A Dynamic Time-Bound Access Control for Secure Hierarchical Content Sharing

Vanga Odelu [1]

[1]Indian Institute of Information Technology SriCity Chittoor

October 30, 2023

Abstract

With rapid growth of mobile users, protecting content from unauthorized users become a complex problem. The concept of temporal role-based access control reduces complexity of user management and restricts access to specified time-slots. But, content privacy is still questionable in case of system resources compromise unexpectedly. Therefore, cryptographic solution for time-bound hierarchical content management is an emerging problem. Most of the related schemes focused on individual user keys and/or revocation, but not on time-bound keys. Hence, these are not well suitable for subscription-based services like pay-TV and newspaper. In this paper, we propose a cryptographic time-bound access control with constant size time-bound keys. In our scheme, subscribed time-slots embed into individual user keys to avoid periodical broadcasting of temporal keys. We prove that our scheme is selectively secure under chosen-ciphertext attack. We then discuss cloud-based application to show the strategies of efficient revocation and reduce user computational overheads.

A Dynamic Time-Bound Access Control for Secure Hierarchical Content Sharing

Vanga Odelu, *Senior Member, IEEE*

Abstract—With rapid growth of mobile users, protecting content from unauthorized users become a complex problem. The concept of temporal role-based access control reduces complexity of user management and restricts access to specified time-slots. But, content privacy is still questionable in case of system resources compromise unexpectedly. Therefore, cryptographic solution for time-bound hierarchical content management is an emerging problem. Most of the related schemes focused on individual user keys and/or revocation, but not on time-bound keys. Hence, these are not well suitable for subscription-based services like pay-TV and newspaper. In this paper, we propose a cryptographic time-bound access control with constant size time-bound keys. In our scheme, subscribed time-slots embed into individual user keys to avoid periodical broadcasting of temporal keys. We prove that our scheme is selectively secure under chosen-ciphertext attack. We then discuss cloud-based application to show the strategies of efficient revocation and reduce user computational overheads.

Index Terms—Security, Hierarchy, Time-Bound Access Control, Key Management, Content Sharing.

1 Introduction

In subscription-based services such as pay-TV, newspapers and live video/audio broadcast, content sharing becomes complex problem due to rapid growth of mobile such as smartphones, set-top box, smart-TV, and laptops users in the recent days. In these services, data is periodically generated and shared with the set of authorized users in specified time-intervals [1]. A concept *temporal role-based access control* reduces the management of users by assigning roles at system level and restrict access in the specified time-slots [2], [3]. But, content privacy is still questionable in case of the system resources unexpectedly compromise to the adversary [4]. Hence, it is not sufficient for the practical applications, like subscription-based services, under the system resource compromise attacks.

In 1983, Akl-Taylor [5] first presented the seminal work on hierarchical key management, and later many variants are presented in the literature [6], [7], [8], [9]. In hierarchical access control, users are divided into disjoint groups, called security classes, based-on their access rights. The set of security classes forms a hierarchy (partially order set) such that the predecessor security class can access the content of it's successor security classes. A trusted administrator assign the keys to each security class, known as class access key and share it with the users of that class. It is clear that the user key is same as class access key and also group-oriented key, that is common to all users in that class. In addition, the key once assigned to set of users, it will never expire. Hence, the compromise of one user in the security class will affect the entire class. So, these approaches are not well suitable for subscription-based services. In order to address the limitation above approach, the concept of time-bound hierarchical key assignment were introduced in the literature [10], [11]. In this approach, the users are allowed to access class content for a specified time-slots only. However, the group-oriented user keys concept still remain in these approaches. Due to the group oriented user keys, it is difficult to handle the individual user revocation/tracing in the system.

The more flexible access control model *Attribute-Based Encryption* (ABE) [12], [13], [14] presented in the literature. However, the user/attribute revocation is little challenging in the ABE models [15]. With the motivation of the above limitations, Zhou et al. [15] was first introduced a concept of cryptographic role-based access control, known as Role-Based Encryption (RBE), for cloud-based content sharing. Zhou et al. [15] scheme offers constant size secret keys and ciphertext. In RBE, the cryptographic keys embed to roles in the role-hierarchy such that the user who holds required roles can access the encrypted content. In RBE, the content in the role is encrypted such that all users in the predecessor roles can access the content using their user keys. Later, Zhu et al. [4] presented a similar approach for role-key hierarchy. It provides individual user keys, and therefore, it can support for user revocation, undeniability and traceability. But, the master key size is linear to the number of roles. In addition, the ciphertext size is also linearly with the number of roles. In Zhu et al. [4] scheme, the revoked users list embeds in the encrypted content. Therefore, data owner must know in advance the revoked list of users. Thus, the revocation becomes bit complex with the increase in the number of revoked users [16]. In [16], a RBE Scheme proposed for securing outsourced cloud data in a multi-organization context. As a summary, most of the existing related schemes in the literature are more focused only on some of the features required for subscription-based services, such as individual user keys, time-bound access control, efficient user revocation and role/class revocation. This motivates us to design a time-bound access control scheme which offers most of the required features for the subscription based services with focused on less overhead for end users.

In this paper, we propose a novel time-bound access control

• *V. Odelu with the Department of Computer Science and Engineering, Indian Institute of Information Technology Sri City, Chittoor-517646, Andhra Pradesh, Indian.*

E-mail: odelu.vanga@gmail.com

Manuscript received xxxx; revised xxxx.

scheme with constant-size keys. In our scheme, user keys are embed with subscribed time-slots, so that it restrict the unauthorized access of content beyond the subscribed time-slots. That is, the user keys are not allowed to access the content encrypted in a time-interval if the user is not subscribed to that time-slot. In order to show the efficient key management and reduced user overheads, we discuss a cloud-based content management application. In this application, we discuss the strategies to handle efficient revocation of user keys, when required, from the system. A brief summary of review is presented in Table 1.

