An Efficient Approach for Data Control Mechanism with Time Factors in Public Cloud

Mrs. A.V.L. Prasuna[1], Mr. Sandeep Kodavati[2]
[1]Associate Professor, [2]M.Tech Scholar,
Dept of IT, MGIT, Hyderabad, India

Abstract - The new worldview of outsourcing information to the cloud is a twofold edged sword. From one viewpoint, it liberates data owners from the specialized administration, and is simpler for data owners to impart their information to proposed clients. Then again, it postures new difficulties on privacy and security insurance. To secure information secrecy against the legitimate yet inquisitive cloud service provider, various works have been proposed to help fine-grained information get to control. In any case, till now, no plans can bolster both fine-grained get to control and time-delicate information distributing. In this paper, by implanting planned discharge encryption into CP-ABE (Ciphertext-Policy Attribute-based Encryption), we propose An Efficient Approach for Data Control Mechanism with Time Factors in Public Cloud. In light of the proposed scheme, we additionally propose an effective way to design access policies faced with diverse requirements with time-delicate information. Broad security and execution investigation demonstrates that our proposed system is exceedingly proficient and fulfils the security prerequisites for time- delicate information in public cloud.

Keywords - storage, time sensitiveinformation, security, data access control.

I. INTRODUCTION

Cloud storage benefit has noteworthy points of interest on both advantageous information sharing and cost diminishment. Along these lines, an ever increasing number of endeavours and people outsource their information to the cloud to be profited from this administration. Be that as it may, this new worldview of information stockpiling postures new difficulties on information privacy conservation. as cloud benefit isolates the information from the cloud benefit customer (people or substances), denying their immediate control over these information, the data owner can't believe the cloud server to lead secure information get to control. Consequently, the safe access control issue has turned into a testing issue in broad daylight distributed storage ease of use.

Cipher text-policy attribute-based encryption (CP-ABE) is a helpful cryptographic technique for information get to control in distributed storage. All these CP-ABE based plans empower data owners to acknowledge fine-grained and adaptable access control without anyone else information. Be that as it may, CP-ABE decides clients' entrance benefit construct just in light of their innate qualities with no other basic variables, for example, the time factor. In actuality, the time factor as a rule assumes a vital part in managing time-delicate information.

Cloud storage benefit has noteworthy points of interest on both advantageous information sharing and cost diminishment. Along these lines, an ever increasing number of endeavors and people outsource their information to the cloud to be profited from this administration. Be that as it may, this new worldview of information stockpiling postures new difficulties on information privacy conservation. As cloud benefit isolates the information from the cloud benefit customer (people or substances), denying their immediate control over these information, the data owner can't believe the cloud server to lead secure information get to control. Consequently, the safe access control issue has turned into a testing issue in broad daylight distributed storage.

Cipher text-policy attribute-based encryption (CP-ABE) is a helpful cryptographic technique for information get to control in distributed storage. All these CP-ABE based plans empower data owners to acknowledge fine-grained and adaptable access control without anyone else information. Be that as it may, CP-ABE decides clients' entrance benefit construct just in light of their innate qualities with no other basic variables, for example, the time factor. In actuality, the time factor as a rule assumes a vital part in managing time-delicate information (e.g. to distribute a most recent electronic magazine, or to uncover an organization's future strategy for success). In these situations, both the component of access benefit coordinated discharging and fine-grained get to control ought to be as one considered. Give us a chance to take the undertaking information presentation for example: An organization more often than not readies some vital records for various planned clients, and these clients can pick up their entrance benefit at various time focuses. For instance, the future arrangement of this organization may contain some business privileged insights. Subsequently at an early time, the entrance benefit can be discharged to the CEO as it were. At that point the directors of some pertinent offices could get to benefit at a later time point, when they assume liability for the arrangement execution. Finally, different representatives in some particular bureaus of the organization can get to the information to assess the culmination of this endeavor design. While transferring time-delicate information to the cloud, the data owner needs extraordinary clients to get to the substance after various time focuses. To the outsourced information stockpiling, CP-ABE can describe distinctive clients and give fine-grained get to control. Nonetheless, to

our best learning, these plans can't bolster steady access benefit discharging.

To understand the capacity of coordinated discharging, it is important to present a compelling plan, which won't discharge the information get to benefit to proposed clients until achieving pre-characterized time focuses. An unimportant arrangement is to let data owners physically discharge the time-touchy information: The proprietor transfers the encoded information under various approaches at each discharging time with the end goal that the expected clients can't get to the information until the point when the relating time arrives. In any case, this arrangement powers the proprietor to over and again transfer the diverse encryption adaptations of similar information, which puts superfluous and substantial weight on the data owner.

