A Delegatable Attribute Based Encryption Scheme for a Collaborative E-Health Cloud

Harsha Sandaruwan Gardiyawasam Pussewalage, *Member, IEEE, Vladimir Oleshchuk, IEEE, Senior Member, IEEE*

***Abstract*—With the popularization and growing utilization of electronic health records (EHRs) coupled with the advancements in cloud computing, healthcare providers are interested in storing EHRs in third-party, semi-trusted cloud platforms. Given the collaborative nature of modern e-health environments, integrating access delegation is of paramount importance to strengthen the flexibility of the sharing of health information. However, access delegation has to be enforced in a controlled manner so that it will not jeopardize the security of the system. For such applications, attribute based encryption (ABE) mechanisms are quite useful given the fact that ABE facilitates an efficient way of enforcing secure, fine-grained access control over encrypted data. However, incorporating delegatability with ABE mechanisms is tricky, and the existing schemes lack the control over the process of delegation of encrypted data. As a solution, we propose a novel ABE based access control scheme which can enforce multi-level, controlled access delegation and demonstrated how it could be deployed in an e-health environment to securely share outsourced EHRs of patients. Furthermore, we have shown that the proposed scheme is secure against chosen plaintext attacks as well as attacks mounted via attribute collusion.**

***Index Terms*—Access control, attribute based encryption, controlled delegation, security, privacy**

I. Introduction

THE emergence of electronic healthcare (e-health) solutions has revolutionized healthcare systems through providing a means for efficient sharing of health resources.Such systems intend to provide healthcare services with high efficiency and flexibility while establishing a platform for secure sharing of health related data across different healthcare settings and work-flows among different healthcare providers. The discussion on progressing towards ehealth received a significant boost especially in United States (US), with the US Institute of Medicine issuing a major report in 1991, indicating the necessity of adopting computer based patient records. With this transition, paper based records advanced to their respective digitized electronic versions; commonly termed as electronic health records (EHRs).

The advancements in cloud computing have become a significant factor in the progression of e-health, given that it allows EHRs of patients to be stored remotely in cloud platforms. This approach not only ensures better availability of patients' health information but also helps in reducing the overhead associated with health information management from the care providers. Although such benefits exist, the privacy of the remotely stored data has become a major concern, mainly because the cloud platforms are managed by thirdparties who may have an interest on the stored data [1], [2].Besides, such storage servers could become targets for various malicious activities and may lead to illegitimate disclosure of patients' privacy sensitive information [3]. Hence, utilizing appropriate measures to safeguard the outsourced private health data is of utmost importance.

A promising approach would be to adopt attribute based encryption (ABE) schemes [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14] which allow EHR data to be encrypted by the care provider according to an access policy which defines the potential users who are eligible to access before being stored in a cloud infrastructure. Thus, a user who is having a set of valid attributes that satisfies the governing access policy could potentially decrypt the encrypted EHR data with the associated attribute keys upon downloading the relevant ciphertexts from the cloud.

Generally, an e-health environment is a collaborative environment meaning that a treatment process of a patient may require collaborative efforts of different parties including emergency staff members, healthcare professionals in the patient's home care provider as well as professionals from foreign healthcare providers (FHPs) in certain situations [3],[15]. In such environments, a user's ability to delegate access is quite vital to achieving timely and flexible sharing of EHRs of patients [16], [17], [18]. For instance, consider the following scenario. Suppose, Alice is a patient of hospital A and her EHR stored in hospital A is associated with an attribute access structure $\mathcal{T} = (Cardiologist \wedge Hospital\_A)$ giving permission to cardiologists in hospital A to access Alice's EHR. After a recent consultation session, Dr.

Bob who is a cardiologist at hospital A finds some anomalies in Alice's blood test results and wants to refer the recent findings to oncologists affiliated with hospital A. With the presence of delegation capability, Dr. Bob is able to

THE The authors are with the Department of Information and Communication Technology, University of Agder (UiA), N-4898 Grimstad, Norway.
E-mail: harsha.sandaruwan, vladimir.oleshchuk@uia.no.

Manuscript received 26 March 2021; revised 9 March 2022; accepted 5 May 2022. Date of publication 13 May 2022; date of current version 10 April 2023.(Corresponding author: Harsha Sandaruwan Gardiyawasam Pussewalage.) Digital Object Identifier no. 10.1109/TSC.2022.3174909
IEEE TRANSACTIONS ON SERVICES COMPUTING, VOL. 16, NO. 2, MARCH/APRIL 2023 787

temporarily delegate the access to the necessary sections of Alice's EHR (EHR objects) for oncologists affiliated to hospital A.Incorporating such delegating capability induces the following challenges which need to be taken into consideration.First of all, Dr. Bob may want to ensure that delegated access is only granted for the intended EHR objects. Note that other EHR objects of Alice's EHR might be associated with the same access structure, but the access to such EHR objects must be prevented to oncologists in hospital A, given that it may potentially violate the privacy of the patient. Suppose, Dr. John who is an oncologist in hospital A accessed the delegated EHR objects of Alice's EHR and wants to get some further expertise from Dr. Charlie (oncologist affiliated with hospital B) through further delegation of access to oncologists in hospital B. However, this delegation must only be feasible, if and only if Dr. Bob has allowed the option of further delegations. We refer the delegation of access complying with the challenges mentioned above as controlled access delegation.

As we have pointed out, ABE capable of providing an efficient and secure way of outsourcing data to third-party cloud platforms while provisioning users with fine-grained access to the outsourced data. However, when the data are encrypted and stored under the control of a semi-trusted commodity, allowing access via delegation becomes tricky, especially considering the fact that the delegation could be a many-to-many delegation as given in the aforementioned example (delegating access to any oncologist in hospital A) and there has not been much progress made in enabling access delegation in a controlled manner. To address this gap, in this paper, we propose an efficient ABE based access control mechanism, which is not only capable of provisioning its users with fine-grained access to the outsourced data but also allowing the users to delegate access (i.e. multilevel, many-to-many access delegation) to the outsourced data in a controlled manner when required.

II. RELATED WORK

When we consider the cloud based health information sharing systems, ABE is influential due to the fact that it can enforce fine-grained access control over the encrypted data.Thus, we begin the section by summarizing the ABE based EHR sharing schemes. Thereafter, we present the existing work related to the main focus of this research - facilitating access delegation over the encrypted data.

The health information sharing scheme proposed in [19] is one of the initial research work which utilized ABE for the purpose of sharing health information. This solution uses the ciphertext-policy attribute based encryption (CP-ABE) scheme in [20] while introducing the concept of personal and public domains. It consists with a central trusted authority (TA) for managing attributes in the public domain while the data owner or the patient himself acting as the TA for the personal domain for the purpose of issuing attributes relevant for the personal domain. However, the main issue is the use of a single TA for administrating the user attributes of the public domain which could not only lead to a single point of failure but also may cause key-escrow problems given the fact that the TA can access all the encrypted data. Furthermore, similar cloud based personal health information sharing schemes with a central TA can be found in [21], [22], [23], [24]. In the quest for dealing with the problem of centralized TA, a new variant of ABE known as multi-authority attribute based encryption (MA-ABE) has been introduced. Such schemes [7], [9], [25], [26] equipped with multiple authorities to manage attributes and thereby eliminates the issue of single point of failure. Furthermore, the solutions proposed in [3], [7] provide evidence for the utilization of MA-ABE in e-health environments.

Although many ABE schemes have been proposed to this date, there have only been few related works which have explored the issue of access delegation on data encrypted with ABE schemes. In [27], [28], the authors have proposed ABE based mechanisms to delegate access for encrypted data by allowing users to delegate their attributes. Among them, the solution in [28] allows a user (delegator) to delegate a subset of the attributes he owns to another user (delegatee) by issuing secret keys. This scheme uses a central TA to issue attributes in the form of secret keys for the higher level users, and with delegatability, it is possible for the other users to obtain secret keys from the higher level users.Thus, the computation and management overhead imposed on the TA is substantially reduced. However, this scheme is not equipped with any mechanism to control subsequent delegations made by the higher level users or the delegatees. This means that a user U_2 who obtained attributes from another user U_1 will be able to re-delegate the delegated attributes without the consent of the initial delegator U_1 . Furthermore, the scheme in [27] only facilitates the central TA to carry out revocation meaning that the intermediate delegators cannot revoke the delegatees who acquired attributes from them. This issue is addressed in [28] using a third-party mediator which facilitates the delegation process. During delegation of attributes, the delegator needs to inform the mediator about the impending delegation including which attributes are delegated. Then, the delegatee can obtain the relevant delegated attribute keys from the delegator and the mediator as partial keys. This induces some control over the process of delegation, given that delegator's permission over re-delegation of attributes is enforced through the mediator. However, since the mediator has a full view of who delegates which attribute to whom, might affect the privacy of users.

Proxy re-encryption (PRE) schemes can also be used for achieving access delegation over encrypted data. In general, PRE allows a third-party entity (proxy) to translate a ciphertext encrypted under a specific entity's public key in such a way that the translated ciphertext can be decrypted by a different entity's private key. During the process of conversion, the proxy will not learn either of the decryption key or the plaintext. Although, the traditional PRE schemes have been primarily developed for enforcing one-to-one access delegation [29], [30], [31], [32], [33], [34], the integration of PRE with ABE has made it possible to achieve many-to-many access delegation [35], [36], [37]. However, the schemes presented in [**?**], [37] have no control over re-delegation while the scheme in [37] has control over the re-delegation to a

certain extent. This is due to the fact that the data owner who initially encrypts the data can allow or deny the ability of re-encryption of the ciphertext and thereby the further delegations. However, if the data encryptor has allowed the ciphertext to be delegated, further re-delegations by the delegatees will be uncontrolled. In addition, all the aforementioned solutions allow a delegatee to access all the data that can be accessed by the delegator with the delegated attributes. Hence, a delegator will not be able to selectively provide access only to a subset of data (selective delegatability) that can be accessed with the delegated attributes. This is also an important feature that needs to be considered when enforcing access delegation.

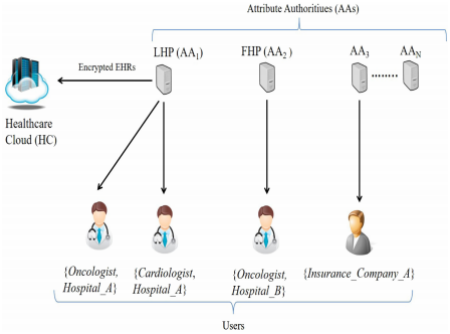


Fig. 1. System model.

III. Our Contributions

As we have pointed out in Section 2, the existing schemes suffer from the inability of provisioning controlled access delegation over the data encrypted with ABE which is essential in enforcing flexible, fine-grained access in multiuser environments, especially when the data are stored under the control of third-party entities. As a solution, in this paper, we extended the multi-authority ciphertext-policy ABE scheme (MA-CP-ABE) in [7] to facilitate controlled access delegation on EHR data of patients stored in an ehealth cloud. The proposed scheme is referred to as the multi-authority ciphertext-policy re-encryption (MA-CPRE) scheme. The novelty of the proposed MA-CP-RE scheme is as follows.

- The scheme is capable of provisioning multi-level, many-to-many controlled access delegation. This means that a user who is eligible to decrypt a specific ciphertext can delegate the access to that ciphertext based on attributes to a set of users. Furthermore, any user who is eligible to access the delegated ciphertext can re-delegate it further, if and only if the preceding delegator has given the permission to do so.
- Unlike the existing ABE schemes with delegation capabilities, where a delegatee becomes eligible to access all data items associated with the delegated attributes, our scheme enables a user to delegate access to a subset of data that can be accessed with the delegated attributes (selective delegatability of data).

The rest of the paper is organized as follows. In Section 4, we present the case description as well as the associated security requirements. The background information associated with the proposed scheme is presented in Section 5.We provide a general overview of the proposed MA-CP-RE scheme in Section 6 while the functionality of the scheme is described in detail in Section 7. The security analysis of the proposed scheme is given in Section 8 while in Section 9, we evaluate the performance of the proposed scheme with respect to the associated computational cost. Thereafter, we compare our scheme with similar existing schemes with respect to their end-user computational complexity and delegatability characteristics in Section 10. Finally, the paper is concluded in Section 11.