TABLE 1
Brief Summary of Review

Scheme	$F1$	$F2$	$F3$	$F4$	$F5$	$F6$
Odelu et al. [7]	No	No	No	No	Yes	No
Castiglione et al. [9]	No	No	No	No	Yes	No
Tzeng [10]	No	No	No	No	Yes	No
Bertino et al. [11]	No	No	No	No	Yes	No
Zhou et al. [15]	Yes	Yes	No	Yes	No	No
Zhu et al. [4]	Yes	Yes	No	Yes	No	No
Proposed scheme	Yes	Yes	Yes	Yes	Yes	Yes

$F1$: Whether class encryption key is different for different encryptions;
$F2$: Whether each subscribed user issued unique key;
$F3$: Whether user key is embed with subscribed time-slots;
$F4$: Whether scheme support for individual user revocation;
$F5$: Whether efficiently support for dynamic properties;
$F6$: Whether supports hierarchical time-bound encryption.

Contributions: Major contributions of the paper are listed below.

- The content is encrypted with public-key so that each user in the predecessor class can access the successor class content using his/her user key.
- Both user and class keys are generated with time-bound constraint. Therefore, user can access content encrypted in specified time-intervals, that is, only if the user key subscribed to that time-slot.
- We prove the proposed scheme is selectively secure under the chosen ciphertext attack.
- We present the cloud-based application and discuss the strategies to reduce computational overhead for end users as well as the efficient revocation.

Organization: Rest of the paper is organized as follows. In section 2, we discuss access control preliminaries. In Section 3, presented the time-bound access control encryption and selective security game. Then we propose our time-bound access control scheme presented in Section 4. In Section 5, we proved that the proposed scheme is selectively secure under the chosen-ciphertext attack. In Section 6, we present the cloud-based content management application and also future possible extension of the work. Finally in Section 7, we conclude the paper.

2 Access Control Preliminaries

In this section, we discuss the time-bound partial order relation and other required mathematical preliminaries.

2.1 Time-Bound Partial-Order Relation

The entity parameters involved in the access control are as follows:

- Set of n security classes $\mathbb{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots \mathcal{C}_n\}$.
- Sequence of time-intervals $\mathbb{T} : \mathcal{T}_1 < \mathcal{T}_2 < \ldots < \mathcal{T}_t \ldots$
- Set of users $\mathbb{U} = \{\mathcal{U}_1, \mathcal{U}_2, \ldots, \mathcal{U}_l, \ldots\}$.

The partially ordered relation, called hierarchy, is denoted by $\mathcal{H} = (\mathbb{C}, \geq)$, where $\geq$ is a binary relation. The set of successor security classes of class $\mathcal{C}_i$ is defined as $\mathbb{C}_i = \{\mathcal{C}_j | \mathcal{C}_i \geq \mathcal{C}_j, \text{ for } \mathcal{C}_j \in \mathbb{C}\}$, where $\mathcal{C}_i \geq \mathcal{C}_j$ means that $\mathcal{C}_i$ is predecessor of class $\mathcal{C}_j$ and $\mathcal{C}_j$ is successor of class $\mathcal{C}_i$. We denote the time-interval $[\mathcal{T}_{t-1}, \mathcal{T}_t]$ as $\mathcal{T}_t = [\mathcal{T}_{t-1}, \mathcal{T}_t]$ and set $\mathbb{C}_{it} = \mathbb{C}_i \cup \{\mathcal{T}_t\}$ is the set of successors of class $\mathcal{C}_i$ at time-interval $\mathcal{T}_t$. A set $\mathbb{U}_{il} = \mathbb{C}_i \cup \mathbb{T}_{il}$ is assigned for l-th user $\mathcal{U}_l$ who is subscribed to the security class $\mathcal{C}_i$ for a set of time-slots $\mathbb{T}_{il}$. Note that the maximum time-slots will be fixed to a number. If $\mathbb{C}_{jt} \subseteq \mathbb{U}_{il}$, then user $\mathcal{U}_l$ can access content encrypted for the class C_j in the time-interval $\mathcal{T}_t$. We update the hierarchy $\mathcal{H}$ to time-bound hierarchy, say $\mathcal{H}_t$, as $\mathcal{H}_t = (\mathbb{C}_t, \geq)$, where $\mathbb{C}_t = \{\mathcal{C}_{1t}, \mathcal{C}_{2t}, \ldots \mathcal{C}_{nt}\}$.

Consider an example hierarchy, say $\mathcal{H} = (\mathbb{C}, \geq)$, given in Figure 1 with set of five security classes, say $\mathbb{C} = \{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4, \mathcal{C}_5\}$, where $\mathbb{C}_1 = \{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4\}$; $\mathbb{C}_2 = \{\mathcal{C}_2, \mathcal{C}_4\}$; $\mathbb{C}_3 = \{\mathcal{C}_3, \mathcal{C}_4\}$; $\mathbb{C}_4 = \{\mathcal{C}_4\}$; and $\mathbb{C}_5 = \{\mathcal{C}_5, \mathcal{C}_3, \mathcal{C}_4\}$. Now, the time-bound hierarchy $\mathcal{H}_t = (\mathbb{C}_t, \geq)$ at time-interval $\mathcal{T}_t$ is defined as $\mathbb{C}_t = \{\mathcal{C}_{1t}, \mathcal{C}_{2t}, \mathcal{C}_{3t}, \mathcal{C}_{4t}, \mathcal{C}_{5t}\}$, where $\mathbb{C}_{1t} = \{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4, \mathcal{T}_t\}$; $\mathbb{C}_{2t} = \{\mathcal{C}_2, \mathcal{C}_4, \mathcal{T}_t\}$; $\mathbb{C}_{3t} = \{\mathcal{C}_3, \mathcal{C}_4, \mathcal{T}_t\}$; $\mathbb{C}_{4t} = \{\mathcal{C}_4, \mathcal{T}_t\}$; and $\mathbb{C}_{5t} = \{\mathcal{C}_5, \mathcal{C}_3, \mathcal{C}_4, \mathcal{T}_t\}$. We can also observe that the security class $\mathcal{C}_5$ act as that the private class to the class $\mathcal{C}_3$ like private permissions in the role-based access control system [2].

