Exponential Cipher Based Concealed Policy In CP-Abe For Trusted Cloud Environment

A.C.Ashmita, Dr.C.Yamini

Abstract: In several access control systems of a cloud, any data can be accessed by many different users and expecting a trusted environment from those scenarios is always a challenging and complex task. Cloud server determines the access rights of the users based on their authorization. Conventional access policy using public key cryptosystem is a tedious task. Generally, the properties of the access policy are defined by attributes of a user. This paper is an enhancement of conventional CP-ABE in concealed policy, which constructs exponential ciphers based on user’s attributes. The proposed scheme supports all sorts of access structures without supporting bilinear mapping. The exponential form of ciphertext creates complexity for attackers to decrypt the message. Prior updatable CP-ABE systems were suggested on restricted access structures. The bound representation of the access tree size during setup process proved the efficiency of the systems. Additionally, encryption and decryption time were also studied to prove the flexibility and scalability of the systems.

Index Terms: Access control systems, Public key encryption, Exponential ciphers, Access Tree, and Encryption.

1. INTRODUCTION

A recent development made in Information Technologies (IT) system has enabled the growth of data and the demand for storing, processing and maintaining those data is a trivial task. With the help of cloud computing, the monitoring of those voluminous data becomes quite simpler than before. Cloud computing becomes an emerging technology among individuals and organization use. This system embraces the traditional data modelling with an added advantage of distributed, parallel and grid computing. Broad network access and services on its request describes the utilization of the cloud servers. The cloud user’s data are being stored either locally (or) remotely [1, 2]; services offered by cloud technologies have met the user’s requirement to some extent. Variant types of clouds such as private, public, hybrid and community clouds in which each of them offers limited data storage services and security policies. Coming to data security policies, survey analysis states that the security policy in cloud storage systems needs to be enhanced in terms of confidentiality and privacy, so as to avoid data breaching attacks. Privacy is the most important security parameter when it comes to cloud environment. In perspective of organizations, data represents an extremely important asset where security policies should be consistent and covers the following entities [3]. Activity patterns: It depicts the protection of sensitive data pertaining to organization entity. Capturing the networks: Network devices like routers, switches, etc. capture the data for security and traceability purpose. Audit-Trail logs: Mutual auditability between each entity for better resource utilization. In the conventional ABE system, shared attributes by users used for computing secret keys. In some cases, users with similar attributes adopt similar decryption process. Due to traceability, original key and its users were not found [4] which is a serious concern. When a user leaks his secret key to any unauthorized party, then detecting those activities is a difficult task. The paper is arranged as describing prior works in Section II that followed by proposed methodology in Section III, Section IV deals with experimental analysis and concludes in Section V.

2 RELATED WORK

Here, existing techniques explored by other researchers are discussed.

2.1 CP-ABE Scheme:

CP-ABE scheme is an extended version of the Key Policy based Attribute Based Encryption which resolved the issues of fine grained access control in shared data, in specific to, one to many communications. Though, data owners have control over access policy, improvements need to be done in cryptographic mechanisms. In recent times, it has been widely used in cloud servers because it’s trustworthy [5]. User’s attributes assist to compute secret keys and it’s combined with access tree to form ciphertext. Further, ciphertext is decrypted using secret key. CP-ABE composes of three phases, namely,

Setup: It takes public key, secret key and attributes to generate Private Key.

Encryption: It takes public key, plaintext, access policy to generate Ciphertext.

Decryption: It takes public key, private key and ciphertext and decrypts the ciphertext, as plaintext.

CP-ABDE scheme – Issues classification:

According to previous researchers, several issues are identified and discussed. The issues on attribute revocation, constant ciphertext, fine grained access control, hidden access policy, and multi-authority. Attribute revocation: It is a significant process in CP-ABE algorithm where the attributes changes according to cloud users. A new secret key is often generated based on given attributes during registration process. It has serious drawback in practical application i.e validating the secret key for live users. It holds two properties, namely, forward secrecy and collusion-resistance. It ensured the revoked user has no rights to further actions. Likewise, re-encrypting is also not possible, once the users are revoked. Hence, it is the main reason to study on attribute revocation. Jiang [6] studied the CP-ABE with Access Policy Update (APU) which supports users during policy updates. Dynamic Binary Tree was suggested by [7] to improve the scalability of the system. Then, it’s again revised by [8] with the help of matrix update algorithm for policy updates on CP-ABE.