IV. Case Description and Security Requirements

In this section, we introduce the health information sharing scenario to illustrate the applicability of the proposed MACP-RE scheme. We also present the security requirements.

which must be satisfied in order to realize a secure and flexible access control scheme.

A. Case Description

We consider a local healthcare provider (LHP) which is responsible for providing healthcare services for people who reside in a specific geographical area. Every patient who is registered in LHP has an associated EHR. LHP stores all EHRs of patients on a third-party cloud platform, which we denote as the healthcare cloud (HC). We require EHRs of patients to be securely shared among a set of users according to the access restrictions imposed by LHP on EHRs. Fig. 1 represents the system model and its main components are defined below.

- *Attribute Authorities (AAs):* Each AA is a trusted entity responsible for managing a set of attributes and issuing attributes to users in the form of secret keys. We assume that LHP also acts as an AA to issue attributes specific to LHP.
- *Users:* We call the subjects who are eligible to access the stored EHRs of patients as the users. Subjects include healthcare professionals affiliated with LHP and FHPs as well as members of other organizations such as insurance companies. Note that a user could also function as a delegator or a delegatee during access delegation. When a user is delegating the access for a particular EHR of a patient (for which the user has access) to a set of users (based on their attributes), the user who is delegating the access we call as the delegator. Furthermore, we call the set of users who receive the access as the delegatees.

We assume that LHP is honest and trusted whereas HC is semi-trusted. This means that HC will follow the specified operational protocol while being curious about the stored data. It is also assumed that the users may be curious about the stored data and potentially interested in extracting more information than what they are allowed via colluding with other users.

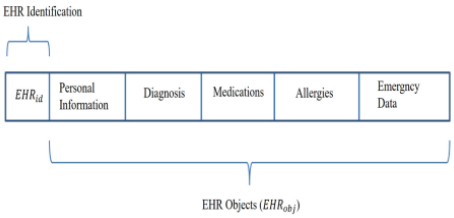


Fig. 2. Structure of an EHR of a patient.

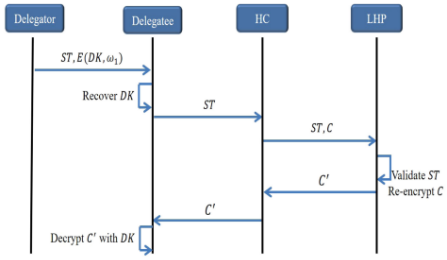


Fig. 3. EHR access delegation mechanism.

B. Security Requirements

The main security requirements to be satisfied in the proposed EHR sharing scheme are summarized as follows

- *Confidentiality of EHR data:* EHR data of patients must be kept secret from unauthorized parties. Any user who does not possess enough attributes to satisfy the access requirements must be prevented from granting access to EHR data of patients even under attribute collusion.
- *Controlled access delegation:* Access delegation on patients' EHR data must be controlled meaning that further delegations by a delegatee are only feasible with the consent of the delegator.
- *Resistance to attribute collusion:* Users should be prevented from colluding their attributes to gain access to the stored EHRs of patients.
- *User revocation:* The proposed scheme should be equipped with a mechanism (which we refer to as user revocation) to revoke the eligibility of users to access the health information of patients when required.

V. PRELIMINARIES

This section is dedicated to providing the necessary background information, which enables us to describe the functionality of the proposed MA-CP-RE scheme in detail in Section 7.

A. EHR Access Structure

We associate each EHR of a patient with a unique EHR identification EHR_{id}. Each EHR can contain different data categories such as personal information, diagnosis, medications, allergies, emergency data, etc. and we define them as EHR objects $(EHR_{obj})$of an EHR. Hence, each EHR (with a unique EHR_{id} can have many different EHR objects as shown in Fig. 2. Moreover, we associate each (EHR_{obj}) with an access structure $(\mathcal{T})$ which governs the attribute requirement for accessing the specific . An access structure is defined as a Boolean statement with the disjunction V and conjunction $(\wedge)$operations combining subject attributes.An example access structure $\mathcal{T}_{I,O}$ for the $EHR_{id} = I$ corresponding to $EHR_{id} = I$ is shown below.

$$\mathcal{T}_{I,O} : (\ Physician\ \vee\ Cardiologist\ ) \wedge (\ Hospital_A\ )$$

This statement states that any user who is employed as a cardiologist or a physician at hospital A is authorized to access the $\text{EHR}_{\text{obj}} = O$ associated with the EHR, where $\text{EHR}_{\text{id}} = I$.

B. Access Sub-Structures

We represent an access structure T as the disjunction of a set of sub-structures $\{\mathcal{T}_i\}_{i=1.2...q}$ such that,

$$\mathcal{T} = \mathcal{T}_1 \vee \mathcal{T}_2 \vee ... \vee \mathcal{T}_q$$

where each $\mathcal{T}_i$ is a conjunction of some subject attributes.We call each $\mathcal{T}_i$ as an access sub-structure of $\mathcal{T}$. For instance, we can represent $\mathcal{T}_1$ as the disjunction of two sub-structures $\mathcal{T}_1$ and $\mathcal{T}_2$ such that,

$\mathcal{T}_{\text{I,O}}$: (Physician $\wedge$ Hospital_A) $\vee$ (Cardio log ist $\wedge$ Hospital_A)

$\mathcal{T}_1$:Physician $\wedge$ Hospital_A,$\mathcal{T}_2$:Cardio log ist $\wedge$ Hospital_A

VI. OVERVIEW OF THE MA-CP-RE SCHEME

As presented in Section 4, our system consists of a set of distributed AAs. Each AA manages a disjoint set of attributes and issues attributes for the users upon validating their eligibility. It is assumed that the secret attribute keys are securely handed over to the corresponding user.

LHP acts as the owner of EHRs of the registered patients.

To outsource an EHR_{obj} to HC, LHP first constructs the access structure $\mathcal{T}$. Note that the attributes in $\mathcal{T}$ can have a combination of attributes from one or more AAs. Then, the EHR_{obj} is encrypted with the help of public attribute keys corresponding to the attributes in $\mathcal{T}$ (details will be given in Section 7.3). The generated ciphertext along with $\mathcal{T}$ is sent to the HC to be stored. When a user is required to access a specific EHR_{obj}, an EHR access request is sent to the HC indicating the EHR_{id} and the relevant EHR_{obj}. The user will only be able to decrypt the encrypted EHR_{obj}, if and only if the user has secret keys corresponding to a set of attributes that satisfies $\mathcal{T}$ associated with the encrypted EHR_{obj}.

The following describes the access delegation mechanism and it is also illustrated in Fig. 3 with the help of a flow diagram. Suppose, a user U (delegator) wants to delegate the access to EHR_{obj}, O (U has necessary attributes ω to access O) to any user who is having the set of attributes ω_1. First, U generates a re-encryption key $RK_{\omega \to \omega_1}$ which is used to translate the ciphertext C(of O stored in HC) into a form that allows it to be decrypted with the attributes ω_1. In addition, U generates a decryption key, DK which facilitates the ciphertext to be decrypted after the re-encryption (using $RK_{\omega \to \omega_1}$). U encrypts $RK_{\omega \to \omega_1}$ with the public key of PK_{LHP} (that is, $E\left(RK_{\omega \to \omega_1}, PK_{LHP}\right)$) and DK with the public attribute keys corresponding to the attributes in ω_1 (that is,

). Furthermore, U generates a token including the information about the access delegated EHR_{obj}, $E\left(RK_{\omega\rightarrow\omega_1}, PK_{LHP}\right)$, some auxiliary information (details are given in Section 7.5) and signs it with the attribute secret keys associated with ω. The resulting signed token (ST) is finally sent to the delegatees along with $E\left(DK, \omega_1\right)$.

When a delegatee requires to access the $EHR_{obj} = O$; the delegatee should send the signed token ST to the HC. This allows HC to determine that the requested access is subjected to a delegation, hence HC will forward ST along with the relevant ciphertext C to LHP. Then, the LHP will validate the token signature, if validated successfully, the reencryption key $RK_{\omega\rightarrow\omega_1}$ is recovered and C is re-encrypted with RKv!v1 . Afterwards, the re-encrypted ciphertext C' is forwarded to the delegatee through HC. Finally, the delegatee will be able to decrypt C' with the help of the decryption key DK.

VII. MA-CP-RE SCHEME

In this section, we present the proposed MA-CP-RE scheme in detail. The functionality of the proposed scheme is presented, by dividing it into eight phases: system initialization, attribute key distribution, EHR encryption, EHR decryption without access delegation, EHR access delegation, EHR decryption under access delegation, how the scheme is extended to multi-level access delegation and the revocation of users.

A. System Initialization

To initialize the system, first, a set of global public parameters is generated which is shared among all AAs. AAs agree on two multiplicative cyclic groups $\mathbb{G}_0, \mathbb{G}_1$ of prime order p with g_0, g_1 being generators of $\mathbb{G}_0$ and a bilinear map e : $e : \mathbb{G}_0 \times \mathbb{G}_0 \rightarrow \mathbb{G}_1$. In addition, an encoding $\mathcal{E} : \mathbb{G}_0 \rightarrow \mathbb{G}_1$ between the groups $\mathbb{G}_0, \mathbb{G}_1$ is also agreed upon along with two secure hash functions H_1, H_2, where $H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_p^*$ and $H_2 : \mathbb{G}_1 \rightarrow \mathbb{Z}_p^*$. AAs also agree on a shared secret $\xi \in \mathbb{Z}_p^*$. Then, AAs publish the set of global public parameters $(\mathbb{G}_0, \mathbb{G}_1\ \mathcal{E}, H_1, H_2, e, g_0, g_1, p)$. After the global initialization of AAs, each AA (including LHP) is locally initialized, and the initialization procedure is described below. We assume that the k^{th}AA is denoted with AA_k while the attribute set administered by AA_k is denoted by AT^k.

- AA_k chooses two random master exponents $\alpha_k, \beta_k \in \mathbb{Z}_p^*$ and computes $X_k = {g_0}^{\beta_k}$, $Y_k = {g_1}^{\beta_k}$. Then a unique random identifier $t_{k,i} \in \mathbb{Z}_p^*$ for each attribute i in AT^k is selected. In addition, each attribute administered by AA_k is also associated with public attribute components $T_{k,i}$; $D_{k,i}$, where $T_{k,i} = g_0^{t_{k,i}}, D_{k,i} = g_1^{t_{k,i}}$.
- AA_k will keep $\{\xi, \alpha_k, \beta_k, t_{k,i}\}_{i=1,2,\ldots,|AT^k|}$ as the master secret denoted by MK_k and publish $\{X_k, Y_k, Z_k\ T_{k,i}, D_{k,i}\}_{i=1,2,\ldots,|AT^k|}$ as the authority's public key denoted by PK_k.

B. Attribute Key Distribution

Users can obtain attribute related secret keys from the relevant AAs by providing evidence that they satisfy the requirements to ascertain the requested attributes. Let us assume that the user U_m wants to acquire attribute keys for the set of attributes AT_m. In addition, assume that AT_m^k denotes the subset of attributes in AT_m which should be acquired from AA_k. AA_k issues the attribute secret keys for the attributes in AT_m^k to U_m as follows.

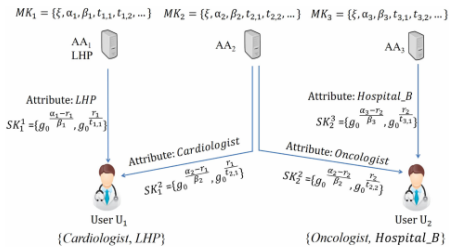


Fig. 4. Attribute key distribution scenario.

