA * PB “. And, B generates the secret key as: “ K = nB * PA “.

The shared secret is then used for further communication. This is proven to be secure as no party can derive the private key unless they can solve ECDLP.

C. *Encryption Phase*

Elliptic Curve Integrated Encryption Scheme(ECIES)is used for encrypting and decrypting the messages. ECIES combines elliptic curve asymmetric encryption and AES symmetric encryption algorithm with SHA-1 hash algorithm as such to provide ease of using encryption scheme along with authentication support[5].

The plain text ‘M’ is converted as a point ‘Pm’ on the elliptic curve Ep(a, b) so that ‘nG = O’ is a large prime number for the smallest value of n. The Generator G and Ep are made public[8]. To transmit a message ‘Pm’ to B, A selects a random positive integer ‘k’ and yields the cipher text comprising of a pair of points

“Cm = (kG, (Pm + kPB)) “.

D. *Decryption Phase*

To decrypt the message, B first multiplies the foremost point of cipher text by B’s secret key and then deducts the result from the second point. This gives the original message ‘M’.

“M = ((Pm+kPB) – (nB (kG))) “

The decryption phase comprises of two partial decryptions i.e., PDecrypt1 at the cloud server and PDecrypt2 at the user to obtain the original message. A has concealed the message Pm by adding kPB to it. As the value of k is unknown, even though PB is a public-key, it is hard to unmask ‘kPB’.

E. *Authentication Phase*

Elliptic Curve Digital Signature Algorithm (ECDSA) is used to provide authentication. It is a variant of the Digital Signature Algorithm using the elliptic curve cryptography. The signatures are created and verified using it. To provide a security of 80 bits, ECDSA would require a 160-bit public key whereas other DSA would require at least 1024-bit public key[5][9]. This shows that an attacker have to perform 2^80 operations to find out the private key. ECDSA comprises of Signature generation and verification algorithms to provide the authentication.

(i)*Signature Generation:* Suppose A wants to transmit a signed message to B, first they must coincide on the curve parameters (E, G, n). Now, for A to sign a message, it proceeds as follows[6][7]:

a. Calculate e = HASH (m).
b. Assume ‘z’ to be the Ln leftmost bit of e where Ln is the bit length of group order n.
c. Select a random integer ‘k’.
d. Calculate the point on the curve C(x1, y1) as
 C = k * G.
e. Calculate “r = x1modn”. If r = 0, then select another ‘k’ and repeat.
f. Calculate “s =k^-1(z + r.nA)modn”.
 If s = 0, choose another ‘k’ and try anew.
g. The pair “ (r, s) “ is the signature

Using this signature pair, A signs the message.

(ii)*Signature Verification:* To check the validity of the signature, B first checks that[6]:

- Public key (PA) is not equal to O.
- PA lies on the curve.
- n * PA = O.

This is done to confirm that the public key of A is a valid curve point or not. Then to verify the signature, B does the following[6][7]:

a. Verify that ‘r’ and ‘s’ are integers in [1,n-1].
b. Calculate e = HASH (m).
c. Assume ‘z’ to be the Ln leftmost bit of e.
d. Calculate w = s^{-1} modn.
e. Calculate u = (zw) mod n and
 u2 = (rw) mod n .
f. Calculate the point on the curve C(x1, y1) as C = u1*G + u2*PA.
g. The signature is a valid one if r = x1modn i.e., if
 C = k*G.

Thus signing a message and verifying it using the points on the curve makes it hard to break it.

The proposed scheme (Fig:1) works as follows:

1. First of all, Authority runs SetUp algorithm so that all the parties agree on the same domain parameters.
2. Then, KeyGen algorithm is run at the data owner and the user to form a shared secret key for further communication.
3. If the data owner wants to upload a file, he runs the encryption algorithm and the signature generation algorithm to encrypt the data and to provide authentication.
4. The user who wants to access the data, sends a request for data along with his public key to the cloud server.
5. Then, the cloud server partially decrypts (PDecrypt1) the data using the user’s public key. The decrypted message with signature is sent to the user.
6. The user then decrypts (PDecrypt2) the data and verifies the signature using signature verification algorithm. If the user is authenticated, then he has the access to the data.

IV. SECURITY ANALYSIS

In this section, we present that our scheme is secure:

A. Man-in-the-middle Attack:It is an attack where an adversary delays the communication between two parties and alters it secretly while the two parties believe that they are in a direct communication. To overcome this attack, a process of authentication should be used [4]. By using ECDH with a signing algorithm (ECDSA) overcomes the attack in our scheme. Moreover, ECDH relies on the hardness of solving ECDLP.