Constant Ciphertext: This is the second issue which deals on ciphertext generation. Based on number of attributes, the ciphertext is being generated. Thus, a voluminous amount of ciphertext should be generated where it reduced the overheads [9]. Privacy preserving algorithms [10] were
suggested to protect the access policies which are impossible in some real-time applications. Henceforth, constant ciphertext schemes were constructed to preserve the access policies using Linear Secret Sharing Scheme (LSSS). It computed higher Boolean was required larger than size of the attributes. It was resolved by [10] suggested, an algorithm named, Privacy Preserving Constant-Size (PP-CS)- CP-ABE that enforced access policies with wildcards and generated constant ciphertext. Then, a constant ciphertext with constant secret key was also studied by [11] using Ordered Binary Decision Diagram (OBDD). Eventually, CP-ABE with fine grained access control [12] was explored for smaller size ciphertext generation. Likewise, author [13] extended the constant ciphertext without restriction user’s count. Dispensability matrix used to eliminate the redundant attributes. Fine Grained Access Control: It is helpful during data protection from unregistered users. Conventional CP-ABE leaked the private key to anonymous users. In [14], an improved CP-ABE was suggested in terms of verifiable delegation models which presented a concrete ciphertext and ensured the correctness of the delegated computing. Performance of CP-ABE in practical application has exposed data integrity and scalability issues. Thus, a multithreading concept [15] was introduced to devise flexibility and availability of the cloud servers. The phases in CP-ABE are processed in parallel. Switch modes like Cipher Block Chaining (CBC) and Counter (CTR) were used in encryption process. Then, fine grained properties [16] were enhanced without changing its properties of user control and access control. In [17], forward and backward securities were studied for devising delegate mechanisms. Hidden Access Policy: It represents the Boolean formula used for attributes in access tree model. Mostly, AND and OR gates were used for constructing access tree. In some cases, anonymous tries to hack the access policy for decryption process. It helps them to learn the access formula for fake key generation. Henceforth, it is important to conceal those access policies for efficient security system. This is taken as research study. Conventional schemes do not hide the access policies constructed using AND gate [18]. Several schemes were explored to hide the access policies. The author in [19] improved CP-ABE scheme by concealing the attributes and access policies. Generally, it is concealed by hash and polynomial functions. A scheme was suggested by [20, 21] where the users were not allowed for keeping access policies during encryption process. Here, a Composite order bilinear model was applied which accepted AND gate with wildcards. A searchable CP-ABE with attribute revocation mechanism was suggested to protect the sensitive information from decryptor’s using linear assumptions. But the security of the system is not improved. Multi- Authority: It comprises of authority for a user and multiple users. In recent times, multi-authority was explored by several researchers due to its practical possibilities. Anyway, it possesses two issues which depend on cloud servers. First issue is that any user can join (or) leave the cloud system at any time where the users can modify the authorization. Second thing is that insert and revoking attributes can also alters at any time which helps for altering the setup process [23]. System failed on processing the high number of corrupted users. Likewise, users rate increases, decrypting process rate decreases. When a number of users increase, the efficiency of decrypting the ciphertext decreases. It was resolved by [24] using threshold based key generation approaches. In [25, 26], multiple authorities were allowed to control the key using exponential number of attributes and thus, ensured security at initial phase. Similar extended, using fast decryption models which rapidly increases the computational time of decryption process. In [27], multi-authority models were described by devising the owners of attributes. Authorized users shall remove the revoked users. Here, decryption process is complex due to incessantly updating secret keys. Decentralized multi-authority scheme was done [28] that any party shall generated ciphertext using the attributes.

3. PROPOSED MODEL

This section covers the primitives and proposed framework of the research study.

3.1 Primitives:
These are the basic entities of an enhanced CP-ABE and explained below. Key generation center: Secret keys and public keys are generated. It helps the users to issue, revoke and updating the keys based on given attributes. Registered users are allowed for accessing the policies. Mostly, it is trustworthy, in some cases, it tries to learn the data from encrypted contents. Data storing center: It assists for data sharing services. It restricts the access rights for revoked and unauthorized users. It integrates with KGC for generating and storing the keys with the help of fine grained access controls. Data owner: It’s the users who own the data. The users shall upload (or) download the data from anywhere at any time. It, purely, works on defined attribute access policy. User: It is an entity that accesses the data on its demand. Decryption process is done by authorized users.

3.2 Proposed framework: Exponential cipher based concealed policy in CP-ABE.
In our construction, secret keys will be identified and generated using exponential functions with a set of attributes. If any users desire to decrypt the data, the access tree and its concealed policy have to be satisfied. Prior access tree structure is employed with an enhancement in generating the ciphertext. Here, exponential function is used for creating the secret keys. The proposed EC-CP-ABE is explained as follows:

**Setup:** In this step, access tree structure for concealed policy is defined. Let \( T \) be the tree structure with leaf and non-leaf nodes. Each non-leaf nodes is AND (or) OR gate. Each record is administered with an index value and for given attributes, the secret keys are generated. For given attributes \( A \), the setup \( (A) = (S_i, P_i) \) where \( S_i \) is a secret key and \( P_i \) is a public key. Then, select the finite field \( Z \) for each attributes \( Z = \{A_1, ..., A_n\} \) in an exponential form. The public parameters \( P_i \) are given as:

\[
P_k = e(g_i, z_i)^{\sum_{x \in \text{path}(T)} x_{i,j}}
\]

Where \( T \) is access tree structure, \( e \)-Exponential function \( j, x \) are users and its index value.