- Suppose, the identity of U_m is denoted by ID_m (we may use the public key associated with the user's public key infrastructure (PKI) certificate). Then, AA_k uses the hash function H_1 to map the identity of U_m to a unique identifier $r_m \in \mathbb{Z}_p^*$ such that,$r_m = H_1\left(H_1\left(ID_m\right) + \xi\right)$.
- Thereafter, AA_k generates the secret key set $SK_m^k = \left\{sk_0^k, sk_i^k\right\}_{i=1,2,\ldots,|AT_m^k}$, such that

$$\begin{aligned} sk_0^k &= {g_0}^{(\alpha_k - r_m)/\beta_k} \\ sk_i^k &= {g_0}^{r_m/t_{k,i}}, \end{aligned} \tag{1}$$

  where $t_{k,i}$ is the master secret component of the i^{th} attribute in AT_m^k defined by AA_k. Note that the secret key component sk_0^k relates the user identity to the identity of the issuing authority AA_k whereas the secret key component sk_i^k relates the user identity to the i^{th} attribute in AT_m^k.
- Finally, SK_m^k is securely transferred to U_m.

Fig. 4 illustrates a secret key distribution instance and we use this scenario as an example to illustrate the distribution of secret keys. According to Fig. 4, U_1 owns the attribute set LHP, Cardiologist, where the attribute LHP is issued by the LHP denoted with AA_1 and the attribute Cardiologist is issued by the AA_2. On the other hand, U_2 owns the attribute set Hospital B, Oncologist where the attribute Hospital B is ascertained from AA_3 while the attribute Oncologist is obtained from AA_2. Let us assume that the mapped identifiers for user identities of U_1 and U_2 are r_1 and r_2 respectively.The secret exponent of the attribute LHP at AA_1 is $t_{1,1}$ and secret exponents of the attributes Cardiologist, Oncologist at AA_2 are denoted with $t_{2,1}$ and $t_{2,2}$ respectively. Further assume that the secret exponent related to the attribute Hospital B is given by $t_{3,1}$. U_1 receives attributes from both AA_1 and AA_2. If the corresponding secret key sets are denoted by SK_1^1 and SK_2^1 , then $SK_1^1 = \left\{{g_0}^{(\alpha_1 - r_1)/\beta_1}, {g_0}^{r_1/t_{1,1}}\right\}$ and $SK_1^2 = \left\{{g_0}^{(\alpha_2 - r_1)/\beta_2}, {g_0}^{r_1/t_{2,1}}\right\}$: Similarly, the corresponding secret key sets of U_2 are denoted by SK_2^2 and SK_2^3 where, $SK_2^2 = \left\{{g_0}^{(\alpha_2 - r_2)/\beta_2}, {g_0}^{r_2/t_{2,2}}\right\}$and $SK_2^3 = \left\{{g_0}^{(\alpha_3 - r_2)/\beta_3}, {g_0}^{r_2/t_{3,1}}\right\}$.

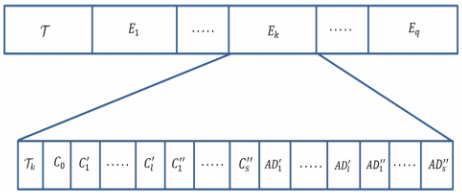


Fig. 5. Structure of the ciphertext *E(M)*.

C. EHR Encryption

Let us assume that LHP wants to encrypt EHR data $M \in \mathbb{G}_1$, which corresponds to the $EHR_{obj} = O$, of a given patient's EHR I. First, LHP generates the access structure $\mathcal{T}$ and deduces a set of access sub-structures $\{\mathcal{T}_k\}_{k=1,2,\ldots,q}$ as explained in Section 5.2. Thus, the ciphertext of M encrypted with $\mathcal{T}$ is given by,

$$E(M) = \left(\mathcal{T}, \{E_k\}_{k=1,2\ldots,q}\right),$$

where E_k denotes the ciphertext of M encrypted with the access sub-structure $\mathcal{T}_k$. The process of computing E_k is described below. Furthermore, we have illustrated the structure of the ciphertext $E(M)$ in Fig. 5.

Let us assume that the k^{th} sub-structure $\mathcal{T}_k$ contains s attributes and they are administered by l AAs such that, $l \leq N$, where N is the total number of AAs in the system. Note that any AA may manage more than one attribute of the considered s attributes. Then, we can represent the ciphertext E_k using ciphertext components $C_0, \{C_i,' AD_i'\}_{i=1,2,\ldots,l}$ and $\{C_i,'' AD_i''\}_{i=1,2,\ldots,s}$, such that,

$$E_k = \left(\mathcal{T}_k, C_0, \{C_i,' AD_i'\}_{i=1,2,\ldots,l,}, \{C_i,'' AD_i''\}_{i=1,2,\ldots,s}\right).$$

Ciphertext components of E_k are computed as follows. LHP first generates a random exponent $h \in \mathbb{Z}_p^*$ and using the public keys of the l AAs, it computes the ciphertext components C_0 and $\{C_i,' AD_i'\}_{i=1,2,\ldots,l}$ such that,

$$\begin{aligned} C_0 &= M \prod_{i=1}^{l} Z_i^h = M e\left(g_0, g_0\right)^{h \sum_{i=1}^{l} \alpha_i} \\ C_i' &= X_i^h = {g_0}^{\beta_i h} \\ AD_i' &= Y_i^h = {g_1}^{\beta_i h}. \end{aligned} \tag{2}$$

To compute $\left\{C_i'' AD_i''\right\}_{i=1,2\ldots,s}$, a secret share of h is assigned for each attribute in $\mathcal{T}_k$ by following the steps given below.

- For each attribute in $\mathcal{T}_k$ except the last, a random exponent $h_i \in \mathbb{Z}_p^*$ is assigned while the last element is assigned the value equals to $l \cdot h - \sum_{i=1}^{s-1} h_i$.
- Then, the LHP computes $\{C_i,'' AD_i''\}_{i=1,2,\ldots,s}$ such that,

$$\begin{aligned} C_i'' &= T_i^{h_i} = {g_0}^{t_i h_i} \\ AD_i'' &= D_i^{h_i} = {g_1}^{t_i h_i} \end{aligned} \tag{3}$$

where T_i, D_i correspond to the public attribute components of the i^{th} attribute in $\mathcal{T}_k$.

Similarly, LHP generates the ciphertexts relevant for all the sub-structures of $\mathcal{T}$ and uploads $E(M)$ along with the EHR identification I and the EHR_{obj} information ($EHR_{obj} = O$) to HC.

D. EHR Decryption Without Access Delegation

Before we present the details on how the proposed MA-CPRE scheme facilitates access delegation, it is important to briefly present the EHR decryption process, without delegation.

Suppose, U_m wants to access a specific EHR_{obj}, O associated with a given EHR stored in the HC. U_m should first send an access request indicating the EHR_{id} and EHR_{obj} information corresponding to the access required EHR to HC. Then, HC fetches the corresponding $\mathcal{T}$ associated with the requested EHR_{obj} and sends it back to U_m. Let us assume that the attribute set owned by U_m is denoted with AT_m. Then, U_m determines the smallest subset of attributes AT_m' that satisfies the received $\mathcal{T}$. Based on AT_m', U_m generates a sub-structure $\mathcal{T}'$ and sends it to the HC.According to the received $\mathcal{T}'$, HC fetches the corresponding EHR ciphertext E' and sends it back to U_m which enables him to decrypt the encrypted data using the relevant attribute secret keys. The decryption process is as follows.

For illustrative purposes, let us assume that the received ciphertext E' is associated with s attributes which are administered by l AAs. Then, $E' = \left(\mathcal{T}' C_0, \{C_i'\}_{i=1,2,\ldots,l} \{C_i''\}_{i=1,2,\ldots,s}\right)$,where C_0, C_i' and C_i'' are given in (3), (4) and (6) respectively. Note that HC will not send the ciphertext components $\{AD_i'\}_{i=1,2,\ldots,l}, \{AD_i''\}_{i=1,2,\ldots,s}$ to U_m, given that these components are only required during decryption subjected to access delegation. Further assume that $\{sk_i\}_{i=1,2\ldots,s}$ denotes the relevant attribute secret key set owned by U_m for the attribute subset AT_m' and $\left\{sk_0^i\right\}_{i=1,2,\ldots,l}$ refers to the set of secret key components which relates the identity of U_m to the l AAs who issued the s attributes. According to (1) and (2),

$$sk_i = {g_0}^{r_m/t_i} \text{ and} sk_0^i = {g_0}^{(\alpha_i - r_m)/\beta_i} \tag{4}$$

where t_i denotes the attribute secret exponent associated with the corresponding attribute. Then, U_m will be able to decrypt E' and discover the EHR data M as follows.

$$\frac{C_0}{\prod_{i=1}^{s} e\left(C_i,'' sk_i\right) \cdot \prod_{i=1}^{l} e\left(C_i,' sk_0^i\right)} = \frac{M \cdot e\left(g_0, g_0\right)^{h \sum_{i=1}^{l} \alpha_i}}{e\left(g_0, g_0\right)^{h \sum_{i=1}^{l} \alpha_i}} = M.$$

E. EHR Access Delegation

Suppose, now U_m wants to delegate the access to the EHR_{obj}, O which is associated with an access structure $\mathcal{T}$ to any user who is having attributes that satisfy an access sub-structure $\mathcal{T}''$ ($\mathcal{T}''$ is not a sub-structure of $\mathcal{T}$). The delegation procedure is divided into three parts as re-encryption key generation, decryption key generation and signed delegation token generation.

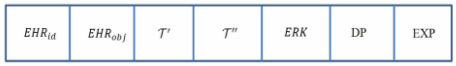


Fig. 6. Structure of the delegation token issued by U_m

1) Re-Encryption Key Generation: As we have shown in Section 7.4, it is possible for U_m to decrypt the object O using his secret keys associated with the sub-structure $\mathcal{T}'$. Hence, U_m uses the attribute keys associated with the attributes in $\mathcal{T}'$ to generate the re-encryption keys. The process of generating the re-encryption keys is described below. Note that AT'_m denotes the attribute subset of U_m associated with the sub-structure $\mathcal{T}'$ and $|AT'_m| = s$. Furthermore, it is assumed that the s attributes have come from l AAs.

- U_m selects a random exponent $d \in \mathbb{Z}_p^*$.
- If the set of re-encryption keys is denoted by RK, then,

$$\begin{aligned} RK &= \left\{ \{RK_i\}_{i=1,2....,s}, \{RK_0^i\}_{i=1,2,...,l} \right\} \\ RK_i &= sk_i \cdot {g_1}^d = {g_0}^{\frac{r_m}{t_i}} \cdot {g_1}^d \\ RK_0^i &= sk_0^i \cdot {g_1}^d = {g_0}^{\frac{\alpha_i - r_m}{\beta_i}} \cdot {g_1}^d \end{aligned} \tag{5}$$

in which sk_i and sk_0^i represent the secret keys of U_m associated with the i^{th} attribute in AT'_m. In order to allow the set of re-encryption keys RK to be only available to the LHP, elements in RK are encrypted with the public key component of LHP (AA_1). To achieve that, U_m selects a random exponent $a \in \mathbb{Z}_p^*$ and computes the encrypted reencryption key,

$$ERK = \left\{ \left\{ RK_i \cdot {g_0}^{a\beta_1} \right\}_{i=1,2...,s}, \left\{ RK_0^i \cdot {g_0}^{a\beta_1} \right\}_{i=1,2...,l}, {g_0}^a \right\}$$

2) Decryption Key Generation: The decryption key allows a user with a set of attributes that satisfies the sub-structure $\mathcal{T}''$ to access the delegated EHR_{obj}, after it is re-encrypted with RK. If the decryption key is given by DK, then $DK = {g_0}^d$. To ensure that DK can only be accessed by a user having attributes that satisfy $\mathcal{T}''$, DK is first encoded using $\mathcal{E}$ (i.e. $\mathcal{E}$ maps DK to an element in $\mathbb{G}_1$) which enables it to be encrypted with the attributes in $\mathcal{T}''$ as explained in Section 7.3. We denote the encrypted decryption key with EDK.