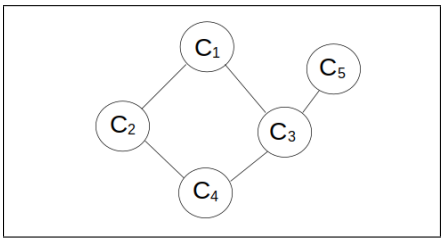


Fig. 1. Example hierarchy

2.2 Polynomial Functions

The following polynomials are used to define and control access in the hierarchy. Let $h : \{0,1\}^* \rightarrow \{0,1\}^\rho$ be a secure hash function.

$$f(x, \mathbb{U}_{il}) = \left(\prod_{x_c \in \mathbb{U}_{il}} (x + h(x_c)) \right)$$

$$f(x, \mathbb{C}_{it}) = \left(\prod_{x_c \in \mathbb{C}_{it}} (x + h(x_c)) \right)$$

$$F(x, \mathbb{U}_{il}, \mathbb{C}_{jt}) = \frac{f(x, \mathbb{U}_{il})}{f(x, \mathbb{C}_{jt})}$$

The access control is based on the fact that the function $F(x, \mathbb{U}_{il}, \mathbb{C}_{jt})$ is a polynomial function $\prod_{x_c \in (\mathbb{U}_{il} - \mathbb{C}_{jt})} (x + h(x_c))$ if and only if $\mathbb{C}_{jt} \subseteq \mathbb{U}_{il}$.

2.3 Bilinear Pairings

Let G_1, G_2 be two elliptic curve cyclic groups of large prime order q and G_3 be a cyclic multiplicative group of same prime order q. For any $P \in G_1$, $Q \in G_2$ and for all $a, b \in Z_q^*$, we define a map $e : G_1 \times G_2 \rightarrow G_3$ with the following properties, called bilinear map [4], [12]:

- *Bilinearity*: $e([a]P, [b]Q) = e(P, Q)^{ab}$.
- *Non-degeneracy*: $e(P, Q) \neq 1$ unless P or Q is 1.
- *Computability*: $e(P, Q)$ is efficiently computable.

Note that $[a]P$ denotes an elliptic curve scalar multiplication with scalar $a \in Z_q$. Let the points P and Q are generators of the cyclic groups G_1 and G_2, respectively. Then $g = e(P, Q)$ is a generator of target G_3. We call that tuple $G = (q, P, Q, G_1, G_2, G_3, e)$ is the bilinear pairing group.

3 Time-Bound Access Control Scheme

In section, we present time-bound access control scheme and a selective security game that is used to prove the security of proposed scheme under chosen-ciphertext attack.

3.1 Time-Bound Hierarchical Encryption

The time-bound hierarchical access control scheme (THACS) comprises of four algorithms such as $Setup$, Class Content Encryption (Enc), User Key Generation ($KGen$) and Class Content Decryption (Dec). These algorithms are defined as follows:

- **Setup** algorithm takes a security parameter ρ and set of n security classes $\mathbb{C} = \{\mathcal{C}_i | i = 1, 2, \ldots, n\}$ as input, and outputs a pair of master secret & public keys, say (MSK, MPK) and hierarchy $\mathcal{H} = (\mathbb{C}, \geq)$. Note that the general hierarchy $\mathcal{H} = (\mathbb{C}, \geq)$ will be updated periodically to $\mathcal{H}_t = (\mathbb{C}_t, \geq)$ in every time-interval $\mathcal{T}_t$ to control access according to the time-slots.
- **Class Content Encryption** algorithm $Enc(\mathbb{C}_{it}, MPK, \mathcal{H}_t)$ takes input a set $\mathbb{C}_{it}$ - the set of successors of $\mathcal{C}_i$ at time-interval $\mathcal{T}_t$, the master public key MPK and time-bound hierarchy $\mathcal{H}_t$. Then, it outputs the class ciphertext CT_{ir} for class $\mathcal{C}_i$ in the time-interval $\mathcal{T}_t$.
- **User Key Generation** algorithm $KGen(\mathbb{U}_{il}, MSK)$ takes input a set $\mathbb{U}_{il}$ defined as union of $\mathbb{C}_i$-successors of class C_i, and subscribed time-slots $\mathbb{T}_l$, that is, $\mathbb{U}_{il} = \mathbb{C}_i \cup \mathbb{T}_l$, and the master secret key MSK. Then, it outputs user key K_{il} for user $\mathcal{U}_l$.
- **Class Content Decryption** algorithm $Dec(\mathbb{U}_{il}, \mathbb{C}_{jt}, K_{il}, CT_{jr}, \mathcal{H}_t)$ takes input a class ciphertext CT_{jr} of class C_j generated with $\mathbb{C}_{jt}$ in the time-interval $\mathcal{T}_t$, user key K_{il} of $\mathbb{U}_{il}$, the master public key MPK and time-bound hierarchy $\mathcal{H}_t$. Then, the decryption algorithm outputs message M or $\perp$ (null).

The proposed THACS must satisfy that the decryption algorithm $Dec(\mathcal{U}_{il}, \mathbb{C}_{jt}, K_{il}, CT_{jt}, \mathcal{H}_t)$ always outputs the correct M using user secret key K_{il} corresponding to the set $\mathbb{U}_{il}$ only if the relation $\mathbb{C}_{jt} \subseteq \mathbb{U}_{il}$ holds.

3.2 Selective Security Game

Suppose $\mathcal{A}$ is an adversary try to decrypt the message encrypts for the predecessor classes. In this model, the adversary act on behalf of corrupted users to recover plaintext. The game between adversary $\mathcal{A}$ and challenger $\mathcal{B}$ is detailed below [12].