The Secret key is given as:

\[
S_k = P_k \cdot T_{i,j}
\]

Key Generation (\( M_{sk}, S_k \): Assume a user \( U \) with an attribute set \( S_U \). This algorithm lets the users to encrypt and decrypt the messages under bounded access tree \( T \). Each user selects random exponential \( e^{a} \) for children node \( x \) in universal access tree \( T \). From top-to-bottom wise, the user’s nodes are selected. Exponential function is defined for each node \( x \), as set \( e^{a}(0) = e^{\text{path}(k)(\text{index}(x))} \). Once the exponential function is
decided, the master secret key to the user is given as:

$$M_{sk} = g^{\frac{j + x}{2 \cdot x}} \quad \text{where} \quad j \epsilon M_{sk}, x \epsilon \varphi_T$$

The above $M_{sk}$ is then used as decryption key $D$.

Encryption ($M, P_k, T'$): With the help of decryption key, $D$, the ciphertext is generated. $\varphi$ defines the map between nodes in access tree $T$. Then, non-lead node $x$ in $T$, selects random from $\alpha_x$ and then empty child node created as map$(x)$. Let $f(j, x)$ be the Boolean function and publish the ciphertext $E$ as follows:

$$\{T, E\} = M, M_{sk}, \{E_{j, x} = T_i^{\frac{j}{\lambda_{MON}(x)}}\} j \epsilon U, x \epsilon \varphi_T; f(j, x) = 1$$

Decryption ($E, D$): It operates by taking inputs such as ciphertext $E$, private key $M_{sk}$, and node $x$ in access tree $T$. Let $j$ be $\text{attr}(x)$ and $w$ be the parent $x$. Then, the decryption process is follows:

$$\text{Decrypt}(E, M_{sk}, x) = \{e(D_{j, \text{map}(w)}, E_{j, w}) = e(g^{\frac{j + \text{map}(w)}}{2 \cdot \text{map}(w)}) + j$$

It invokes on root node and divides out encryption process and recovers $M$.

**Fig.1 Proposed workflow**

### 4. EXPERIMENTAL RESULTS AND ANALYSIS

This section presents the experimental analysis of the proposed framework. The performance of the proposed framework is analyzed in terms of computational time taken during key generation time, encryption time and decryption time and its compared to previous schemes, updatable CP-ABE [ ].

**Key Generation Time:** It is the time taken for generating the master secret keys to the user. The number of exponentiation required in setup algorithm is assumed as $m + 3$, since there are no exponentiation operations. The fig.2 illustrates the time taken for generating the keys based on number of attributes. It is vital to analysis the key generation time for given number of attributes. As the number of attributes increases, the time stability for setup process defines the scalability of the system.

**Fig.2. Key generation time**

- **ii) Encryption time:** It is defined as the time taken for generating the ciphertext based on validating the secret keys. Fig. 3 presents the encryption time taken to generate the ciphertext for variant users. At user’s side, the encryption process holds some constraints, so as to eliminate the overheads. Hence, encryption time analysis is a significant performance analysis. Depends on attributes, the encryption time is computed and compared with updatable CP-ABE.

**Fig.3. Encryption time analysis**

- **iii) Decryption time:** It is the time taken to decrypt the ciphertext under stipulated period of time by satisfying the concealed policy constraints. The fig.4 illustrates the decryption time analysis. Exponential form of setup process incurs higher computational costs for even small number of attributes. In this work, it is formed without any pairing operation. Time taken for exponential process is lesser, even attributes increases.ss A concealed policy constraint is also simpler to preserve and index the data which makes the decryption process easier than the previous scheme, which is shown in fig.4.

**Fig.4. Decryption time analysis**

### 5. CONCLUSION

In this paper, an investigation about the issue of restricted bounded access tree structure has been analyzed using proposed, 'exponential cipher based concealed policy in CP-ABE'. This system differs from previous schemes, in terms of constructing the setup process. The deficiency in trusted authority has been resolved by refining the concealed policy which makes the system more flexible. Depends on given attributes, the exponential ciphers are generated which
defines a high-end security solutions. Through experimental evaluation and analysis, computational time for key generation, encryption and decryption have been explored under predefined set of attributes. Stability of time analysis based on attributes determines the effectiveness of the exponential cipher, in case of any security attacks. The system creates a complexity when any attackers try to decrypt the exponential cipher. The proposed scheme is compared with updatable CP-ABE and proved the system efficiency by nullifying the

REFERENCES