3) Signed Delegation Token Generation: The delegation token (DT) allows a delegatee to provide evidence for the LHP that he has the right to access the delegated EHR_{obj}. The token includes the information on access delegated EHR_{obj}, sub-structure associated with the delegator $(\mathcal{T}')$, sub-structure associated with the delegatees $(\mathcal{T}'')$, encrypted re-encryption key (ERK), delegation permission (DP) index, and the token expiration information (EXP).

The DP index (either 0 or 1) defines the permission for further delegations by the delegatees. DP index of 0 meaning that the delegatees do not have the right to further delegate while an index of 1 allows the delegatees to further delegate the access to the considered EHR_{obj}. The token expiration information determines the validity period of an issued token. Fig. 6 illustrates the structure of the delegation token issued by U_m to delegate the access to the $EHR_{obj} = O$ to delegatees having attributes that satisfy $\mathcal{T}''$.

Before issuing the token to delegatees, U_m must sign the token using the secret keys associated with the attributes in $\mathcal{T}'$ which enables LHP (when a delegatee is claiming the access) to determine that the token is generated by a user who has the right to decrypt the associated EHR_{obj}. The token signature generation is as follows. We considered earlier that $\mathcal{T}'$ contains s attributes issued by l AAs. The secret keys corresponding to these attributes $\{sk_i\}_{i=1,2,...,s}$ $\left\{sk_0^i\right\}_{i=1,2,...,l}$ are given in (8).

- U_m chooses a random exponent $b \in \mathbb{Z}_p^*$ and a set of exponents $\{u_i\}_{i=1,2,...,l}$ where u_i is the number of attributes in $\mathcal{T}'$ which belongs to AA_i.
- Then, U_m generates $\delta_0 = \prod_{i=1}^{l} e\left(g_0, g_0\right)^{\alpha_i b u_i}$ and (σ_1, σ_2) such that,

$$\sigma_1 = \left\{sk_i^b\right\}_{i=1,2,...,s} = \left\{ {g_0}^{\frac{r_m b}{t_i}} \right\}_{i=1,2,...,s}$$
$$\sigma_2 = \left\{ \left(sk_0^i\right)^{u_i b} \right\}_{i=1,2,...,l} = \left\{ {g_0}^{\frac{(\alpha_i - r_m) u_i b}{\beta_i}} \right\}_{i=1,2,...,l}.$$

- Thereafter, U_m computes $Q, R \in \mathbb{Z}_p^*$ such that $Q = H_1(DT)$ and $R = H_2\left(\delta_0\right)$. Using Q, R, U_m computes,

$$\sigma_3 = {g_0}^{\frac{\beta_1}{(Q+R)}},$$

where ${g_0}^{\beta_1}$ is a public component of LHP (AA_1). Then, the signed token is given by $ST = DT \parallel (\sigma_1, \sigma_2, \sigma_3)$.

To complete the delegation of access, finally U_m sends the signed token ST along with the encrypted decryption key EDK to the delegatees.

F. EHR Decryption Under Access Delegation

Now, we extend the considered scenario from the previous subsection to describe how a user who has the right to access a specific EHR_{obj} through delegation can successfully access and decrypt the intended EHR_{obj}. Consider a user U_n, who is having attributes that satisfy the substructure $\mathcal{T}''$ and has received the signed delegation token $ST = DT \parallel (\sigma_1, \sigma_2, \sigma_3)$ from U_m to access the object O. To gain access, U_n first of all sends an access request to HC with the received ST. With the reception of ST, HC determines that the sender is a delegatee and therefore forwards the signed token ST to LHP along with the EHR ciphertext components associated with the object O. The ciphertext components sent by HC to LHP are given by $E' = \left(\mathcal{T}, C'_i\right\}_{i=1,2,...,l}, \{C''_i\}_{i=1,2,...,s}\Big)$, where C'_i and C''_i are as mentioned in (4) and (6). Upon receiving ST and E' , LHP primarily has two tasks: verification of the signed token and re-encryption of the ciphertext as described below.

1) Verification of the Signed Token: LHP first checks the expiration information of the token and if it is not expired, the signature verification is carried out as follows. Note that the signature components $(\sigma_1, \sigma_2, \sigma_3)$ are as given in (13), (14), and (15). Using σ_1, σ_2 along with the public attribute components $\{T_i\}_{i=1,2,...,s}$ and the authority related public components $\{X_i\}_{i=1,2,...,l}$ associated with the attributes in $\mathcal{T}'$

(delegator's sub-structure) LHP computes the helper string δ_1 such that,

$$\begin{aligned}\delta_1 &= \prod_{i=1}^{s} e\left(sk_i^b, T_i\right) \cdot \prod_{i=1}^{l} e\left(\left(sk_0^i\right)^{u_i b}, X_i\right) \\ &= \prod_{i=1}^{s} e\left({g_0}^{\frac{r_m b}{t_i}}, {g_0}^{t_i}\right) \cdot \prod_{i=1}^{l} e\left(g_0 \frac{\left(\alpha_i - r_m\right) u_i b}{\beta_i}, {g_0}^{\beta_i}\right) \\ &= \prod_{i=1}^{l} e\left(g_0, g_0\right)^{\alpha_i b u_i}.\end{aligned}$$

Thereafter, LHP computes $Q,' R' \in \mathbb{Z}_p^*$ such that $Q' = H_1(DT)$ and $R' = H_2(\delta_1)$. LHP determines the token signature is valid, given that the condition,

$$e\left(\sigma_3, {g_0}^{Q'+R'}\right) \stackrel{?}{\rightarrow} e\left(g_0{}^{\beta_1}, g_0\right) \quad \text{isheld.}$$

2) Re-Encryption of the Ciphertext: After validating the token signature, LHP extracts the encrypted re-encryption key ERK (given in (12)) and recovers the re-encryption key RK with the help of ERK and the master secret exponent of LHP β_1. Note that the expression for RK and its components $\{RK_i\}_{i=1,2,\ldots,s}$ and $\left\{RK_0^i\right\}_{i=1,2,\ldots,l}$ are given in (10) - (11). With the help of RK and the ciphertext components $\{C_i'\}_{i=1,2,\ldots,l}$, $\{C_i''\}_{i=1,2,\ldots,s}$, LHP computes the re-encrypted cipher RC,

$$\begin{aligned}RC =& \prod_{i=1}^{s} e\left(RK_i, C_i''\right) \cdot \prod_{i=1}^{l} e\left(RK_0^i, C_i'\right) \\ &= \prod_{i=1}^{s} e\left({g_0}^{\frac{r_m}{t_i}} \cdot {g_1}^{d}, {g_0}^{t_i h_i}\right) \cdot \prod_{i=1}^{l} e\left(g_0 \frac{\alpha_i - r_m}{\beta_i} \cdot {g_1}^{d}, {g_0}^{h\beta_i}\right) \\ &= \prod_{i=1}^{l} e\left(g_0, g_0\right)^{\alpha_i h} \cdot e\left(g_0, g_1\right)^{d\beta_i h} \cdot \prod_{i=1}^{s} e\left(g_0, g_1\right)^{dt_i h_i}.\end{aligned}$$

Finally, LHP will send RC to HC to complete the re-encryption process. After receiving RC, HC will forward RC along with the ciphertext components $C_0, \{AD_i'\}_{i=1,2,\ldots,l}$,$\{AD_i''\}_{i=1,2,\ldots,s}$ to U_n.

With the reception of the aforementioned ciphertext components from HC, U_n can carry out the decryption as follows.

- First, U_n recovers $DK = {g_0}^d$ from the encrypted decryption key EDK using the attribute secret keys associated with the attributes in $\mathcal{T}''$. This decryption process is as mentioned in Section 7.4.
- Then, U_n will be able to obtain EHR data M with the help of $DK, C_0, \{AD_i'\}_{i=1,2,\ldots,l}, \{AD_i''\}_{i=1,2,\ldots,s}$ as follows.

$$\begin{aligned}M' &= \frac{C_0 \cdot \prod_{i=1}^{s} e\left(DK, AD_i{}''\right) \cdot \prod_{i=1}^{l} e\left(DK, AD_i{}'\right)}{RC} \\ &= \frac{C_0 \cdot \prod_{i=1}^{s} e\left({g_0}^d, {g_1}^{t_i h_i}\right) \cdot \prod_{i=1}^{l} e\left({g_0}^d, g_1 \beta_i h\right)}{\prod_{i=1}^{l} e\left(g_0, g_0\right)^{\alpha_i h} \cdot e\left(g_0, g_1\right)^{d\beta_i h} \cdot \prod_{i=1}^{s} e\left(g_0, g_1\right)^{dt_i h_i}} \\ &= M.\end{aligned}$$

G. Extending to Multi-Level Access Delegation

In Section 7.6, we presented how the proposed MA-CP-RE scheme enables first-level access delegation (i.e. U_m who is eligible to access the object O delegates the access to any user having attributes that satisfy the sub-structure $\mathcal{T}''$).Now, let us see, how we can enable multi-level delegation.Suppose, U_n (who received access from the first-level delegation) wants to delegate further to any user who is having attributes that satisfy the access sub-structure $\mathcal{T}'''$.U_n proceeds as follows.

- First of all, U_n encrypts the decryption key $DK = {g_0}^d$ (received from U_m) using the attributes in $\mathcal{T}'''$ and generates the encrypted decryption key EDK_1.
- Then, U_n generates a new delegation token DT_1 including the delegator's sub-structure $\mathcal{T}''$, delegatee's sub-structure $\mathcal{T}'''$, delegation permission (DP) index and the token expiration information (EXP). Note that this new token does not require to have information on the delegated EHR_{obj} and the encrypted re-encryption key due to the fact that this token is used together with the delegation token associated with the firstlevel delegation as an aggregation.
- Suppose, the signed token received by U_n from the first-level delegation is given by $ST = \{DT, [DT]_{\mathcal{T}'}\}$. The notation $[DT]_{T'}$ denotes the signature of DT made using the attributes in the sub-structure $\mathcal{T}'$. Then, U_n generates an aggregated token (AGT) using the received ST and the new delegation token DT_1 such that,

$$AGT = \{DT, [DT]_{\mathcal{T}'}\} \parallel \{DT_1, [DT \parallel DT_1]_{\mathcal{T}''}\}.$$

- Finally, U_n forwards AGT along with EDK_1 to the delegatees.

Suppose, now a user U_r (who has attributes that satisfy $\mathcal{T}'''$) wants to access the object O with the aggregated token AGT. Similar to the first-level delegation scenario, U_r should forward AGT to LHP through HC. The procedure utilized for the re-encryption (at LHP) and decryption at the user's end is as same as the first-level delegation scenario.However, the only difference is the protocol that LHP adopts to validate the aggregated token. For an aggregated token (with two delegation tokens) to be valid the following conditions must be maintained.

- The first token (DT) must not be expired, must have a DP index of 1 and the associated signature $\{DT, [DT]_{\mathcal{T}'}\}$ should be verified.
- The delegatee's sub-structure in the first token must be same as the delegator's sub-structure in the next token (DT_1).
- DT_1 should not be expired and its signature $\{DT_1\ [DT \parallel DT_1]_{T''}\}$ must be valid.

Similarly, by aggregating the tokens appropriately as mentioned above, it is possible to achieve higher-order delegations (third-level delegation and higher) effectively.

H. Revocation of Users

In this scheme, revocation of users must be handled on both attribute level and token level (to revoke delegatees).

TABLE I
FILTER DESIGN EQUATIONS ...

Order of filter	Arbitrary coefficients e_m	coefficients b_{ij}
1	$b_{ij} = \hat{e}.\hat{\beta_{ij}}$,	$b_{00} = 0$
2	$\beta_{22} = (\ 1, -1, -1,\ \ 1,\ \ 1,\ \ 1)$	
3	$b_{ij} = \hat{e}.\hat{\beta_{ij}}$,	$b_{00} = 0$,

Attribute level revocation ensures that a user will not be able to use the secret keys related to the revoked attribute in any further transactions. In the proposed scheme, the attribute level revocation is handled by the AA which is responsible for the attribute to be revoked and the procedure is same as the one used in [7]. Hence, we will not discuss the procedure in this extended version. MA-CP-RE adopts two mechanisms to revoke delegatees via revoking their associated delegation tokens. Each token includes its validity period and when issuing a delegation token, issuing entity can choose the validity period appropriately based on the delegatee. This provides revocation through token expiration.