From the point of view of cryptography, the capacity of planned access benefit discharging can be accomplished by Timed-Release Encryption (TRE). In a TRE-based framework, a trust time operator, instead of data owner, can consistently discharge the entrance benefit at a particular time. A few plans have been proposed to incorporate TRE into remote information get to control. In any case, these plans either need fine-grained get to control or leave a deplorable weight.

How to accomplish the limit of both timed-release and access control in cloud storage?

A direct however native strategy is to deal with the time factor as an attribute. However, agonizing number of time-related keys should be issued to every client at each pre-defined time point, which presents heavy overhead on both computation and communication.

In this paper, we propose a novel technique for user to control data access using combination of attribute and time factors in public cloud for time-delicate information out in the public cloud. Our plan has two critical abilities: 1) It acquires the property of fine granularity from CP-ABE; 2) By presenting the trapdoor system, it additionally holds the component of coordinated discharge from TRE. The presented trapdoor system is just identified with the time factor, and just a single comparing mystery should be distributed while uncovering the related trap-entryways. This makes our plan exceptionally productive, which just realizes minimal overhead to the first CP-ABE based plan. We should deliver how to outline a proficient access structure for subjective access benefit development with both time and characteristic components, particularly when an entrance strategy inserts numerous entrance benefit discharging time focuses. As an expansion of the past meeting rendition, we give the potential sub-arrangements for time-delicate information, and after that present an effective and reasonable technique to build significant access structures.

The principle commitments of this paper can be outlined as takes after:

1) By incorporating TRE and CP-ABE openly distributed storage, we propose a proficient plan to acknowledge secure fine-grained get to control for time-delicate information. In the proposed conspire, the data owner cans self-governingly des-ignate planned clients and their significant access benefit discharging time focuses. Other than understanding the capacity, it is demonstrated that the irrelevant weight is upon proprietors, clients and the confided in CA.

2) We present how to configuration get to structure for any potential planned discharge get to arrangement, particularly implanting various discharging time focuses for various proposed clients. To the best of our insight, we are the first to consider the way to deal with configuration structures for general time-touchy access prerequisites.

3) Furthermore, a thorough security evidence is given to approve that the proposed conspire is secure and viable.

Whatever remains of this paper is composed as takes after. We initially survey some current work that is identified with information get to control for time-delicate information in Section II. In Section III, we introduce the framework engineering and express the securities demonstrate. Segment IV depicts fundamental strategies. In Section V, we give point by point calculation of our proposed system, and examine the plan regarding its security and execution in Section VI. At last, we close this paper in Section VII.

II. RELATED WORK

In view of different cryptographic primitives, there have been various chips away at secure information partaking in distributed storage. Among these plans, some went for securing the trustworthiness of the mutual information and some went for ensuring the privacy and access control of the information. In the region of information get to control, property based encryption (ABE) is used as an essential cryptographic technique.

These ABE-based access control plans, all in all, can be separated into two primary classifications: Key-Policy ABE (KP-ABE) based plans and Ciphertext-Policy ABE (CP-ABE) based plans. The last one is more reasonable for accomplishing adaptable and fine-grained get to control for the general population cloud, in which each record is marked with an entrance structure, and every client owes a security key inserted with an arrangement of properties.

Be that as it may, the current ABE based plans don't bolster the situation where the entrance benefit of one record is required to be separately discharged to various arrangements of clients after various time focuses, yet needs just a single time of the cipher text transfer. A paltry arrangement is to let the data owner him/herself recover the document, re-scramble it under the new strategy, and transfer it again when the discharging time arrives. Nonetheless, such arrangement realizes substantial weight of both correspondence and calculation overhead on the data owner. The previous authors have proposed arrangement refresh techniques for KP-ABE based and CP-ABE based plans individually. If the data owner needs to discharge the entrance benefit to new arrangements of clients, he/she doesn't have to re-encode and transfer the entire record. The

data owner produces and sends an approach refresh key to the cloud, and the cloud can re-encode the put away document. With the change of access strategy, new arrangements of clients can get to the record. In any case, Yang's plan have recently talked about how to refresh the entrance structure, however not implanted the time factor into the entrance structure, which requires that the data owner must be online while executing strategy refreshing. In this manner, it is frantically expected to devise an effective plan, in which the data owner can assign the greater part of the document's future access arrangements when it is first encrypted.