- **Initialization:** $\mathcal{A}$ outputs a security class $\mathcal{C}_i^*$ at time-interval $\mathcal{T}_t$, i.e., a set $\mathbb{C}_{it}^*$, and send to the challenger $\mathcal{B}$.
- **Setup:** The challenger $\mathcal{B}$ runs $Setup$ algorithm under the security parameter ρ. $\mathcal{B}$ generates master key pair (MPK, MSK) and hierarchy $\mathcal{H}_t = (\mathbb{C}_t, \geq)$. Then, $\mathcal{B}$ sends master key pair MPK and hierarchy $\mathcal{H}_t$ to the adversary $\mathcal{A}$.
- **Query:** The adversary $\mathcal{A}$ makes the following queries to the challenger $\mathcal{B}$.
  - User key K_{il} query on set $\mathbb{U}_{il}$.
  - Decryption query on $Enc(\mathbb{C}_{it}, MPK, \mathcal{H}_t)$.
- **Challenge:** The adversary $\mathcal{A}$ outputs two messages, say (M_0, M_1) and appoints a time-interval $\mathcal{T}_t$ and class C_i^* on which adversary wishes to challenge $\mathcal{B}$. We assume that the $\mathcal{A}$ is not queried on user key for the set $\mathbb{U}_{il}$ which satisfies the relation $\mathbb{C}_{it}^* \subseteq \mathbb{U}_{il}$. Next, $\mathcal{B}$ randomly picks a bit b from $\{0, 1\}$, and sends the challenge ciphertext CT_b^* to the adversary $\mathcal{A}$.
- **Query:** The adversary $\mathcal{A}$ can continue user key query on $\mathbb{U}_{il}$ that does not which satisfy the relation $\mathbb{C}_{it}^* \subseteq \mathbb{U}_{il}$ and no query on decryption.
- **Guess:** At the end, adversary $\mathcal{A}$ outputs a guess bit b' for b, and he wins the game if $b' = b$.

In the above game, the users collude and try to recover the palintext which is not authorized to access by them. The advantage of the t-time adversary $\mathcal{A}$ is then defined as follows:

$$Adv_{\epsilon,\mathcal{A}}^{ind-cca}(t) < \epsilon = |Pr[b' = b] - 1/2|$$

Definition 1. *The proposed access control scheme THACS is (t, q_d, q_s, ϵ)-selectively secure under chosen-ciphertext attack if the adversary advantage $Adv_{\epsilon,\mathcal{A}}^{ind-cca}(t) < \epsilon$, where any t-polynomial time adversary $\mathcal{A}$ who makes at most q_s user key queries and q_d decryption queries, with ϵ is a negligible function of the security parameter ρ.*

4 Proposed Access Control Scheme

In this section, we discuss in detail the various phases of the proposed time-bound access control encryption scheme.

4.1 System Setup

The admin initializes the system parameters as follows:

- Chooses bilinear group $G = (q, P, Q, G_1, G_2, G_3, e)$ and a cryptographic hash function $h : \{0,1\}^* \rightarrow \{0,1\}^\rho$, where ρ is a fixed integer.
- Randomly select $\alpha \in Z_q^*$ and sets $MSK = \{P, \alpha\}$ as master secret key.
- Construct hierarchy, say $\mathcal{H} = (\mathbb{C}, \geq)$ with n security classes and for each $\mathcal{C}_i$, the set of successors defined as $\mathbb{C}_i = \{\mathcal{C}_j | \mathcal{C}_i \geq \mathcal{C}_j\}$ for all $\mathcal{C}_j \in \mathbb{C}$.
- Next, outputs master public key $MPK = \{q, Q, G_1, G_2, G_3, e, e(P, Q), h, \mathcal{H}\}$.

4.2 General Hierarchy Construction

The construct of cryptographic hierarchy as follows.

- Compute $P_c = [\alpha^c]P$ and $Q_c = [\alpha^c]Q$, for $c = 1, 2, \ldots, n$, where n is the number of security classes in the hierarchy $\mathcal{H}$.
- For each $\mathcal{C}_i$ in $\mathcal{H}$, compute hierarchy control public keys

$$B_i = \Big[\frac{1}{f(\alpha, \mathbb{C}_i)}\Big]P$$

- Finally, declares the general hierarchy

$$\mathcal{H} = \{P_c, Q_c, B_i, \text{ where } c, i = 1, 2, \ldots, n\}$$

4.3 Time-Bound Hierarchy Construction

Construct the cryptographic time-bound hierarchy $\mathcal{H}_t$ as follows. Let $m \geq |\mathcal{T}_{il}|$ be, fixed-number, the maximum time-slots allowed for each user subscription. We then have $2 \leq |\mathbb{U}_{il}| = |\mathbb{C}_i \cup \mathbb{T}_{il}| \leq \mu = n + m$ and also denote the maximum cardinality of set of successors of any class in the hierarchy $\mathcal{H}_t$ is $n + 1$. That is, $2 \leq |\mathbb{C}_{it}| \leq n + 1$ for all $i = 1, 2, \ldots, n$.

- Computes $P_c = [\alpha^c]P$ and $Q_c = [\alpha^c]Q$ for $c = n + 1, n + 2, \ldots, \mu$
- For each $\mathcal{C}_i$, $i = 1, 2, \ldots n$, at time-interval $\mathcal{T}_t$, compute time-bound hierarchy control keys B_{it} as

$$B_{it} = \Big[\frac{1}{(\alpha + h(\mathcal{T}_t)}\Big]B_i = \Big[\frac{1}{f(\alpha, \mathcal{C}_{it})}\Big]P$$

- Then, declares the updated time-bound hierarchy control keys $\mathcal{H}_t = \{B_{it}, P_c, Q_c\}$, where $i = 1, 2, \ldots, n$ and $c = 1, 2, \ldots, \mu$.

Remark 1. *The parameters P_c and Q_c for $c = 1, 2, \ldots, \mu$ are fixed and no need to update for each time-interval unless the number of classes or maximum time-slots-allowed for subscription are increased.*

Remark 2. *The time-bound hierarchy control keys B_{it} for each class C_i needs to update for every time-interval $\mathcal{T}_t$. It requires only n number of group operations for every time-interval. If the time-interval is reasonably large, then it will not significantly affect the performance of system.*

Remark 3. *For new class addition/deletion, it requires the $O(\log n)$ computations to change the hierarchical structure to update predecessor security classes. In addition, it also requires to update the user keys of predecessor classes. Since the user key are temporal, periodical update of user keys are mandatory. Hence, it also will not affect the system performance if we choose appropriately the action of hierarchy update.*

4.4 Class Content Encryption

An encryption of content, say Ψ_{ir} with random key k_{ir} for class $\mathcal{C}_i$ at time-interval $\mathcal{T}_t$ is works as follows.