Suppose, a delegator wants to revoke a specific token before it expires. Then, the delegator can enforce it by generating a unique token identifier using H_1 and pass it to the LHP.LHP will add the token identifier to its blacklist, which prevents delegatees from using the associated token.

VIII. SECURITY ANALYSIS

In this section, we introduce the security model in which we define two adversarial models. Thereafter, we provide evidence for the fact that the proposed MA-CP-RE scheme is secure against the introduced adversarial models.

A. Security Model

As we have presented in Section 4, we assume that the HC is semi-trusted meaning that it will follow the operational protocol while being curious on the stored data. Also, we consider that all AAs are trusted and they will only issue attribute keys to users after validating the identity as well as the eligibility of the users. In addition, it is also assumed that the aforementioned secret key issuance as well as the secret key issuance under delegation occurs after establishing secure communication channels. Thus, the secret keys will only be available to legitimate users. Furthermore, we also assume that the users might be curious on the stored data, meaning that they could potentially be interested in extracting information illegitimately via colluding with other users. To validate the security of the proposed scheme, we define two types of adversaries - Type-I adversary and Type-II adversary as mentioned below. Type-I adversary: We define the Type-I adversary as an adversary who can distinguish the ciphertexts.

Type-II adversary: Suppose, there exists two adversaries who want to collude their attribute keys and decrypt data that cannot be decrypted alone. We define such an adversary as a Type-II adversary.

In the following, we provide evidence for the fact that the proposed scheme is secure against Type-I and Type-II adversarial models. For both cases, first we provide the security proof without considering access delegation and thereafter we argue that the aforementioned properties hold under access delegation.

B. Resistance to Type-I Adversaries

To show that the proposed scheme is secure against the Type-I adversarial model, we provide evidence for the fact that the proposed MA-CP-RE scheme is IND-CPA secure (indistinguishable under chosen plaintext attacks) under the Decisional Bilinear Diffie-Hellman (DBDH) assumption.

Decisional Bilinear Diffie-Hellman (DBDH) Assumption: Suppose $\mathbb{G}_0, \mathbb{G}_1$ are cyclic groups of order p. Let g_0 be a generator of $\mathbb{G}_0$ and $e : \mathbb{G}_0 \times \mathbb{G}_0 \rightarrow \mathbb{G}_1$ be a bilinear mapping function. Given that $a, b, c, z \in \mathbb{Z}_p^*$ are randomly chosen, there is no polynomial-time adversary that can distinguish the tuple $\left({g_0}^a, {g_0}^b, {g_0}^c, e\left(g_0, g_0\right)^{abc}\right)$ from the tuple $\left({g_0}^a, {g_0}^b\ {g_0}^c, e\left(g_0, g_0\right)^z\right)$ with non-negligible probability.

1) IND-CPA Security Without Access Delegation: We define the model for IND-CPA security, with the help of the following game between a challenger C and a Type-I adversary AI , where $\mathcal{C}$ simulates the protocol and answers queries from AI .

- Init: The Type-I adversary $\mathcal{A}^I$ selects the challenge access structure $\mathcal{C}$ and gives it to the challenger $\mathcal{C}$.
- Setup: $\mathcal{C}$ generates the master secret $MK_{\mathcal{C}}$ and the public tuple $PK_{\mathcal{C}}$ and $PK_{\mathcal{C}}$ is sent to $\mathcal{A}^I$
- Phase 1: $\mathcal{A}^I$ sends attribute key requests to C for the attributes which are not elements in $\mathcal{T}^*$ and $\mathcal{C}$ responds with relevant secret keys.
- Challenge: $\mathcal{A}^I$ sends two messages M_0, M_1 to the challenger $\mathcal{C}$. $\mathcal{C}$ picks a random bit $v \in \{0, 1\}$ and generates the ciphertext E_v using the attributes in $\mathcal{T}^*$ .Then, the ciphertext E_v is forwarded to the Type-I adversary $\mathcal{A}^I$.
- Phase 2: Phase 1 is repeated
- Guess: The adversary $\mathcal{A}^I$ outputs a guess $v' \in \{0, 1\}$.

The advantage of the adversary $\mathcal{A}^I$ in the aforementioned game is defined as,

$$\epsilon = \left| \Pr\left[v' = v\right] - \frac{1}{2} \right|.$$

Theorem: If the DBDH assumption is held, the proposed MA-CP-RE scheme is secure under the IND-CPA security model with no access delegation.

Proof: To prove the theorem, we show that if an adversary $\mathcal{A}^I$ can win the IND-CPA security game with an advantage ϵ, it is possible to use the adversary $\mathcal{A}^I$ to build a simulator $\mathcal{S}$ that can break the DBDH assumption with an advantage $\epsilon/2$. tu First of all, the challenger $\mathcal{C}$ generates two cyclic groups $\mathbb{G}_0$ and $\mathbb{G}_1$ with g_0, g_1 being generators of $\mathbb{G}_0$, an efficiently computable bilinear mapping function $e : \mathbb{G}_0 \times \mathbb{G}_0 \rightarrow \mathbb{G}_1$ and a set of random exponents $a, b, c, z \in \mathbb{Z}_p^*$. Then, the challenger $\mathcal{C}$ feeds a DBDH instance $\left(g_0, {g_0}^a, {g_0}^b, {g_0}^c, R_\delta\right)$ to the simulator $\mathcal{S}$ in which R_δ is set through flipping a fair coin δ where,

$$R_\delta = \begin{cases} e\left(g_0, g_0\right)^{abc} & \text{if} \delta = 0 \\ e\left(g_0, g_0\right)^z & \text{otherwise'} \end{cases}$$

along with $e, g_1, {g_1}^b$ and ${g_1}^c$. The simulator $\mathcal{S}$ acts as the challenger for the adversary $\mathcal{A}^I$ and the simulation proceeds as follows.

Init: $\mathcal{A}^I$ selects a challenge access sub-structure $\mathcal{T}^*$ (which contains s attributes from l out of N AAs) and gives it to $\mathcal{S}$. Note that we denote the attribute set in $\mathcal{T}^*$ with AT_{T^*} .

Setup: We assume that the simulator $\mathcal{S}$ simulates onbehalf of all N AAs. For each attribute $i \in AT_{T^*}$, the simulator S chooses a random element $q_i \in \mathbb{Z}_p^*$ and thereby sets the public attribute components for each aforementioned attribute as $T_i = {g_0}^{q_i}$ and $D_i = {g_1}^{q_i}$. For all the other attributes ($(i \notin AT_{T^*})$), the simulator $\mathcal{S}$ sets $T_i = {g_0}^{b/q_i}$ and $D_i = g_1^{b/q_i}$. Furthermore, the simulator $\mathcal{S}$ selects a set of random exponents $\{\gamma_i, \beta_i\}_{i=1,2,\ldots,N} \in \mathbb{Z}_p^*$ and by allowing, $e(g_0, g_0)^{\alpha_i} = e(g_0, g_0)^{ab/l} \cdot e(g_0, g_0)^{\gamma_i}, \mathcal{S}^p$, implicitly sets each AA's secret key $\alpha_i = \frac{ab}{l} + \gamma_i$. Then, the set of public parameters are forwarded to $\mathcal{A}^I$.

Phase 1: The adversary $\mathcal{A}^I$ sends attribute key requests to the simulator $\mathcal{S}$ for a set of attributes such that each attribute $i \notin AT_{\mathcal{T}^*}$. For the adversary $\mathcal{A}^I$, the simulator $\mathcal{S}$ generates the secret key sk_0^i which relates the identity of the issuing authority and the identity of the adversary as,

$$sk_0^i = g_0^{\frac{\gamma_i - \hat{r}b}{\beta_i}} = g_0^{\frac{\alpha_i - \left(\hat{r}b + \frac{ab}{l}\right)}{\beta_i}}$$

where $\hat{r} \in \mathbb{Z}_p^*$. Then, $\mathcal{S}$ should generate the attribute secret key sk_i corresponding to each requested attribute. To have a valid simulation of attribute secret keys, sk_i must be in the form,

$$sk_i = g_0 \frac{(\hat{r}b + aq)q_i}{b}$$

due to the fact that the simulator $\mathcal{S}$ sets the secret exponent of any attribute $i \notin AT_{\mathcal{T}^*}$ as b/q_i. Hence, $\mathcal{S}$ sets $sk_i = {g_0}^{\hat{r}} q_i \cdot {g_0}^{\frac{aq_i}{l}}$. It is evident that this is a valid simulation of secret keys, since, $sk_i = g_0 \frac{\left(\hat{r}b + \frac{ab}{l}\right)q_i}{b} = {g_0}^{\hat{r}q_i} \cdot {g_0}^{\frac{aq_i}{l}}$. Then, $\mathcal{S}$ sends the secret keys $\left(sk_0^i, sk_i\right)$ for each attribute $i \notin AT_{\mathcal{T}^*}$.

Challenge phase: $\mathcal{A}^I$ sends two plaintexts$M_0, M_1 \in \mathbb{G}_1$ to $\mathcal{S}$. Then, $\mathcal{S}$ will first encrypt one of M_0, M_1 according to $\mathcal{T}^*$ by flipping a fair binary coin v. To encrypt M_v, the simulator $\mathcal{S}$ first computes C_0 and $\{C_i,' AD_i'\}_{i=1,2,\ldots,l}$ such that,

$$C_0 = M_v \prod_{i=1}^{l} Z_i^c = M_v \cdot e(g_0, g_0)^{\sum_{i=1}^{l} (\alpha_i)c}$$

$$= M_v \cdot e(g_0, g_0)^{\sum_{i=1}^{l} \left(\frac{ab}{l} + \gamma_i\right)c} = M_v \cdot R_\delta \cdot e(g_0, g_0)^{\sum_{i=1}^{l} (\gamma_i)c}$$

$$C_i' = {g_0}^{\beta_i c} \text{ and} AD_i' = g_1 \beta_i c$$

Then, for each attribute in $AT_{\mathcal{T}^*}$ except the last, a random exponent $h_i \in \mathbb{Z}_p^*$ is chosen and assigns ${g_0}^{h_i}$, while the last element is assigned the value equals to $g_0 l_c / \prod_{i=1}^{s-1} g_0 h_i$. Thereafter, $\mathcal{S}$ computes $\left\{C_i'' AD_i''\right\}_{i=1,2,\ldots,s}$ such that, $C_i'' = g_0^{q_i h_i}$ and $AD_i'' = g_1^{q_i h_i}$. Thus, the ciphertext of M_v is given by,

$$E_v = \left(\mathcal{T}^*, C_0, \{C_i,' AD_i'\}_{i=1,2,\ldots,l}, \{C_i,'' AD_i''\}_{i=1,2,\ldots,s}\right)$$

To complete the challenge phase, the simulator S sends the ciphertext E_v of the message M_v to the adversary $\mathcal{A}^I$.

Guess: The adversary $\mathcal{A}^I$ submits a guess $v' \in \{0, 1\}$. If $v' = v$, the simulator $\mathcal{S}$ will guess that $\delta' = 0$ and outputs a 0 indicating that $R_\delta = e(g_0, g_0)^{abc}$. If $v' \neq v$, the simulator $\mathcal{S}$ will guess that $\delta' = 1$ and outputs a 1 indicating that $R_\delta = e(g_0, g_0)^z$.