Towards this test, Timed-Release Encryption (TRE) turns into a promising primitive, in which, a trusted time specialist, rather than data owners, consistently executes the coordinated discharge work. Such idea has been generally integrated to numerous situations. TRE to be coordinated to the accessible encryption conspire, in which the expected client is obliged to sit tight for a specific time to look through the outsourced information. The blend of TRE and intermediary encryption were proposed in cloud condition. TRE additionally accomplishes a restrictive absent exchange plan with the end goal that the entrance design is uncovered after a particular time.

In the situation of information get to control for open distributed storage, a few plans that receive the essential thought of TRE have been proposed as an intermediary encryption conspire for information sharing, where the information get to privilege can be precisely appropriated to planned clients who possess a specific property set amid a particular era. The proposed plan can well safeguard information privacy. Nonetheless, it can't fulfill the necessity that clients are compelled to get to information after specific assigned time.

Some looks into have likewise endeavored to consolidate the systems of TRE and CP-ABE, to give an adaptable and fine-grained get to control for time-delicate information. a fleeting access control framework for distributed storage, in which the cloud server deals with the time as a widespread clock benefit. Such development can't avoid the conspiracy between cloud server and clients. Diverse creators proposed a period space get to control framework, in which get to control takes both client's trait set and the entrance time into thought. Not quite the same as past this work accomplishes information get to benefit consequently discharging for clients without data owner's online investment. Be that as it may, it presents substantial additional overhead: The specialist needs to produce refresh keys for every potential ascribe each opportunity to actualize the time-related capacity, and the computational multifaceted nature increments with the measure of included properties. A more savvy plot is expected to acknowledge fine-grained get to control for time-delicate information in cloud storage.

III. SYSTEM AND SECURITY MODEL

A. System Model - Similar to most CP-ABE based schemes; the system in this paper consists of the following entities: a central authority (*CA*), several data owners (Owner), many data consumers (User), and a cloud service provider (Cloud).

The central authority (CA) is dependable to oversee the security insurance of the entire framework: It distributes framework parameters and conveys security keys to each client. Likewise, it goes about as a period specialist to keep up the Planned discharging capacity.

The Data Owner (Owner) chooses the entrance strategy based on a particular quality set and at least one discharging time focuses for each record, and afterward encodes the document under the chosen arrangement before transferring it.

The Data consumer (User) is allocated a security key from CA. He/she can question any cipher text put away in the cloud, yet can decode it just if both of the following requirements are fulfilled: 1) His/her quality set fulfills the entrance approach; 2) The present access time is later than the particular discharging time.

Cloud service provider (Cloud) incorporates the executive of the cloud and cloud servers. The cloud embraces the capacity undertaking for different elements, and executes get to benefit discharging calculation under the control of CA. As portrayed in Fig. 1, the cipher texts are transmitted from proprietors to the cloud, and clients can question any cipher texts. CA controls the framework with the accompanying two operations: 1) It issues security keys to every client, as per client's quality set; 2) At each time point, it distributes a Time token (TK), Which is utilized to discharge get to benefit of information to clients.

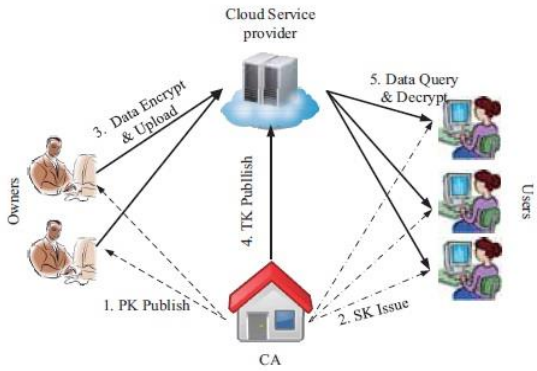

Figure - 1

B. Security Assumption - In our entrance control framework, the cloud is thought to be fair however inquisitive, which is like that expected in most of the related literary works on secure distributed storage. From one viewpoint, it offers dependable capacity benefit and effectively executes each calculation mission for different substances; On the other hand, it might attempt to increase unapproved data for its possess benefits.