*B. Collusion Attack:*This attack refers to combining of several copies to form a new copy. As the ECIES deals with the points on the curve to encrypt /decrypt the data, it doesn’t give a scope to the attacker to perform this attack.

C. Perfect Forward Secrecy: Perfect Forward Secrecy refers to preserve the past sessions against the future negotiations of secret key. In our scheme, even if an adversary knows G and Public key, it is hard to determine ‘k’ because of the ECDLP. So, our scheme provides Perfect Forward Secrecy as we use a new random ‘k’ for every message. Therefore, the ‘k’ value changes for every message making it hard for an adversary to find it.

V. EFFICIENCY ANALYSIS

Here we demonstrate the comparison of efficacy between Xu et al.’s scheme and our proposed scheme. Table 1 gives the comparisons.

It shows that efficiency of the proposed scheme is efficient than Xu et al.’s scheme. The exponential, hash and XOR operations are reduced to simple multiplications and addition operations.

Table 1: Comparison of efficiency between Xu et al.’s scheme and our scheme

Phases	***Comparision of schemes***	
	Xu et al’s Scheme	***Proposed Scheme***
Computataions of SetUp Phase	3H, 3e	0
Computations of KeyGeneration Phase	4e, 2M, 1H, if f(x)=1 4e, 2M, 1H, if f(x)=0	4M
Computations of Encryption Phase	1σ,3e,2H,1⊕,1f(), if f(x)=1 1σ,3e,2H,1⊕,1f(), if f(x)=0	2M, 1A
Computations of Decryption Phase	1e,if f(x)=1 1e, if f(x)=0	1M,1S
Computations of Authentication Phase	1E,2H,1d,1⊕,if f(x)=1 1σ,3e,2H,1⊕,1f(), if f(x)=0	1H,6M,4mod,2I,2A
:computation operation of hash function e:exponential computation operation ⊕ :XOR	E:encryption/decryption operation f(): function to find sharing of wire M: multiplication A: addition	I: inverse operation mod:modulus operation S: subtraction d: division σ: signing operation

VI. CONCLUSION

The proposed work shows that we have overcome the problems of the Xu et al scheme. The computations of the proposed work are far more reduced to minimal operations with the help of ECC and its algorithms. The computational time, cost and the execution time are also reduced. Our scheme also proves to be secure against attacks like man-in-the-middle attack, collusion attack and provides perfect forward secrecy. And it also ensures greater security with smaller key sizes.

VII. REFERENCES

[1] Jie Xu, Qiaoyan Wen, Wenmin Li and Zhengping Jin, “ Circuit Cipher-text-Policy Attribute-Based Hybrid Encryption with Verifiable Delegation in Cloud Computing” in Proc. IEEE Transactions on parallel and distributed systems, 2016.

[2] K. Kurosawa and Y. Desmedt,”A New Paradigm of Hybrid Encryption Scheme,” in Proc. CRYPTO, pp.426-442, Springer-Verlag Berlin, Heidelberg, 2004.

[3] B.Waters, “Cipher text-Policy Attribute-Based Encryption: An Expressive, Efficient and Provably Secure Realization,” inProc. PKC, pp.53-70, Springer-Verlag Berlin, Heidelberg, 2011.

[4] Rakel Haakegaard and Joanna Lang , “ The Elliptic Curve Diffie-Hellman (ECDH)”, December 2015.

[5] Kefa Rabah, “ Implementation of Elliptic Curve Diffie-Hellman and EC Encryption Schemes”, Information Technology Journal 4(2): 132-139, 2005. ISSN 1812-5638.

[6] Andrea Corbellini, “Elliptic Curve Cryptography: ECDH and ECDSA”, ,http://andrea.corbellini.name/2015/05/30/elliptic-curve-cryptography-ecdh-and-ecdsa/, Accessed on June 20, 2017.

[7] Johannes Bauer, “Elliptic Curve Cryptography Tutorial”, https://www.johannes-bauer.com/compsci/ecc/#anchor24, Accessed on June 20, 2107.

[8] William Stallings, “Cryptography and Network Security Principles and Practices”, Prentice Hall Publications, 4th Edition, 2005.

[9] V. Gayoso, Martínez, L. Hernández Encinas, and C. Sánchez Ávila,” A Survey of the Elliptic Curve Integrated Encryption Scheme”, Journal of Computer Science and Engineering,Volume2, Issue 2, August2 2011.