# Anonymous Role-Based Access Control on E-Health Records

Xingguang Zhou
zhouxingguang@buaa.edu.cn

Jianwei Liu
liujianwei@buaa.edu.cn

Weiran Liu[*]
liuweiran900217@gmail.com

Qianhong Wu
qianhong.wu@buaa.edu.cn

School of Electronic and Information Engineering
Beihang University
Beijing 100191, China

## ABSTRACT

Electronic Health Record (EHR) system facilitates us a lot for health record management. Privacy risk of patients' records is the dominating obstacle in the widely deployed EHRs. Role-based access control (RBAC) schemes offer an access control on EHRs according to one's role. Only the medical staff with roles satisfying the specified access policies can read EHRs. In existing schemes, attackers can link patients' identities to their doctors. Therefore, the classification of patients' diseases are leaked without actually knowing patients' EHRs. To address this problem, we present an anonymous RBAC scheme. Not only it achieves flexible access control, but also realizes privacy-preserving for individuals. Moreover, our scheme maintains the property of constant size for the encapsulated EHRs. The proposed security model with both semantic security and anonymity can be proven under decisional bilinear group assumptions. Besides, we provide an approach for EHR owners to search out their targeted EHR in the anonymous system. For better user experience, we apply "online/offline" approach to speed up data processing in our scheme. Experimental results show that the time consumption for key generation and EHR encapsulation can be done in milliseconds.

## Keywords

anonymous; electronic health record; privacy preserving; access control; online/offline

## 1. INTRODUCTION

There is a trend for Electronic Health Record (EHR) system to be a preferable alternative in healthcare service. An EHR system enables users to share their health related information in an efficient and flexible way. For instance, to find one's prescription, the patient or his doctors just need to retrieve it from database, instead of seeking from piled-up paper. Due to the sensitivity of health records, providing secure storage and access to EHRs is the major challenge in morden EHR system. As most EHRs are computerized in cloud server, they are open to potential threats and vulnerable to loss, leakage, and theft [4]. To prevent EHRs from illegal access, a standard way is to encapsulate patients' EHRs before uploading them to server. In details, the EHR owner encapsulates the EHR by a symmetric session key and only the proper medical staff has the access privilege to decapsulate. However, this solution leads to inflexible data sharing. One issue is that it raises complicated key management and repetitive encryption [16]. i.e., patients usually do not know who will be allowed to access their EHRs, so they encapsulate many pieces with distinct session keys and distribute the keys to different medical staff. This limits the ability of patient to share their data at a coarse-grained level.

To address this problem, several schemes employing the attribute-based encryption (ABE) have been proposed for fine-grained access control [3, 22] in EHR system. Users whose attributes satisfies the access policy would be able to decrypt the EHR data. Furthermore, some advanced models have been presented recently. For instance, an access control model with modular and dynamic management in EHR data [11], and a view-based access control model [10] allowing patient to create a view on EHR data and to specify a list of able users and not able users. Besides, the role-based access control scheme (RBAC) [20] enables a fine-grained access control without ABE system. It offers a role-based access policy in a hierarchical identity-based broadcast encryption (HIBBE) system. While all above proposals are shown to achieve data privacy protection very well in EHR system, patient privacy is still an open issue. i.e., a patient named Lily , her EHR is encapsulated and stored in cloud server, which is secure enough that no attacker can decapsulate it. However, the attackers can get her disease related information by linking Lily to her cheif doctor. If the doctor

[*]Weiran Liu is the corresponding author of this paper.