Past the cloud, the entire framework comprises of one CA, A few proprietors and clients, in which CA is thought to be fully trusted, while clients could be noxious. CA is mindful for key dispersion and time token distributing. A malevolent client will endeavour to unscramble the ciphertexts to

acquire unapproved information by any conceivable means, incorporating intriguing with other malicious clients.

The proposed system can understand a fine-grained and timed releasing get to control framework: Only one client with a fulfilled quality set can get to the information after the particular time. The proposed systems are characterized to be bargained if both of the accompanying two sorts of clients can effectively decode the cipher text:

1) A client whose characteristic set does not fulfill the get to arrangement of a comparing cipher text;
2) A client who tries to get to the information before the predetermined discharging time, regardless of whether he/she has fulfilling characteristic set.

IV. TECHNICAL PRELIMINARIES

A. Bilinear Pairings and Complexity Assumption - Let G1 and G2 be two multiplicative cyclic groups of prime order p.

Let e : G1 × G1 → G2 be a bilinear map with the following properties:

1) Computability. There is an efficient algorithm to compute
   e(u, v) ∈ G2, for any u, v ∈ G1.
2) Bilinearity. For all u, v ∈ G1 and a, b ∈ Zp, we have
   e(ua, vb) = e(u, v)ab.

3) Non-degeneracy. If g is a generator of G1, then e(g, g) is also a generator of G2.

Definition 1: (Decisional BDH Assumption, DBDH). The DBDH assumption is that no polynomial-time adversary is able to distinguish the tuple (ga, gb, gc, e(g, g)abc) from another tuple (ga, gb, gc, e(g, g)z), if the adversary has no knowledge of the random elements a, b, c, z ∈ Z∗p.

B. Cipher text-Policy Attribute-based Encryption - CP-ABE is a cryptography model for one-to-numerous secure correspondence. In a CP-ABE based plan, other than the capacity stage, the framework comprises of three essential gatherings: The authority, the owner and the user.

The authority is acquainted with distribute framework parameters and issue secret keys for the clients. The owner shares documents to the planned clients by assigning an entrance approach and encoding the record under the approach. In CP-ABE based approach, the entrance strategy is communicated as a tree over a set of attributes and logic gates, which will be represented in detail later. Every client acquires his/her secret key from the authority in view of his/her own characteristics.

The usefulness and security model of CP-ABE accept that the capacity stage (e.g., cloud server) does not lead the entrance control administration. This sort of plans permit the client to inquiry any Cipher text, however he/she can unscramble the cipher text if and just if his/her characteristic set fulfils the get to approach of the record. A CP-ABE plot comprises of the following four calculations:

Setup: It takes a security parameter λ and the attribute universe description *U* as the input, and outputs a master key *MK*, and a public parameter *PK*.

Key Generation: It takes the master key *MK* and a set of attributes as the input, and outputs the security key *SK* associated with the input attribute set.

Encryption: It takes the public parameter *PK*, a message *M*, and an access policy *T* over some attributes as the input. It outputs the cipher text *CT*.

Decryption: It takes the security key *SK*, and the Cipher text *CT* as the input, and outputs either a message *M* or the distinguished symbol $\perp$.

C. Timed-Release Encryption - The idea of time release encryption is for situations that somebody needs to safely make an impression on another in what's to come in future. In detail, the owner encrypts his/her message for the reason that planned clients can decode it after an assigned time. From the security angle, TRE fulfills that:

1) Except the planned clients, nobody can get any data of the message;2) Even the proposed client can't get the plaintext of the message before the assigned discharging time.

With a specific end goal to bolster an exact planned discharge instrument, a confided in time agent is required to deal with the clock of the system. At each time point T, the agent discharges a time token TKT, which is a vital idea in TRE.

While encrypting the message, the ciphertext is created with people in public key of the planned client and the assigned discharging time T. The ciphertext holds the component that as it were with the comparing client's secret key and time token TKT , can a user correctly get the plaintext of the message; otherwise, if without either of the two components, the user cannot successfully conduct the decryption.

V. MAIN CONSTRUCTION OF OUR SCHEME

We firstly give an overview of our proposed system, mainly discussing how to achieve timed-release function in this paper.

Then, we introduce the concepts of access policy, time trapdoor and token. Lastly, we describe our proposed system in details. Table I describes the basic notations in this paper.