- Select $r_{it} \in Z_q^*$ and compute key $k_{ir} = e(P, Q)^{r_{it}}$ in G_3 and ciphertext $\Psi_{ir} = E(k_{ir}, \Phi_{ir})$, the symmetric encryption of message Φ_{ir} with key k_{ir}.
- Next, compute the corresponding public parameters as

$$B_{ir} = [r_{it}]B_{it} = \Big[\frac{r_{it}}{f(\alpha, \mathbb{C}_{it})}\Big]P$$
$$P_{cr} = [r_{it}]P_c = [r_{it}\alpha^c]P$$

  where $c = 1, 2, \ldots, (\mu - |\mathbb{C}_{it}| + 1)$.
- The class cipher $CT_{ir} = \{\Psi_{ir}, B_{ir}, P_{cr}\}$.

Remark 4. *If any content which make accessible in entire system life-time, the it will be encrypted using the general hierarchical control keys $\mathcal{H}$. On the other hand, to restrict the content access for specified time-intervals, then it will be encrypted using time-bound hierarchical control keys $\mathcal{H}_t$.*

4.5 User Key Generation

Assume that l-th user $\mathcal{U}_l$ wants to subscribe to the i-th class $\mathcal{C}_i$ for time-slots, say $\mathcal{T}_{il}$. Note that the set of successor classes with time-slots for user $\mathcal{U}_l$ is defined as $\mathbb{U}_{il} = \mathbb{C}_i \cup \mathbb{T}_{il}$. Then generates user subscription key as follows:

- Choose random number $s_{il} \in Z_q^*$.
- Computes the user key $K_{il} = \{K_{l1}, K_{l2}\}$, where

$$K_{l1} = \Big[\frac{(s_{il} - 1)}{\alpha}\Big]Q \quad \text{and} \quad K_{l2} = [s_{il} f(\alpha, \mathbb{U}_{il})]Q$$

4.6 Class Content Decryption

A user $\mathcal{U}_l$ subscribed to a security class $\mathcal{C}_i$ will derive a successor security class $\mathcal{C}_j$'s access key $k_{jr} = e(P, Q)^{r_{jt}}$ generated in the time-interval $\mathcal{T}_t$ if $\mathbb{C}_{jt} \subseteq \mathbb{U}_{il}$. Assume that $N = |\mathbb{U}_{il} - \mathbb{C}_{jt}| \leq \mu - 2$. The time-bound content decryption key k_{jr} of $\mathcal{C}_j$ generated at time-interval $\mathcal{T}_t$ can be derived as follows:

$$U = e\Big(P_{1r}, \sum_{c=1}^{N}[F_c]Q_{c-1}\Big) = e(P, Q)^{r_{jt}(F(\alpha) - F_0)}$$
$$V = e(B_{jr}, K_{l2}) = e(P, Q)^{r_{jt} s_{il} F(\alpha)}$$
$$W = e\Big(\sum_{c=1}^{N+1}[F_{c-1}]P_{cr}, K_{l1}\Big)$$
$$= e(P, Q)^{s_{il} r_{jt} F(\alpha) - r_{jt} F(\alpha)}$$
$$k_{jr} = e(P, Q)^{r_{jt}} = \left(\frac{V}{UW}\right)^{1/F_0}$$

where $F(x) = F(x, \mathbb{U}_{il}, \mathbb{C}_{jt}) = \sum_{c=0}^{N} F_c x^c$. Finally, decrypt the ciphertext as $\Phi_{jr} = D(k_{jr}, \Psi_{jr})$ using key k_{jr} of class $\mathcal{C}_j$, which is generated in the time-interval $\mathcal{T}_t$.

5 Security Analysis

In our proof we use the $(\eta_1, \eta_2, \epsilon)$-$aMSE$-$DDH$ the modified Augmented Multi-Sequence of Exponents Decision Diffie-Hellman problem from [17]. Consider the pairing group $G = \{q, P, Q, G_1, G_2, G_3, e\}$ and let $f(x)$ and $g(x)$ are co-primes polynomials with degree η_1 and η_2, respectively. Suppose P_0 and Q_0 are the respective generators of pairing groups G_1 and G_2. Given hierarchy $(\mathbb{C}, \geq)$, $\mathbb{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_n\}$ and the following instance

$$P_0, [\alpha]P_0, [\alpha^2]P_0, \ldots, [\alpha^{\eta_1 - 1}]P_0 \quad (1)$$
$$[\omega]P_0, [\omega\alpha]P_0, [\omega\alpha^2]P_0, \ldots, [\omega\alpha^n]P_0 \quad (2)$$
$$Q_0, [\alpha]Q_0, [\alpha^2]Q_0, \ldots, [\alpha^{\eta_2 - 1}]Q_0; \quad (3)$$
$$[\omega]Q_0, [\omega\alpha]Q_0, [\omega\alpha^2]Q_0, \ldots, [\omega\alpha^n]Q_0 \quad (4)$$
$$[\frac{f(\alpha)}{f(\alpha, \mathbb{C}_1)}]P_0, [\frac{f(\alpha)}{f(\alpha, \mathbb{C}_2)}]P_0, \ldots, [\frac{f(\alpha)}{f(\alpha, \mathbb{C}_n)}]P_0 \quad (5)$$
$$[\alpha f(\alpha)]P_0, [\alpha^2 f(\alpha)]P_0, \ldots, [\alpha^n f(\alpha)]P_0 \quad (6)$$
$$[\gamma\alpha f(\alpha)]P_0, [\gamma\alpha^2 f(\alpha)]P_0, \ldots, [\gamma\alpha^n f(\alpha)]P_0 \quad (7)$$
$$[\frac{\gamma f(\alpha)}{f(\alpha, \mathbb{C}_i^*)}]P_0 \quad (8)$$
$$e(P_0, Q_0)^{f(\alpha)} \quad (9)$$

the $(\eta_1, \eta_2, \epsilon)$-$aMSE$-$DDH$ problem decides whether the element $R = e(P_0, Q_0)^{\gamma f(\alpha)}$ or a random element R of G_3.

For all t-polynomial time adversaries, if the maximum advantage of solving $(\eta_1, \eta_2, \epsilon)$-$aMSE$-$DDH$ problem is ϵ, then the problem is considered a $(\eta_1, \eta_2, \epsilon)$-hard problem. Note that $|\mathbb{C}_i^*| \geq \eta_2$, where the class $\mathcal{C}_i^*$ is to be challenged.