*CCS '16 May 30–June 3, 2016, Xi'an, China*

ACM ISBN 978-1-4503-2138-9. . . $15.00

DOI: 10.475/123_4

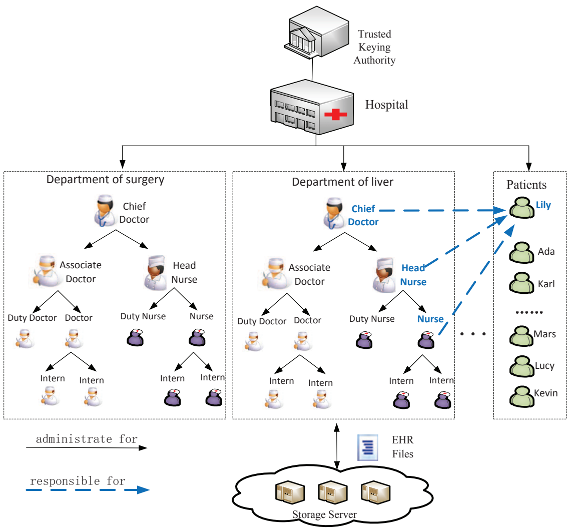


Figure 1: System Architecture: a typical hospital system

is responsible for the disease of hepatitis B, the attackers can infer that "Lily" may carry hepatitis B without reading her EHR. Once the information spreads, Lily may be unfair treated at work, or subject to the deliberate threat of criminals. If there is an anonymous scheme which blinds all the related identities during Lily's examination, the attackers can only get "someone" carries hepatitis B. Thus, the patient privacy is preserved.

## 1.1 Contribution

We propose a novel scheme to provide an anonymous role-based fine-grained access control for EHR sharing. Our scheme employing hierarchical identity-based broadcast encryption is constructed in a typical hospital system, as Figure 1 depicted. We group the patient and his responsible medical staff into a certain access policy. Every user has one private key corresponding to his role which is used to decapsulate the entry which in turn provides access to the encapsulated EHR. A user can access patient EHR if and only if his role satisfy the access policy. Scalable EHR sharing is supported by allowing senior medical staff to delegate access privilege to their subordinates. To achieve identity privacy-preserving, we blind identity related information in the system. Third parties or attackers get no useful information of EHRs nor patient identities. Specific techniques are highlighted as followings.

*Identity Privacy.* We build our scheme on a bilinear group with two subgroups [7]. Identity-related information is hidden in one of the subgroups. The element in this subgroup cannot be distinguished from a random element chosen from the bilinear group.

*Versatile Access Control.* A user encapsulates the EHR with an on-demand access policy. It enables one-to-many encryption, that is, one only needs to encapsulate EHR one time and allows different medical staff to access it.

*Constant Size Ciphertext.* Our scheme achieves constant size ciphertext no matter how many roles of medical staff satisfy the access policy. We stress that it is asymptotical optimal.

*Anonymous Search.* We provide an approach for anonymous search such that the patient and his doctors can link themselves to the targeted EHR, but the outsiders cannot.

*Perfect User Experience.* Online/Offline cryptography is extended to our scheme. The offline phase executes most computations before it knows access control policy and EHR. The online phase rapidly assembles secret key and the encapsulated EHR once these specifics become known.

## 1.2 Related Work

EHR is defined by Iakovidis as "digitally stored healthcare information about an individual's lifetime with the purpose of supporting continuity of care, education and research, and ensuring confidentiality at all times" [17]. The essential of the EHR system is represented by the patients's health data and the ability to guarantee the confidentiality of these data. To secure patients' data in EHR system, the mechanism of access control [23] is widely used. Access control is usually specified by some legislations, i.e. health insurance portability and accountability act (HIPAA)[1], electronic documents [2], or company regulations. It limits who can access the stored EHRs and how they can operate them.

How to achieve appropriate access control is highlighted and required so as to secure EHR data [24]. Existing scheme [21] enables legitimate sharing of access rights in a security architecture. Some proposals employing identity-based encryption [5, 14] allow users to implement access control with security and privacy enabled approaches. Cross-domain EHR sharing [29] and emergency EHR sharing [30] are also presented for some specific requirements in EHR system. Although these models have been proposed and argued to be suitable for healthcare, they are not appealing enough since flexible access control is not available. Attribute-based encryption [13] is one of the solutions for this problem, since attributes can be used to describe users' privileges, and data owner determines the policy on who can access the data. For instance, the scheme [22] with attribute-based encryption achieves fine-grained access in EHR system, where the data owner stores his encrypted data in the cloud and grants access of the data according to the users' identity information. For further accordance with the actual healthcare system with hierarchical organizations, the proposals with hierarchical access control would be a preferable choice. For instance, the scheme [16] enables scalable EHR data sharing, and also combines identity-based and attribute-based encryption together to enforce access control policy. The role-based access control scheme [20] with hierarchical identity-based broadcast encryption achieves flexible access control and scalable data sharing either.