Table I

Notation	Description
MK	Master secret key of CA
PK	Public parameter of the system
M	Plaintext of the data
T	Access policy over attributes and time
CT	Cipher text of the data
Sj	Attribute set of user Uj
SKj	Attribute-associated security key of user Uj
TSx	Time trapdoor upon node x, in unexposed status
TS'x	Time trapdoor upon x, in exposed status
TKt	Time token of time t
FT	Unified format of time
H1	Hash function that maps elements in {0, 1}∗ to element in G∗1
H2	Hash function that maps elements in G∗2 to elements in Z∗p

A. Overview - In order to build a scalable and fine-grained access control system for outsourced time-sensitive data, we combine two advanced cryptographic techniques, namely CP-ABE and TRE. The former one is to provide an expressive access control primitive with determined attribute sets; and the latter one is to realize timed-release function.

The general idea of our unique mechanism is to realize access structures in a new form. As shown in Fig. 3, apart from attributes and logic gates defined in existing CP-ABE, the access structure in our scheme contains one or more time trapdoors (*TS*), each of which represents a time point. The trapdoor is implemented for the timed release function in CP-ABE algorithm. It can be placed upon any node in the structure, arbitrarily defining access privilege releasing time for different users. The accessing time, together with user's attribute set, determines whether the user satisfies the policy.

For every shared file, the data owner him/herself determines the access policy to encrypt the file. Especially, the time trapdoors in the policy are generated according to a time point $t \in$ FT . FT is system's unified time format, such as "dd/mm/yyyy". The time format designates the granularity of timed-release function, e.g., monthly, daily, or hourly. Such mechanism removes the complicated interactions between CA and data owners.

In the access policy, a node attached with a time trapdoor is said to be satisfied if it holds the following features: 1) Just like CP-ABE, if it is a leaf node, the relevant attribute is among the attribute set; otherwise, the number of its satisfied child nodes exceeds a threshold (will be discussed in detail later); 2) The current access time is later than the relevant releasing time point of the time trapdoor. From the cryptographic perspective, such idea is realized since *CA* publishes time token *TKT* in every time point, just like the time agent does in TRE. Our scheme works if the following feature holds: A user can decrypt a file if and only if his/her attribute set and the obtained time tokens satisfy the access policy. For the performance consideration, in our scheme, time related decryption can be outsourced to the cloud without losing confidentiality. Moreover, in order to ensure an approximate time consistency, we could introduce a less tight time synchronization mechanism. For example, a third-party Internet Time Server can be introduced, or owners and users all synchronize with CA, who opens a time synchronization interface for the public.

B. Access Policy and Time-Related Components

1) Access Policy Structure - The access policy is over some attributes and one or more releasing time points. Fig. 2 shows an example of the policy structure.

A structure *T* consists of a policy tree of several nodes, and some time trapdoors *TS*. A leaf node represents a certain attribute (In Fig. 2, *A*0, ... , *A*3 are the relevant attributes), and each non-leaf node represents a threshold gate ("AND", "OR", or others). Each non-leaf node *x* has two logic values *nx* and *kx*, where *nx* is the number of its child node, and *kx* is the threshold. Particularly, $kx = 1$ if *x* is an *OR* gate, or $kx = nx$ if *x* is an *AND* gate. In a structure *T* , the number of included time trapdoors can be zero, one, or more than one. Each trapdoor *TSx* is appended to a node *x*. From the perspective of algorithm, *TS* can be appended to arbitrary node of the structure (*leaf*, *nonleaf*, or even *root*). For instance, in Fig. 2, *TS*1 is appended to a leaf node in order to restrict the attribute *A*1, while *TS*2 is upon a non-leaf node to restrict a sub-policy "*A*2 ∧ *A*3".

2) Time Trapdoors and Time Tokens - Time trapdoor (*TS*) can be embedded in an access structure, such that the corresponding user's access permission is restricted by the status of *TS*. In this paper, we define two statuses, namely *exposed* or *unexposed*, for the time trapdoor.

Unexposed: A trapdoor (*TS*) is *unexposed* if the intended users cannot access the corresponding secret through the trapdoor with their security keys.

Exposed: A trapdoor is *exposed* if the intended users can get the corresponding secret through this trapdoor. An *exposed* trapdoor is denoted as *TS'*.

The status of a trapdoor can be transferred from "Unexposed" to "Exposed" with a relevant time token (*TKt*). After *TKt* is published at time *t*, anyone, including the cloud and any users, can transfer the status of corresponding time trapdoors (In this paper, the cloud server performs the operation of status transferring, which will not bring about user's overhead or introduce other undesired factors).