In the case where $\delta = 0 \quad \left(R_\delta = e(g_0, g_0)^{abc}\right)$, the adversary $\mathcal{A}^I$ sees a valid encryption of M_v. Therefore, $\mathcal{A}^I$ has an advantage of ϵ in winning the game. Hence, according to (17), $\Pr\left[v' = v \mid R_\delta = e(g_0, g_0)^{abc}\right] = 1/2 + \epsilon$: Since, the simulator $\mathcal{S}$ guesses that $\delta' = 0$ when $v' = v$ we have,

$$\Pr[\delta' = \delta \mid \delta = 0] = 1/2 + \epsilon.$$

When $\delta = 1 \left(R_\delta = e(g_0, g_0)^z\right)$, the adversary $\mathcal{A}^I$ will not have any advantage in the game, since $\mathcal{A}^I$ does not gain any information on M_v. Therefore, $\Pr[v' \neq v \mid R_\delta = e(g_0, g_0)^z] = \frac{1}{2}$. Given that the simulator $\mathcal{S}$ guesses $\delta' = 1$ when $v' \neq v$ it is evident that,

$$\Pr[\delta' = \delta \mid \delta = 1] = 1/2.$$

Therefore, according to (18) and (19), the total advantage of the simulator $\mathcal{S}$ to solve the DBDH problem is given by,

$$\frac{1}{2}\Pr[\delta' = \delta \mid \delta = 0] + \frac{1}{2}\Pr[\delta' = \delta \mid \delta = 1] - \frac{1}{2} = \frac{\epsilon}{2}.$$

2) IND-CPA Security Under Access Delegation: We argue the IND-CPA security of the proposed MA-CPRE scheme when subjected to access delegation as follows.Suppose, a message M is encrypted with a set of attributes associated with an access sub-structure $\mathcal{T}'$ which is having s attributes belonging to l AAs. Then, the encryption of M is given by,

$$E' = \left(\mathcal{T},' C_0, \{C_i,' AD_i'\}_{i=1,2,\ldots,l}, \{C_i,'' AD_i''\}_{i=1,2,\ldots,s}\right)$$

where $C_0, C_i', AD_i', C_i'', AD_i''$ are as given in (3) - (7). Now, assume that E' is re-encrypted to ensure that the message M can be accessed by a delegatee having the set of attributes that satisfies the delegatee access sub-structure $\mathcal{T}''$. According to Section 7.6.2, if the re-encrypted ciphertext is given by E'', then

$$E'' = \left(C_0, \{AD_i'\}_{i=1,2,\ldots,l}, \{AD_i''\}_{i=1,2,\ldots,s}, RC\right)$$

Note that the ciphertext component RC is given in (16). Furthermore, if we examine RC, it is evident that it is independent of the message M and C_0 is the only message dependent ciphertext component in E'' as well as in E' . Therefore, if the ciphertext E' is indistinguishable under chosen plaintext attacks (which was proved in Section 8.2.1) then E'' will also be indistinguishable under chosen plaintext attacks. Hence, it is evident that the proposed MA-CP-RE scheme is exhibiting IND-CPA security under access delegation.

C. Resistance Against Type-II Adversaries

In this section, we show that the proposed MA-CP-RE scheme is secure against Type-II adversaries under the Decisional Diffie-Hellman (DDH) assumption.

Decisional Diffie-Hellman (DDH) Assumption: Suppose $\mathbb{G}_0$ is a cyclic group of order p with g_0 being the generator.Given that $\beta_1, \beta_2, z \in \mathbb{Z}_p^*$ are randomly chosen, there is

no polynomial-time adversary that can distinguish the tuple $\left(g_0{}^{\beta_1}, g_0{}^{\beta_2}, g_0{}^{\beta_1 \cdot \beta_2}\right)$ from the tuple $\left(g_0{}^{\beta_1}, g_0{}^{\beta_2}, g_0{}^{2}\right)$ with nonnegligible probability.

1) Resistance Against Type-II Adversaries Without Access Delegation: Let us assume that there exists two users - U_1 and U_2 who owns the attribute sets A_1 and A_2 obtained from a specific $\mathcal{A}^{II}$. Suppose, U_1 and U_2 want to collectively decrypt information encrypted with attributes from A_1 and A_2.

Hence, this will correspond to a Type-II adversarial instance and from here on in, we denote it as a Type-II adversary AII .

To prove that the proposed MA-CP-RE scheme is secure against the Type-II adversary model, first of all, we assume that a Type-II adversary $\mathcal{A}^{II}$ has an advantage of $\epsilon(\epsilon > 0)$ for being successful in decrypting data with colluding attribute keys (i.$\epsilon = \mid \Pr - 1/2 \mid$, where $\Pr[collusion]$ denotes the adversary's probability of successfully executing an attribute collusion attack). Then, we define the adversary model for Type-II adversaries with the help of a game among a challenger $\mathcal{C}$, a simulator $\mathcal{S}$ and the Type-II adversary $\mathcal{A}^{II}$. With this game, we claim that, if the adversary $\mathcal{A}^{II}$ has an advantage of being successful with a collusion attack, then $\mathcal{A}^{II}$ has an advantage of ϵ in winning the game.

Finally, we show that, if such an adversary exists, it is possible to use this adversary to build a simulator $\mathcal{S}$ that can solve the DDH hardness problem in $\mathbb{G}_0$ with an advantage $\epsilon/2$.

The game among the challenger $\mathcal{C}$, simulator $\mathcal{S}$ and the adversary $\mathcal{A}^{II}$ runs as follows.

- Init: $\mathcal{C}$ advertises the attribute sets A_1 and A_2.
- Setup: $\mathcal{C}$ generates the master secret $MK_{\mathcal{C}}$ and the public tuple $PK_{\mathcal{C}}$.
- Phase 1: $\mathcal{A}^{II}$ sends attribute key requests for attributes in A_1 and A_2 with the user identity r_1 and r_2 respectively. $\mathcal{C}$ responds with relevant secret keys.
- Phase 2: The challenger $\mathcal{C}$ feeds the public tuple $PK_{\mathcal{C}}$ to the simulator $\mathcal{S}$ and let $\mathcal{S}$ interact with the adversary $\mathcal{A}^{II}$ for the remainder of the game.
- Challenge: $\mathcal{A}^{II}$ decides on a challenge access structure $\mathcal{T}$ such that $\mathcal{T} = (\omega_1 \wedge \omega_2)$ where $\omega_1 \in A_1$ and $\omega_2 \in A_2$ and sends it along with two messages M_0, M_1 to the simulator $\mathcal{S}$. $\mathcal{S}$ picks a random bit $v \in \{0, 1\}$ and generates the ciphertext E_v using the attributes in $\mathcal{T}$. Then, the ciphertext E_v is forwarded to the adversary $\mathcal{A}^{II}$.
- Guess: The adversary $\mathcal{A}^{II}$ outputs a guess $v' \in \{0, 1\}$.
- Phase 3: Game is repeated.

Given that the secret keys for ω_1 and ω_2 are constructed with different user identity parameters (i.e. to simulate a collusion instance), the adversary $\mathcal{A}^{II}$ must be successful in carrying out a collusion attack to submit the correct guess of $v' = v$. We assumed that $\mathcal{A}^{II}$ has an advantage of ϵ in succeeding with a collusion attack, then the advantage of the adversary $\mathcal{A}^{II}$ in the aforementioned game is defined as,

$$\epsilon = \left| \Pr[v' = v] - \frac{1}{2} \right|.$$

Theorem: If the DDH assumption is held in $\mathbb{G}_0$, the proposed MA-CP-RE scheme is secure against Type-II adversaries with no access delegation.

Proof: We show that, if there exists a Type-II adversary $\mathcal{A}^{II}$ who can win the above-mentioned game with an advantage ϵ, it is possible to use this adversary $\mathcal{A}^{II}$ to build a simulator $\mathcal{S}$ that can solve the DDH hardness problem in $\mathbb{G}_0$ with an advantage of $\epsilon/2$. The simulation proceeds as follows. Init: The challenger $\mathcal{C}$ acts as an AA and advertises the attribute sets A_1 and A_2.

Setup: The challenger $\mathcal{C}$ generates two cyclic groups $\mathbb{G}_0$ and $\mathbb{G}_1$ with g_0, g_1 being generators of $\mathbb{G}_0$ and an efficiently computable bilinear mapping function $e : \mathbb{G}_0 \times \mathbb{G}_0 \rightarrow \mathbb{G}_1$.

Also, $\mathcal{C}$ generates a set of random exponents $\alpha_{\mathcal{C}}, \beta_1, \beta_2, z \in \mathbb{Z}_p^*$. Then, the challenger $\mathcal{C}$ generates a DDH instance $\left(g_0, g_0^{\beta_1}, g_0{}^{\beta_2}, R_\delta\right)$ in which R_δ is set through flipping a fair coin δ where,

$$R_\delta = \begin{cases} g_0^{\beta_1 \cdot \beta_2} & \text{if} \delta = 0 \\ g_0^{z} & \text{otherwise.} \end{cases}$$

In addition, for each attribute index $i \in A_1$ and $i \in A_2$, random secret elements $t_{1,i} \in \mathbb{Z}_p^*, t_{2,i} \in \mathbb{Z}_p^*$ are selected and thereby sets the respective public attribute components $T_{1,i} = g_0{}_{1,i}$ and $T_{2,i} = g_0{}^{t_{2,i}}$. Then, $\mathcal{C}$ computes the master secret set $MK_{\mathcal{C}} = \left\{\alpha_{\mathcal{C}}, \beta_{\mathcal{C}} = \beta_1 \cdot \beta_2, \{t_{1,i}\}_{i=1.2} ... \middle| A, \{t_{2,i}\}_{i=1.2} ... \middle| A_2\right\}$and derives the public tuple $PK_{\mathcal{C}} = \left\{X_{\mathcal{C}} = R_\delta, Y_{\mathcal{C}} = g_1{}^{\beta_{\mathcal{C}}}, Z_{\mathcal{C}} = e\left(g_0, g_0\right)^{\alpha} \mathcal{C}, \{T_{1,i}\}_{i=1,2,...,|A_1|}, \{T_{2,i}\}_{i=1,2}\right.$

Phase 1: The adversary $\mathcal{A}^{II}$ sends attribute key requests to the challenger $\mathcal{C}$ for the attributes in A_1 and A_2. For the i^{th} attribute in A_1 the relevant secret keys are given by $\left(sk_0^1, sk_i\right)$ where,

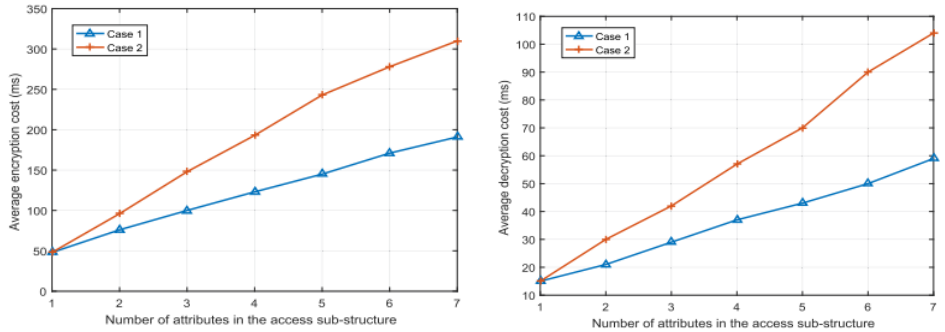
$$sk_0^1 = g_0^{\frac{\alpha_{\mathcal{C}} - r_1}{\beta_{\mathcal{C}}}} \text{ and} sk_i = g_0^{\frac{r_1}{t_{1,i}}}.$$

Similarly, the attribute keys corresponding to the i^{th} attribute in A_2 are given by, $sk_0^2 = g_0 \frac{\alpha_{\mathcal{C}} - r_2}{\beta_{\mathcal{C}}}$ and $sk_i = g_0^{\frac{r_2}{t_{2,i}}}$. Note that, r_1, r_2 denote the relevant user identifiers for the issued attributes in A_1 and A_2 respectively.

Phase 2: The challenger $\mathcal{C}$ feeds the DDH instance $\left(g_0, g_0{}^{\beta_1}, g_0{}^{\beta_2}, R_\delta\right)$ along with the public tuple $PK_{\mathcal{C}}$ to the simulator $\mathcal{S}$ and let $\mathcal{S}$ be the challenger for the adversary $\mathcal{T}^{II}$ in the remainder of the game. Note that the challenger $\mathcal{C}$ does not feed the bilinear mapping function to $\mathcal{S}$ given that the objective of the simulator is to solve the DDH problem in G_0 with the help of the adversary $\mathcal{A}^{II}$.