Remark 5. *Without loss of generality, we consider the general hierarchy in the security proof. However, it becomes more harder if we embed the time-slots into user keys. In general hierarchy, $\mathbb{U}_{il} = \mathbb{C}_i$ for all $i = 1, 2, \ldots, n$. The attack model we consider is that the successor classes collaboratively try to derive the predecessor class decryption keys.*

Theorem 1. *If $(\eta_1, \eta_2, \epsilon)$-$aMSE$-$DDH$ assumption holds true, then the proposed THACS scheme is selectively secure under chosen-ciphertext attack.*

Proof. Suppose there is an adversary $\mathcal{A}$ who can break the security of the proposed $THACS$ scheme with the advantage ϵ. Then, we can construct an algorithm $\mathcal{B}$ that is able to solve the $(\eta_1, \eta_2, \epsilon)$-$aMSE$-$DDH$ problem with advantage ϵ at least by interacting with $\mathcal{A}$ as follows. The attack is model;ed as the successor security class users will try to derive collaboratively their predecessor decryption/user keys.

Initialization: The adversary $\mathcal{A}$ submits a security class $\mathcal{C}_i^*$ to be challenged. The attack scenario is that all the successor class users of $\mathcal{C}_i^*$ are try to to derive the user key of class $\mathcal{C}_i^*$. Without loss of generality, we assume that challenged class is not root class, that is $\mathcal{C}_i^* \neq \mathbb{C}$ and the successors of $\mathcal{C}_i^*$ are collude and decrypt the content encrypted for $\mathcal{C}_i^*$. The $\mathcal{B}$ will set the two polynomials $f(x)$ and $g(x)$ for the security class $\mathbb{C}_1$ as follows.

$$g(x) = f(x, \mathbb{C}_i^*) = \prod_{x_c \in \mathbb{C}_1} (x + h(x_c))$$
$$f(x) = \prod_{x_c \in (\mathbb{C} - \mathbb{C}_i^*)} (x + h(x_c))$$

where $f(x)$ is η_1-degree polynomial and $g(x)$ is η_2-degree polynomial.

Setup: The challenger $\mathcal{B}$ sets the master secret key α which is same as used in the challenge instance. Next, $\mathcal{B}$ first sets $P = [f(\alpha)]P_0$ and $Q = Q_0$. Then simulates the other public key components as follows:

$$P_c = [\alpha^c]P = [\alpha^c f(\alpha)]P_0$$
$$Q_c = [\alpha^c]Q_0$$
$$B_i = [\frac{f(\alpha)}{f(\alpha, \mathbb{C}_i)}]P_0$$
$$e(P, Q) = e(P_0, Q_0)^{f(\alpha)}$$

The value $e(P_0, Q_0)^{f(\alpha)}$ is simulated from instance equation 1 and Q_0, $[\alpha]Q_0$, $f(x)$. The $\mathcal{B}$ sends the public key parameters to the adversary $\mathcal{A}$ except hash function h that sets as random oracle.

Hash Query: The adversary can access hash oracle h and $\mathcal{B}$ maintains a list $\mathcal{L}$ to record query and response. If query responded and recorded in the list, then $\mathcal{B}$ responds with the same result. Otherwise, $\mathcal{B}$ works as follows. Let the query to h is x_c. If $x_i \notin \mathbb{C}_i$, then $\mathcal{B}$ responds $h(x_c)$ with a random number in Z_q. Otherwise, for $x_c \in \mathbb{C}_{it}$, we have two cases

- If $x_c \in \mathbb{C}_i^*$, the $\mathcal{B}$ responds $h(x_c)$ as a new root $g(x)$.
- Otherwise, $\mathcal{B}$ responds $h(x_c)$ as a new root of $f(x)$.

Phase-I: In this phase, we divide users into two groups, one is honest group and other is corrupted group. For honest users, not successors of challenged class, query on key, $\mathcal{B}$ selects random keys and store it in his table.

- For the corrupted users, they are all successor class users, query on key, $\mathcal{B}$ responds as follows:
  For a user key query on $\mathbb{C}_i$, we can write $f(\alpha, \mathbb{C}_i)$ as
  $$f(\alpha, \mathbb{C}_i) = \sum_{x_c \in \mathbb{C}_i} (x + h(x_c)) = f_f(\alpha, \mathbb{C}_i) f_g(\alpha, \mathbb{C}_i)$$
  where all roots of $f_f(\alpha, \mathbb{C}_i)$ are from $f(x)$ and all the roots of $f_g(\alpha, \mathbb{C}_i)$ are from $g(x)$. If $\mathbb{C}_i$ is successor of the class $\mathcal{C}_i^*$, the degree of $f_f(\alpha, \mathbb{C}_i)$ must zero.

Now, $\mathcal{B}$ choose $s \in Z_q^*$ and sets $s_{il} = swx + 1$. If $\mathcal{A}$ ask a query on $\mathbb{C}_i$, then the $\mathcal{B}$ simulates K_{l1} and K_{l2} as follows:

$$K_{l1} = \Big[\frac{s-1}{\alpha}\Big]Q = \Big[\Big(\frac{sw\alpha}{\alpha}\Big)]Q_0 = [sw]Q_0$$
$$K_{l2} = [s_{il} f(\alpha, \mathbb{C}_i)]Q$$
$$= [(sw\alpha + 1) f_g(\alpha, \mathbb{C}_i)]Q_0$$
$$= [sw\alpha f_g(\alpha, \mathbb{C}_i)]Q_0 + [f_g(\alpha, \mathbb{C}_i)]Q_0$$

Since $\mathbb{C}_i \subset \mathbb{C}_i^*$, the polynomial $f_g(x, \mathbb{C}_i)$ is $(\eta_2 - 1)$-degree at most polynomial and so it is computable.

Now, $\mathcal{B}$ responds with $K_{il} = \{K_{l1}, K_{l2}\}$ to adversary $\mathcal{A}$. In addition, $\mathcal{A}$ can also ask decryption query. In this case, $\mathcal{B}$ responds with M if it is in the list, that means it is same as the previous query. Otherwise, $\mathcal{B}$ outputs $\perp$. No query is aborted.