In our proposed system, a trapdoor *TS* is generated by a data owner when encrypting his/her data, and a time token *TK* is generated and published by *CA*. The cloud server can transfer one particular trapdoor's status from *unexposed* to *expose* after obtaining the corresponding *TKt*. Taking Fig. 2 as an example: The trapdoor *TS*1 is related to a time point *t*1, and *TS*2 is related to *t*2. Users that satisfy "*A*0 ∧*A*2 ∧*A*3" (such as *U*1) cannot get access privilege until the token *TKt*1 is published; And users satisfying "*A*0 ∧ *A*1" (such as *U*2) should wait for *CA* to publish *TKt*2 . Note that, any time *ti* in this paper represents a certain time point rather than a length of time interval. In the remaining of this paper, if $ti < tj$, it means that *ti* is an earlier time point than *tj*.

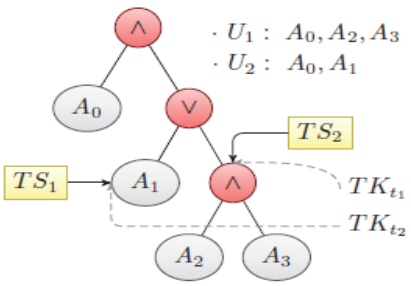

Figure - 2 Example of Access Structure

C. Construction - Our proposed system consists of six procedures: setup, key generation, encryption, token generation, trapdoor exposure and decryption. Fig. 3 depicts

a brief description of our scheme (setup and key generation are not included in the figure).

1) Setup - *CA* generates $I = [p, G1, G2, g, e, H1, H2, FT]$, where $e : G1 \times G1 \rightarrow G2$ is a bilinear map, G1 and G2 are cyclic multiplicative groups of a prime order p, g is a generator of G1, $H1 : \{0, 1\}* \rightarrow G*1$, $H2 : G*2 \rightarrow Zp*$. *FT* is the time format.

CA randomly chooses $\alpha, \beta, \gamma \in Zp*$. The public parameter is published as:

$PK = (I, h = g^{\beta}, f = g^{\gamma}, e(g, g)^{\alpha})$, and the master key *MK* is $(\beta, \gamma, g^{\alpha})$, which implicitly exists in the system, and doesn't need to be obtained by any other entity. (Note that f and γ are used for timed-release function.)

2) Key Generation - For each user *Uj* with attribute set *Sj* ,*CA* firstly chooses a random $uj \in Zp*$ as a unique identity for the user. Each attribute $Atti \in Sj$ is assigned a random ri. Then, *CA* computes the user's security key as:

$SKj = \{D = g^{(\alpha+uj)/\beta}, \forall Atti \in Sj : Di = g^{uj} \cdot H1(Atti)^{-ri}, D\_i = g^{ri}\}$.

At the end of this procedure, the security key *SKj* is sent to *Uj* in a secure tunnel.

3) Encryption - The data owner uses a symmetric cryptography to encrypt the data *M* with a random chosen key$K \in G2$. In this procedure, each node x in the predefined access structure T will associate with three secret parameters, denoted as $s0x$, $s1x$ and $s\tau x$. Here, $s0x$ is shared with its parent node,$s1x$ is shared with its child node (or dealt with the relevant attribute if x is a leaf node), and $s\tau x$ is a time-related parameter. Specifically, if x is the root R, $s0R$ is the base secret of T . The parameter assigning is in a top-down manner, starting from the root R as follows:

f x is R, the owner randomly chooses a random parameters$0R \in Zp$. For each node x with $s0x$, the parameters $s1x$ and $s\tau x$ are chosen as:

$$\begin{cases} s_x^{\tau} \in \mathbb{Z}_p^*, \ s_x^{\tau} \cdot s_x^1 = s_x^0 \\ s_x^{\tau} = 1, \ s_x^1 = s_x^0 \end{cases} \quad \begin{array}{l} x \text{ is linked to a time trapdoor} \\ \text{otherwise} \end{array}$$

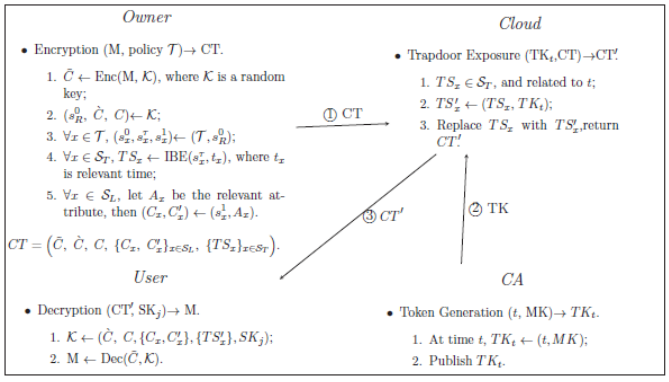

Figure – 3 Procedure description

4) Token Generation - At each time point t ∈ FT , CA generates and publicly publishes a time token TKt as follows: TKt = H1(t)γ .