Challenge Phase: $\mathcal{A}^{II}$ sends a challenge access structure $\mathcal{T}$ with two attributes (ω_1, ω_2) such that $\omega_1 \in A_1$ and $\omega_2 \in A_2$ along with two plaintext messages M_0, M_1. The simulator $\mathcal{S}$, first checks whether $\omega_1 \in A_1$ and $\omega_1 \in A_2$ and if so, it flips a fair binary coin v and thereby encrypt one of M_0, M_1. To encrypt $M_v, \mathcal{S}$ first generates a random exponent $h \in \mathbb{Z}_p^*$ and computes C_0, and $\{C', AD'\}$ such that,

$$C_0 = M_v \cdot Z_{\mathcal{C}}^h = M_v \cdot e\left(g_0, g_0\right)^{\alpha_{\mathcal{C}} \cdot h}$$
$$C' = X_{\mathcal{C}}{}^h = R_\delta{}^h$$
$$AD' = Y_{\mathcal{C}}{}^h = g_1{}^{\beta_{\mathcal{C}} \cdot h}.$$



(a) Encryption cost with the number of attributes in an access sub-structure (b) Decryption cost with the number of attributes in an access sub-structure

Fig. 7. Variation of encryption and decryption (with no delegation) computational cost.

Thereafter, $\mathcal{S}$ assigns a random exponent $h_1 \in \mathbb{Z}_p^*$ for the attribute ω_1 and sets $h_2 = (h - h_1)$ for the attribute ω_2 and computes $\{C_i'', AD_i''\}_{i=1,2}$ such that,

$$C_i'' = g_0^{t_i \cdot h_i} \text{ and} AD_i'' = g_1{}^{t_i \cdot h_i}$$

in which t_i denotes the secret attribute exponent related to the i^{th} attribute in $\mathcal{T}$. Thus, the ciphertext of M_v is given by,

$$E_v = \left(\mathcal{T}, C_0, C, AD, \{C_i,'' AD_i''\}_{i=1,2}\right).$$

To complete the challenge phase, the simulator $\mathcal{S}$ sends the ciphertext E_v along with the public tuple $PK_{\mathcal{C}}$ to the adversary $\mathcal{A}^{II}$.

Guess: The adversary $\mathcal{A}^{II}$ submits a guess $v' \in \{0, 1\}$. If $v' = v$, the simulator $\mathcal{S}$ will guess that $\delta' = 0$ and outputs a 0 indicating that $R_\delta = g_0{}^{\beta_1 \cdot \beta_2}$. Otherwise, the simulator $\mathcal{S}$ will guess that $\delta' = 1$ and outputs a 1 indicating $R_\delta = g_0{}^z$.

In the case where $\delta = 0$ $\left(R_\delta = g_0{}^{\beta_1 \cdot \beta_2}\right)$, the adversary $\mathcal{A}^{II}$ gets a valid encryption of M_v. Then, according to (20), the adversary $\mathcal{A}^{II}$ has an advantage of ϵ in winning the game.

Thus, $\Pr\left[v' = v \mid R_\delta = g_0{}^{\beta_1 \cdot \beta_2}\right] = 1/2 + \epsilon$. Since, the simulator $\mathcal{S}$ guesses that $\delta' = 0$ when $v' = v$ we have,

$$\Pr\left[\delta' = \delta \mid \delta = 0\right] = 1/2 + \epsilon.$$

When $\delta = 1, (R_\delta = g_0{}^z)$, the adversary $\mathcal{A}^{II}$ will not have any advantage in the game, since R_δ embedded in $PK_{\mathcal{C}}$ is just a random value (i.e. R_δ does not associate with the master secret component $\beta_{\mathcal{C}} = \beta_1 \cdot \beta_2$). Therefore, $\Pr\left[v' \neq v \mid R_\delta = g_0{}^z\right] = \frac{1}{2}$. Given that the simulator $\mathcal{S}$ guesses $\delta' = 1$ when $v' \neq v$ it is evident that,

$$\Pr\left[\delta' = \delta \mid \delta = 1\right] = 1/2.$$

Therefore, according to (21) and (22), the total advantage of the simulator $\mathcal{S}$ to solve the DDH problem in $\mathbb{G}_0$ is given by,

$$\frac{1}{2}\Pr\left[\delta' = \delta \mid \delta = 0\right] + \frac{1}{2}\Pr\left[\delta' = \delta \mid \delta = 1\right] - \frac{1}{2} = \frac{\epsilon}{2}.$$

This proves that, if there exists a Type-II adversary who is capable of successfully mounting an attribute collusion attack with an advantage ϵ, it is possible to use this adversary to build a simulator $\mathcal{S}$ that can solve the DDH hardness problem in $\mathbb{G}_0$ with an advantage of $\epsilon/2$. Hence, if the DDH assumption holds in $\mathbb{G}_0$, the proposed MA-CP-RE scheme is secure against the Type-II adversarial model.

2) Resistance Against Type-II Adversaries Under Access Delegation: We argue the security against Type-II adversaries under access delegation using the results in Section 8.3.1 as follows.

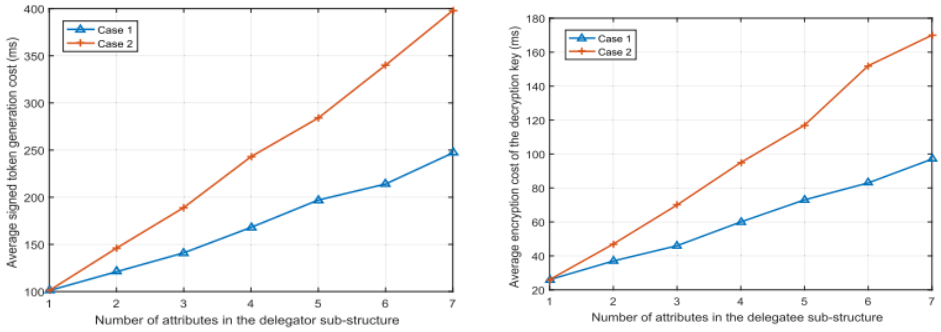
Let us assume that the encryption of a message M with an access sub-structure $\mathcal{T}'$ is given by E' . Then, a user who is eligible to decrypt E' (with his attributes) re-encrypts the ciphertext E' , so that the message M can be recovered by any user having both attributes ω_1 and ω_2. Now, assume that there exist two users U_1 and U_2 such that U_1 owns the attribute ω_1 while the user U_2 owns the attribute ω_2. Given that either of the users are not entitled to decrypt the reencrypted ciphertext, the aforementioned instance reflects a Type-II adversarial instance under access delegation. We use this instance to show that the proposed scheme can resist attacks mounted by Type-II adversaries.

As evident from the Section 7.5, to allow the re-encryption of E' to be decrypted by any user having both attributes $(\omega_1 \wedge \omega_2)$, the delegating user generates the decryption key DK and encrypts it with the attributes $(\omega_1 \wedge \omega_2)$ using the MA-CP-RE encryption mechanism. This allows any user who is having both attributes to decrypt the re-encrypted ciphertext with the help of the decryption key DK. In Section 8.3.1, we showed that the MA-CP-RE encryption mechanism is resistant against collusion attacks under the DDH assumption. Therefore, if the DDH assumption holds in $\mathbb{G}_0$, it is infeasible to recover the decryption key DK via attribute collusion. Thus, it is evident that the proposed MA-CP-RE scheme exhibits security against the Type-II adversaries under access delegation.

IX. PERFORMANCE EVALUATION

In this section, we intend to provide evidence for the performance of the proposed MA-CP-RE scheme with respect to the associated computational cost using simulation results. We carried out the simulations on a Core i5, 2.5 GHz PC with 8 GB of RAM. Furthermore, the elliptic curve $y^2 = x^3 + x$ over a 512-bit finite field having a group order of 160 bits was used to generate the required cyclic groups.

This parameter setting was chosen for the simulations since it can generate keys having the equivalence security of 1024-bit RSA keys [38].



(a) Signed token generation cost with the number of attributes in the delegator 's sub-structure

(b) Encryption cost of the decryption key with number of attributes in the delegatee' s sub-structure

Fig. 8. Variation of the delegator's computational cost

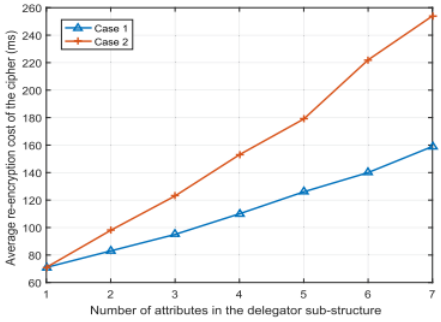


Fig. 9. Variation of the re-encryption cost at LHP

For the analysis, we considered a simple multi-authority environment with 7 AAs each managing 7 attributes. With the help of this simulation environment, we carried out simulations to determine the behavior of associated computational cost for the 5 processes in the scheme: encrypting the data (by LHP before being outsourced to HC), decryption with no delegation, access delegation (delegator's computation cost), re-encryption of the ciphertext (by LHP) and decryption of delegated data (delegatee's computation cost) with the number of attributes in a given access sub-structure under the following two cases.

- Case 1: All the attributes in the sub-structure belong to the same AA.
- Case 2: Each attribute in the sub-structure belongs to a different AA.

The above mentioned two conditions were considered, due to the fact that Case 1 has the lowest computational complexity whereas Case 2 corresponds for the maximum computational complexity and all the other possibilities will lie in-between Case 1 and Case 2.

Fig. 7a shows the variation of encryption cost in relation to the two above stated cases while Fig. 7b represents the variation of decryption cost (with no delegation). Note that we have only considered the encryption cost with respect to one sub-structure and an access policy can have several sub-structures. Fig. 8 depicts the variation of the computational cost associated with access delegation. During a delegation, the delegator needs to generate the signed delegation token (ST) and the encrypted decryption key using the attributes in the delegatee's sub-structure. Fig. 8a shows the variation of the computational cost associated with the generation of the signed token with the number of attributes in the delegator's sub-structure whereas Fig. 8b represents the cost associated with the generation of the encrypted decryption key with the number of attributes in the delegatee's sub-structure. Hence, the total computation cost of a delegation depends upon both the number of attributes in the delegator's sub-structure as well as the delegatee's sub-structure.

Fig. 9 shows the computational cost associated with the re-encryption process of a delegated ciphertext at LHP with the number of attributes in the delegator's sub-structure.

Finally, Fig. 10 depicts the computational cost associated with the decryption of a delegated ciphertext. The decryption of a delegated ciphertext has two processes: recovery of the decryption key using the attribute secret keys associated with the delegatee's sub-structure and then use the decryption key to decrypt the re-encrypted ciphertext to recover the message. The variation of the computational cost associated with the first process with the number of attributes in the delegatee's sub-structure is given in Fig. 10a whereas the computational cost associated with the second process with the number of attributes in the delegator's sub-structure is given in Fig. 10b. Note that, in the simulations, we restricted the number of attributes in a sub-structure to a maximum of 7. We believe that this is a realistic assumption since we may not come across access sub-structures having more than 7 attributes in practice, especially considering our application of interest. However, the simulation results provide evidence for the fact that the variation of the computational cost associated with all the processes of the MA-CP-RE scheme exhibit nearly linear characteristics. Therefore, it is fair to conclude that the proposed scheme will also function efficiently and effectively under the access sub-structures with a substantially larger number of attributes as well.

X. DISCUSSION

In this section, we compare the end-user computational complexity as well as delegatability characteristics of the proposed scheme with existing solutions. We have evaluated

Algorithm 1 Framework of ensemble learning for our system.

Input: The set of positive samples for current batch, P_n; The set of unlabelled samples for current batch, U_n; Ensemble of classifiers on former batches, E_{n-1};

Output: Ensemble of classifiers on the current batch, E_n;

1: Extracting the set of reliable negative and/or positive samples T_n from U_n with help of P_n;
2: Training ensemble of classifiers E on $T_n \cup P_n$, with help of data in former batches;
3: $E_n = E_{n-1} cup E$;
4: Classifying samples in $U_n - T_n$ by E_n;
5: Deleting some weak classifiers in E_n so as to keep the capacity of E_n;
6: **return** E_n;

the end-user computational complexity based on the exponentiation and pairing count for the three processes: decryption without delegation, access delegation and decryption under delegation at the user's end.