Challenge: The adversary $\mathcal{A}$ sends two messages (M_0, M_1) for challenge where all queried user subscription keys do not satisfy the access class $\mathcal{C}_i^*$. The $\mathcal{B}$ chooses $b \in \{0, 1\}$. Next, $\mathcal{B}$ sets $r_{it} = \gamma_{it}$ and chooses random $X^* \in \{0, 1\}^\rho$. Then computes the challenging ciphertext as follows and sends it to adversary $\mathcal{A}$.

$$B_{i\gamma}^* = [\frac{\gamma f(\alpha)}{f(\mathbb{C}_i^*)}]P_0$$
$$P_{cr} = [\gamma]P_c = [\gamma\alpha^c f(\alpha)]P_0$$
$$\Psi_{ir} = Y^*$$

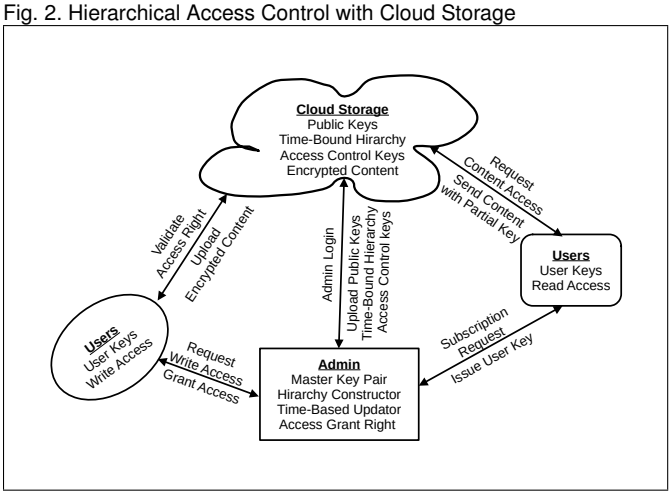
Phase-II: $\mathcal{B}$ responds to the adversary $\mathcal{A}$ same as in the Phase-I, except both the subscription key query which fulfill the challenged class and the decryption query.

Guess: The adversary $\mathcal{A}$ outputs a guess b' of b. If $b = b'$, the adversary $\mathcal{A}$ wins the game. If no query on K_{il}, the adversary $\mathcal{A}$ has no advantage in guessing the encrypted message except with probability $\frac{1}{2}$. So, $\mathcal{A}$ has advantage ϵ in solving proposed scheme, and therefore, $\mathcal{B}$ can solve the $(\eta_1, \eta_2, \epsilon)$-$aMSE$-$DDH$ problem with advantage greater than ϵ. This completes the proof. □

6 Cloud-Based Applications

We consider a cloud-based broadcast application as shown in Figure 2. This framework comprises of four parties such as Cloud Service Provider (CSP), Data Owner ($Admin$), End Users (U). We further divided the users into two kinds based-on the access rights, like *read* and *write*. We assume that the end users with *subscription key* will have the read access and other users we considered as the users with *write* access (broadcast content). However, in the proposed model, subscribed users generally having *read* access with their user key. But, *write* access is simply based-on access of master public key and hierarchy control keys. The application of our scheme to the above scenarios is as follows:

Fig. 2. Hierarchical Access Control with Cloud Storage



1) The user subscription key is divided into two partial parts: First part is (ID_l, K_{l2}), stores it in the cloud; Second part is (ID_l, K_{l1}), issues it to the user as user secret key. Note that K_{l2} will be stored at cloud and can be used as the revocation parameter of the user U_l.
2) In order to avoid the miss use of partial key by unauthorized users, we can mask the first-part key by cloud's private key k_{csp}, that is, compute K_{l2} as $K_{l2} = [s_{il} f(\alpha, \mathbb{U}_{il}]Q_{csp}$, where $Q_{csp} = k_{csp}Q$ is the public key of CSP, and sets $ID_l = e(P,Q)^{s_{il}F(\alpha,\mathbb{U}_{il},\mathbb{C}_i)}$ as identification parameter.
3) When the user with identity ID_l, registered with $\mathcal{C}_i$, requests class access, then CSP checks the database for ID_l. If exists, then checks the validity of $ID_l = e(B_i, K_{l2})^{1/k_{csp}}$. If matches, then consider the request and computes the partial key for the requested content as follow:

$$U = e(P_{1r}, \prod_{c=1}^{N}[F_c]Q_{c-1}) = e(P,Q)^{r_{jt}(F(\alpha)-F_0)}$$
$$V = e(B_{jr}, K_{l2})^{1/k_{csp}} = e(P,Q)^{r_{jt}s_{il}F(\alpha)}$$
$$X = \prod_{c=1}^{N+1}[F_{c-1}]P_{cr}$$

where $F(x) = F(x, \mathbb{U}_{il}, \mathbb{C}_{jt}) = \sum_{c=0}^{N} F_c x^c$.

The CSP then sends the partial key with cipher $\{U, V, X, F_0, \Psi_{jr}\}$ to user $\mathcal{U}_l$.
4) User $\mathcal{U}_l$ computes content decryption key as

$$W = e(X, K_{l1}) = e(P,Q)^{s_{il}r_{jt}F(\alpha) - r_{jt}F(\alpha)}$$
$$k_{jr} = \left(\frac{V}{UW}\right)^{1/F_0} = e(P,Q)^{r_{jt}}$$

Finally, decrypt text as $\Phi_{jr} = D(k_{jr}, \Psi_{jr})$ using the derived key k_{jr} of class $\mathcal{C}_j$, which is generated in the time-interval $\mathcal{T}_t$.

Remark 6. *In the above application, the complexity is reduced to one bilinear pairing and one group exponentiation operation for derivation of content decryption key. That is, irrespective of the hierarchical structure, the computational overhead to derive decryption key is constant.*

Future scope of the work: In the existing works including proposed work, the user key is used to read the encrypted content. The public-key parameters reflect the write access. However, this write access is not exactly reflecting the write access defined in the role-based access control [2]. So, we can further extend the cryptographic access control to restrict the right "write access" in the class level. That is, in each class level the users can encrypt content to their class only, and also can restrict the right to specified time slots. Trivially, we can achieve the above feature by restricting the access to the public parameters which are used to generate class content cipher, that means only the authorized *write access* users can write the content to the respective classes.