5) Trapdoor Exposure - When arriving at the releasing time point t related to *TSx*, the cloud can obtain a corresponding token *TKt*, which is published by *CA*. Then, the cloud server implements this procedure to expose the trapdoor. When the cloud gets *TKt*, it queries all trapdoors associated with t in all access structures associated with the stored files on it. For each trapdoor $TS'x = (Ax, Bx)$, the cloud computes the *exposed* trapdoors as: $TS'x = Bx - H2(e(TKt, Ax))$. If the procedure is correctly implemented, we can get $TS'x = s\tau x$. The cloud replaces *TSx* with *TS'x* i each relevant *CT*, in which the trapdoor can be removed, and the access privilege is transferred to be determined only by the attribute set.

6) Decryption - After querying *CT* from the cloud, a user *Uj* (with the attribute set *Sj*) conducts this procedure with the security key *SKj*. The decryption procedure is performed in a bottom-up manner.

VI. SECURITY AND PERFORMANCE ANALYSIS

A. Security Analysis - We analyze the security properties on some critical aspects as follows:

1) Fine-Grained and Timed-Release Access Control: Our proposed system provides data owners with the capability to define access policies according to flexible association of attributes and releasing times. With the access policy embedded in the cipher text, a user can decrypt the cipher text to access the data, only if his/her attribute set satisfies the policy, and the access time is later than the predefined releasing time.

2) Security against Collusion Attack: Each user's attribute set-associated security key *SKj* is blinded based on a secure random number $uj \in Z*p$. This mechanism is implemented to resist the collusion attack: The adversary cannot combine different security keys (*SK*) to forge a new security key associated with a different attribute combination which comes from multiple attribute sets belong to different users. Therefore, the collusion will not bring more privileges to the adversary.

B. Performance Analysis - In order to give an intuitive evaluation of the performance of proposed system, we make a comparison with other related schemes, such as **LoTAC**, and an approach based on CP-ABE, where time is handled as attribute (denoted as **TasA**). Since the performance differences among these three schemes are mainly on communication and computation cost of *CA* and the data owner, we analyze these two aspects as follows:

1) CA's Cost for Timed-Release Function: Fig. 4 and Fig. 5 show the overhead evaluation of trust entities (including *CA*), with increasing number of users and released data respectively. A time token *TKt* is a universal parameter among all users for one time point t. *CA*, therefore, only needs to calculate and publish one token at each time. On the contrary, if time is handled as an attribute (as in TasA), *CA* should distribute time-associated security key to each user at each time, meaning that the extra cost is linear to the number of users.

In LoTAC, although *CA* does not need to do anything for timed-release function, another trust entity, should

implement the timed-release decryption algorithm for each file at each release time. The overhead of this trust entity for this job is linear to the amount of relevant data, as shown in Fig. 5. The timed-release computation for every file can be outsourced to the *honest-but-curious* cloud, without leaking any unauthorized secret. Thus, our proposed system shows its superiority on *CA*'s cost reduction, when the access control system includes large amount of users and shared data.

2) Owner's Cost versus Number of Intended Users: When the owner uploads his/her file, his/her communication cost depends on the package size of the corresponding ciphertext. If we only consider the number of intended users, the cost of owner in LoTAC is $O(|U|)$, where $|U|$ is the number of intended users; while the cost in TAFC and TasA is $O(Natt)$, where *Natt* is the number of attributes in an access policy. In reality, when the number of intended users increases, *Natt* will increase much more slowly than $|U|$, in quite a high probability. With this assumption, Fig. 6 gives the overhead evaluation of data owner with increasing intended users, when encrypting one data file.

Because of fine granularity inherited from CPABE,it significantly reduce the communication complexity of data owner when the access privilege should be released to quite a number of users.

Based on the performance analysis on various aspects, we can conclude that the proposed system tolerates the increasing number of users and shared data. Thus, can provide a lightweight, flexible, and fine-grained access control system for time-sensitive data in cloud storage.