In Table 1, end-user computational complexity and the delegation capability of the proposed MA-CP-RE scheme are compared with the most relevant schemes in literature. The notations $|A_1|, |A_2|$ and $|A_A|$ denote the number of attributes in the delegator's sub-structure, delegatee's sub-structure and the number of attributes managed by the centralized AA.

Note that, all existing solutions have a centralized AA, whereas our scheme supports multiple AAs, hence to make the comparison feasible, we set the number of AAs to 1 in the analysis. We have used $N_{\exp}, N_e$ to denote the number of exponentiations and the number of pairing operations.

In [37], it is necessary for a user to have secret keys for all attributes in A_A (negative secret keys for attributes that do not belong to the user). Hence, when $|A_A| >> |A_1|$ the scheme in [37] does not scale and will not function efficiently. It is also observable that the scheme in [28] has a slightly lower end-user computational complexity compared to our scheme, but it uses a third-party mediator to assist the delegation which affects the privacy of users as explained in Section 2. Furthermore, our scheme is capable of providing better control over delegation while facilitating delegation of access to a subset of data that can be accessed with the delegated attributes (selective delegatability), a characteristic which is not available in other schemes.

XI. CONCLUSION

In this paper, we extended the multi-authority CP-ABE scheme (MA-CP-ABE) in [7], to facilitate flexible, controlled access delegation on data outsourced to a third-party cloud platform. With the help of the proposed access control mechanism (denoted as MA-CP-RE), we have presented how it can be applied to an e-health environment to enforce secure sharing of EHRs of patients among EHR users when the patient EHRs are stored in a healthcare cloud. The proposed MA-CP-RE scheme has two novel properties. The scheme is capable of provisioning multi-level access delegation in which subsequent delegation of access by a delegatee is only allowed if the corresponding delegator has given the permission to do so. Furthermore, a user is allowed to selectively delegate data, meaning that delegatees will not be able to access other resources which are associated with the same access sub-structure as the resource for which the access is granted. We have also shown that the proposed MA-CP-RE scheme is IND-CPA secure and resistant against attacks mounted via attribute collusion as well as providing superior delegating capabilities compared to similar existing solutions.

XII. REFERENCES SECTION

You can use a bibliography generated by BibTeX as a .bbl file. BibTeX documentation can be easily obtained at: http://mirror.ctan.org/biblio/bibtex/contrib/doc/ The IEEEtran BibTeX style support page is: http://www.michaelshell.org/tex/ieeetran/bibtex/

XIII. SIMPLE REFERENCES

You can manually copy in the resultant .bbl file and set second argument of `\begin` to the number of references (used to reserve space for the reference number labels box).

REFERENCES

[1] S. Bleikertz, M. Schunter, C. W. Probst, D. Pendarakis, and K. Eriksson, "Security audits of multi-tier virtual infrastructures in public infrastructure clouds," in Proc. *ACM Workshop Cloud Comput. Secur.*, 2010, pp. 93–102.

[2] B. Grobauer, T. Walloschek, and E. Stocker, "Understanding cloud computing vulnerabilities," *IEEE Secur. Privacy,* vol. 9, no. 2, pp. 50–57, Mar./Apr. 2011.

[3] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, "Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 1, pp. 131–143, Jan. 2013.

[4] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in Proc. I*EEE Symp. Secur. Privacy,* 2007, pp. 321–334.

[5] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. 13th ACM Conf. Comput. Commun. Secur.*, 2006, pp. 89–98.

[6] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *Proc. Int.Workshop Public Key Cryptogr.*, 2011, pp. 53–70.

[7] H. S. G. Pussewalage and V. A. Oleshchuk, "A distributed multiauthority attribute based encryption scheme for secure sharing of personal health records," in *Proc. 22nd ACM Symp. Access Control Models Technol.*, 2017, pp. 255–262.

[8] S. S. Chow, "A framework of multi-authority attribute-based encryption with outsourcing and revocation," in *Proc. 21st ACM Symp. Access Control Models Technol.*, 2016, pp. 215–226.

[9] S. S. Chow, "A framework of multi-authority attribute-based encryption with outsourcing and revocation," in *Proc. 21st ACM Symp. Access Control Models Technol.*, 2016, pp. 215–226.

[10] Z. Zhou, D. Huang, and Z. Wang, "Efficient privacy-preserving ciphertext-policy attribute based-encryption and broadcast encryption," *IEEE Trans. Comput.*, vol. 64, pp. 126–138, 2015.

[11] S. Wang, J. Zhou, J. K. Liu, J. Yu, J. Chen, and W. Xie, "An efficient file hierarchy attribute-based encryption scheme in cloud computing," *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 6, pp. 1265–1277, Jun. 2016.

[12] J. Han, W. Susilo, Y. Mu, J. Zhou, and M. H. A. Au, "Improving privacy and security in decentralized ciphertext-policy attributebased encryption," *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 3, pp. 665–678, Mar. 2015.

[13] S. Lin, R. Zhang, H. Ma, and M. Wang, "Revisiting attribute-based encryption with verifiable outsourced decryption," *IEEE Trans.Inf. Forensics Secur.*, vol. 10, no. 10, pp. 2119–2130, Oct. 2015.

[14] Y. Yang, X. Chen, H. Chen, and X. Du, "Improving privacy and security in decentralizing multi-authority attribute-based encryption in cloud computing," *IEEE Access, vol.* 6, pp. 18 009–18 021, 2018.

[15] H. S. G. Pussewalage and V. A. Oleshchuk, "An efficient multi-show unlinkable attribute based credential scheme for a collaborative e-health environment," in *Proc. 3rd IEEE Int. Conf. Collaboration Internet Comput.*, 2017, pp. 421–428
[16] H. S. G. Pussewalage and V. A. Oleshchuk, "An anonymous delegatable attribute based credential scheme for a collaborative ehealth environment," *ACM Trans. Internet Technol.*, vol. 19, no. 3, 2019, Art. no. 41.
[17] H. S. G. Pussewalage and V. A. Oleshchuk, "Attribute based access control scheme with controlled access delegation for collaborative ehealth environments," *J. Inf. Secur. Appl.*, vol. 37, pp. 50–64, 2017.
[18] H. S. G. Pussewalage and V. A. Oleshchuk, "Blockchain based delegatable access control scheme for a collaborative e-health environment," in *Proc. 1st IEEE Int. Conf.* Blockchain, 2018, pp. 1204–1211.
[19] L. Ibraimi, M. Asim, and M. Petkovic, "Secure management of personal health records by applying attribute-based encryption," in *Proc. 6th Int. Workshop Wearable Micro Nano Technol. Personalized Health,* 2009, pp. 71–74.
[20] L. Ibraimi, Q. Tang, P. Hartel, and W. Jonker, "Efficient and provable secure ciphertext-policy attribute-based encryption schemes," in *Proc. Int. Conf. Inf. Secur. Pract. Experience,* 2009, pp. 1–12.
[21] L. Ibraimi, Q. Tang, P. Hartel, and W. Jonker, "Efficient and provable secure ciphertext-policy attribute-based encryption schemes," in *Proc. Int. Conf. Inf. Secur. Pract. Experience,* 2009, pp. 1–12.
[22] M. Barua, X. Liang, R. Lu, and X. Shen, "PEACE: An efficient and secure patient-centric access control scheme for eHealth care system," in *Proc. IEEE Conf. Comput. Commun. Workshops,* 2011, pp. 970–975.
[23] C. Danwei, C. Linling, F. Xiaowei, H. Liwen, P. Su, and H. Ruoxiang, "Securing patient-centric personal health records sharing system in cloud computing," *China Commun.*, vol. 11, no. 13, pp. 121–127, 2014.
[24] C.-J. Wang, X.-L. Xu, D.-Y. Shi, and W.-L. Lin, "An efficient cloudbased personal health records system using attribute-based encryption and anonymous multi-receiver identity-based encryption," in *Proc. 9th Int. Conf. P2P Parallel Grid Cloud Internet Comput.*, 2014, pp. 74–81.
[25] M. Chase and S. S. M. Chow, "Improving privacy and security in multi-authority attribute-based encryption," in *Proc. 16th ACM Conf. Comput. Commun. Secur.*, 2009, pp. 121–130.
[26] P. Datta, I. Komargodski, and B. Waters, "Decentralized multiauthority ABE for DNFs from LWE," in *Proc. Adv. Cryptol.*, 2021, pp. 177–209.
[27] S. Ding, Y. Zhao, and H. Zhu, "Extending fuzzy identity-based encryption with delegating capabilities," in *Proc. 6th IEEE Joint Int. Inf. Technol. Artif. Intell. Conf.*, 2011, pp. 19–23.
[28] L. Ibraimi, M. Petkovic, S. Nikova, P. Hartel, and W. Jonker, "Ciphertext-policy attribute-based threshold decryption with flexible delegation and revocation of user attributes," Centre for Telematics and Inf. Technol., Univ. Twente, Enschede, Netherlands, Nov. 2009. [Online]. Available: https://ris.utwente.nl/ws/portalfiles/portal/5098396/Technical_Report.pdf
[29] W. Yin *et al*, "Delegation of decryption rights with revocability from learning with errors," *IEEE Access,* vol. 6, pp. 61 163–61 175, 2018
[30] P. Zeng and K.-K. R. Choo, "A new kind of conditional proxy reencryption for secure cloud storage," *IEEE Access,* vol. 6, pp
[31] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," *ACM Trans. Inf. Syst. Secur.*, vol. 9, no. 1, pp. 1–30, 2006.
[32] C.-K. Chu and W.-G. Tzeng, "Identity-based proxy re-encryption without random oracles," in *Int. Conf. Inf. Secur.*, 2007, pp. 189–202.
[33] M. Green and G. Ateniese, "Identity-based proxy re-encryption," in *Proc. Int. Conf. Appl. Cryptogr. Netw. Secur.*, 2007, pp. 288–306.
[34] T. Matsuo, "Proxy re-encryption systems for identity-based encryption," in *Proc. Int. Conf. Pairing-Based Cryptogr.*, 2007, pp. 247–267.
[35] S. Guo, Y. Zeng, J. Wei, and Q. Xu, "Attribute-based re-encryption scheme in the standard model," *Wuhan Univ. J. Natural Sci.*, vol. 13, no. 5, pp. 621–625, 2008.
[36] X. Liang, Z. Cao, H. Lin, and J. Shao, "Attribute based proxy reencryption with delegating capabilities," in *Proc. 4th ACM Int. Symp. Inf. Comput. Commun. Secur.*, 2009, pp. 276–286.
[37] S. Luo, J. Hu, and Z. Chen, "Ciphertext policy attribute-based proxy re-encryption," in *Proc. Int. Conf. Inf. Commun. Secur.*, 2010, pp. 401–415.
[38] E. Barker, "Recommendation for key management part 1: General (revision 4)," *NIST Special Pub.*, vol. 800–57, pp. 1–160, Jan. 2016.

Harsha Sandaruwan Gardiyawasam Pussewalage (Member, IEEE) received the BSc (Hons.) in engineering degree from the University of Ruhuna, Sri, in 2010, and the MSc and PhD degrees in information and communication technology (ICT) from the University of Agder (UiA), Norway, in 2013 and 2019, respectively. He is currently employed as an Associate Professor with the Department of ICT, UiA. His current research interests include access control models, security protocols, privacy-preserving protocols, and blockchain systems.

Vladimir Oleshchuk (Senior Member, IEEE) received the PhD degree in computer science from the Taras Shevchenko Kyiv National University, Kyiv, Ukraine, in 1988, where he worked as an Associate Professor before joining the University of Agder (UiA), Norway, 1992. He is currently a professor with the Department of Information and Communication Technology, UiA. He is a senior member of the ACM. His current research interests include access control models, blockchain systems, privacy, and trust-aware applications.