7 Conclusion

In this paper, we have proposed a novel time-bound hierarchical access control scheme for subscription-based services. The proposed scheme offers constant size keys and restrict access to specified time-slots. The individual user keys are generated by embed the requested subscription time-slots so that it is not required to broadcast the temporal keys periodically. We proved that the proposed access control scheme is selectively secure under the chosen-ciphertext attack. We then presented the cloud-based application to show the strategies to reduce end user computational overheads and efficient user revocation. The proposed scheme supports dynamic properties along with reduced computational overheads for users. As a conclusion, our proposed scheme is best suitable for practical applications.

References

[1] E. Bertino, P. A. Bonatti, and E. Ferrari, "Trbac: A temporal role-based access control model," *ACM Transactions on Information and System Security (TISSEC)*, vol. 4, no. 3, pp. 191–233, 2001.

[2] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based access control models," *Computer*, vol. 29, no. 2, pp. 38–47, 1996.

[3] J. B. Joshi, E. Bertino, U. Latif, and A. Ghafoor, "A generalized temporal role-based access control model," *IEEE transactions on knowledge and data engineering*, vol. 17, no. 1, pp. 4–23, 2005.

[4] Y. Zhu, G.-J. Ahn, H. Hu, D. Ma, and S. Wang, "Role-based cryptosystem: A new cryptographic rbac system based on role-key hierarchy," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 12, pp. 2138–2153, 2013.

[5] S. G. Akl and P. D. Taylor, "Cryptographic solution to a problem of access control in a hierarchy," *ACM Transactions on Computer Systems (TOCS)*, vol. 1, no. 3, pp. 239–248, 1983.

[6] R. S. Sandhu, "Cryptographic implementation of a tree hierarchy for access control," *Information Processing Letters*, vol. 27, no. 2, pp. 95–98, 1988.

[7] V. Odelu, A. K. Das, and A. Goswami, "A secure effective key management scheme for dynamic access control in a large leaf class hierarchy," *Information Sciences*, vol. 269, pp. 270–285, 2014.

[8] S. Tang, X. Li, X. Huang, Y. Xiang, and L. Xu, "Achieving simple, secure and efficient hierarchical access control in cloud computing," *IEEE Transactions on Computers*, vol. 65, no. 7, pp. 2325–2331, 2015.

[9] A. Castiglione, A. De Santis, B. Masucci, F. Palmieri, A. Castiglione, and X. Huang, "Cryptographic hierarchical access control for dynamic structures," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 10, pp. 2349–2364, 2016.

[10] W.-G. Tzeng, "A time-bound cryptographic key assignment scheme for access control in a hierarchy," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 1, pp. 182–188, 2002.

[11] E. Bertino, N. Shang, and S. S. Wagstaff Jr, "An efficient time-bound hierarchical key management scheme for secure broadcasting," *IEEE transactions on dependable and secure computing*, vol. 5, no. 2, pp. 65–70, 2008.

[12] F. Guo, Y. Mu, W. Susilo, D. S. Wong, and V. Varadharajan, "Cp-abe with constant-size keys for lightweight devices," *IEEE transactions on information forensics and security*, vol. 9, no. 5, pp. 763–771, 2014.
[13] P. He, K. Xue, J. Yang, Q. Xia, J. Liu, and D. S. Wei, "Fase: Fine-grained accountable and space-efficient access control for multimedia content with in-network caching," *IEEE Transactions on Network and Service Management*, vol. 18, no. 4, pp. 4462–4475, 2021.
[14] C. K. Chaudhary, R. Sarma, and F. A. Barbhuiya, "Rma-cpabe: A multi-authority cpabe scheme with reduced ciphertext size for iot devices," *Future Generation Computer Systems*, vol. 138, pp. 226–242, 2023.
[15] L. Zhou, V. Varadharajan, and M. Hitchens, "Achieving secure role-based access control on encrypted data in cloud storage," *IEEE transactions on information forensics and security*, vol. 8, no. 12, pp. 1947–1960, 2013.
[16] N. H. Sultan, V. Varadharajan, L. Zhou, and F. A. Barbhuiya, "A role-based encryption (rbe) scheme for securing outsourced cloud data in a multi-organization context," *IEEE Transactions on Services Computing*, 2022.
[17] J. Herranz, F. Laguillaumie, and C. Ràfols, "Constant size ciphertexts in threshold attribute-based encryption," in *International Workshop on Public Key Cryptography*. Springer, 2010, pp. 19–34.

Vanga Odelu (SMIEEE) received the Ph.D. degree in Theoretical Computer Science (Cryptography and Network Security) and the Master of Technology (M. Tech.) degree in Computer Science and Data Processing both from Indian Institute of Technology (IIT) Kharagpur, India. Master of Science (M.Sc.) degree in Applied Mathematics from Kakatiya University, Warangal, Telangana, India. He is currently working as an Assistant Professor in the Department of Computer Science and Engineering at Indian Institute of Information Technology Sri City, Chittoor, Andhra Pradesh, India. Prior to this, he was with Birla Institute of Technology & Science (BITS), Pilani, Hyderabad Campus and Korea University, South Korea. He was awarded Outstanding Potential for Excellence in Research and Academics (OPERA) by BITS Pilani. He was selected as an Outstanding Young Foreign Scholar "Korean Research Fellowship (KRF-2017)" by the Korean Government (Global competition among 15 positions). He was cleared Council of Scientific & Industrial Research - Junior Research Fellowship (CSIR-JRF, India) exam in December 2008 in Mathematical Sciences and also cleared Graduate Aptitude Test in Engineering (GATE-2009, India) in Mathematics and secured All India Fifth Rank. He is an Organizing Chair of International Conference on Mining Intelligence and Knowledge Exploration (MIKE-2019). Track Chair for the Intelligent Security Systems (MIKE-2017 & 2018), and also Member of Advisory Committee MIKE-2018. He is a guest editor for Topical Issue - Security and Privacy 2020, SN Computer Science, Springer. He is an active reviewer for several SCI-indexed journals including ACM/IEEE Transactions and Technical Program Committee member for several reputed International Conferences. His current research interests include cryptography, network security, access control, and blockchain security. He has authored over 60 papers in international journals and conferences.