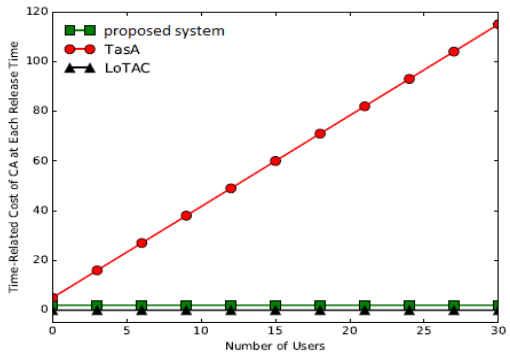

Figure - 4 Cost of CA Versus number of Users

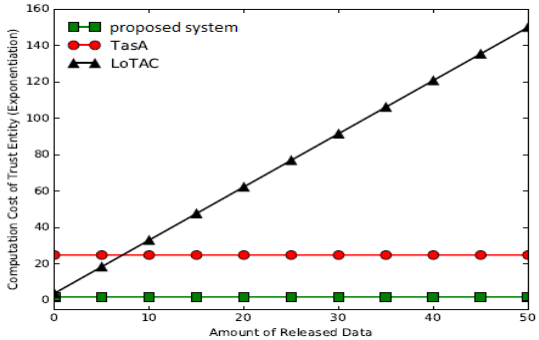

Figure - 5 Computation cost of trust entity versus amount of release data

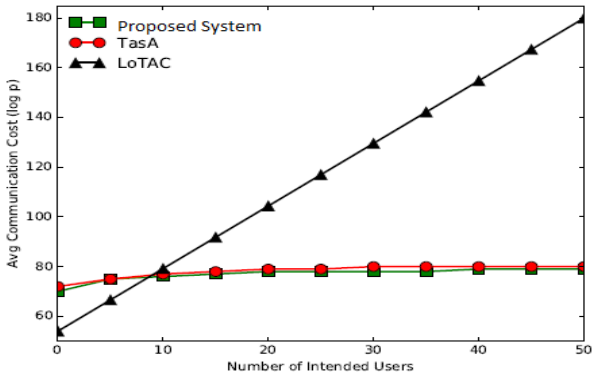

Figure - 6 Cost of owner versus number of intended users.

VII. CONCLUSION

This paper aims at fine-grained access control for time sensitive information in Cloud storage. One challenge is to simultaneously accomplish both flexible timed release and fine granularity with lightweight overhead, which was not observed in existing works. In this paper, we proposed a plan to accomplish this objective. Our plan consistently consolidates the concept of timed-release encryption to the architecture of Ciphertext policy attribute-based encryption. With a suit of proposed mechanisms, this plan gives data owners the capacity to adaptably release the access privileges to various clients at various time, according to a well-defined access policy over attributes and release time.

VIII. ACKNOWLEDGMENT

We sincerely thank the unknown arbitrators for their precious proposals that have prompted the present progress of the proposed system.

IX. REFERENCES

[1]. Z. Qin, H. Xiong, S. Wu, and J. Batamuliza, "A survey of proxy re-encryption for secure data sharing in cloud computing," IEEE Transactions on Services Computing, Avaliable online, 2016.

[2]. Z. Fu, K. Ren, J. Shu, X. Sun, and F. Huang, "Enabling personalized search over encrypted outsourced data with efficiency improvement," IEEE Transactions on Parallel and Distributed Systems, vol. 27, no. 9, pp. 2546–2559, 2015.

[3]. Z. Wan, J. Liu, and R. H. Deng, "HASBE: A hierarchical attribute-based solution for flexible and scalable access control in cloud computing," IEEE Transactions on Information Forensics and Security, vol. 7, no. 2, pp. 743 754, 2012.

[4]. Q. Liu, G. Wang, and J. Wu, "Time-based proxy re-encryption scheme for secure data sharing in a cloud environment," Information Sciences, vol. 258, no. 3, pp. 355–370, 2014.

[5]. F. Armknecht, J.-M. Bohli, G. O. Karame, and F. Y-oussef, "Transparent data deduplication in the cloud," in Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pp. 886–900, ACM, 2015.

[6]. . Masood, M. A. Shibli, Y. Ghazi, A. Kanwal, and A. Ali, "Cloud authorization: exploring techniques and approach towards effective access control framework," Frontiers of Computer Science, vol. 9, no. 2, pp. 297–321, 2015.