Although a large body of schemes have been presented to secure EHR data successfully, especially with a fine-grained manner, there is missing consideration for the privacy of individuals who are the owners of the shared data. Anonymization can be used to preserve the privacy of data [26]. An anonymous attribute-based scheme employing ABE addressing not only the data privacy, but also the individual identity privacy [31]. Besides, an anonymous-ciphertext policy attribute-based encryption (CPABE) is able to ensure security and privacy preserved fine-grained access control in EHR system [25]. These prominent schemes provide detailed analysis for security, flexibility and anonymity. However, there are remaining unaddressed challenges for the deployment in the real word, where a healthcare system is usually structured hierarchically, with a scalable sharing of

EHRs among large amount of users. In this paper, we propose a novel scheme to achieve patient privacy preserving in a hierarchical system. Scalable data sharing is achieved by higher-level medical staff delegating the access privilege for the lower-level one. The delegation algorithm is used to construct the hierarchical structure. Anonymous algorithm [27] is used to achieve patient privacy preserving.

### 1.3 Paper Organization

In Section 2, we review necessary background information related to our work, including the notations, the introductions of bilinear groups and the theoretical assumptions we use. Section 3 formalizes anonymous RBAC system and the security model. Then we describe our proposals and prove its security in Section 4. We introduce the approach to improve user experience in Section 5. Theoretical analysis and experimental performance are described in Section 6. Finally, we conclude our work in Section 7.

## 2. PRELIMINARIES

### 2.1 Notations

We introduce several notations to simplify the illustration of our scheme. For ease of description, we cite parts of symbols used in [20], where a role-base access control was proposed. Table 1 summarizes these notations and their corresponding meanings.

Table 1: Notations

| Notation | Description |
|---|---|
| $\lambda$ | Security Parameter |
| $Id$ | Identity for Patient |
| $\mathcal{R}$ | Atom Role for medical staff |
| $\vec{\mathcal{R}}$ | Role for medical staff |
| $S_{\vec{\mathcal{R}}}$ | Atom Role Set for $\vec{\mathcal{R}}$ |
| $\mathcal{P}$ | Access Policy |
| $S_{\mathcal{P}}$ | Atom Role Set for $\mathcal{P}$ |
| $Pref(\vec{\mathcal{R}})$ | Prefix of $\vec{\mathcal{R}}$, defined as $\{(\mathcal{R}_1, ..., \mathcal{R}_{d'}) : d' \leq d\}$ |
| $Pref(\mathcal{P})$ | Prefix of $\mathcal{P}$, defined as $\bigcup_{\vec{\mathcal{R}} \in \mathcal{P}} Pref(\vec{\mathcal{R}})$ |
| $MSK$ | Master Secret Key |
| $SK^{\vec{R}}$ | Secret Key for a Role $\vec{\mathcal{R}}$ |
| $EHR$ | Electronic Health Record |
| Hdr | Header of an uploaded EHR |
| $K$ | Message Encapsulation Key |
| $CT$ | Ciphertext for the encapsulated EHR |
| $H$ | Collision resistant hash function $\{0,1\}^{\star} \to \mathbb{Z}_N$ |
| $SymEnc$ | A secure symmetric encryption algorithm |
| $SymDec$ | A secure symmetric decryption algorithm |

### 2.2 Bilinear Groups

Let $\mathcal{G}$ be a group generation algorithm that takes a security parameter $\lambda$ as input and outputs the description of a bilinear group $(N, \mathbb{G}, \mathbb{G}_T, e)$, where $p$ and $q$ are two large prime factors, $\mathbb{G}$ and $\mathbb{G}_T$ are cyclic groups of order $N = p \cdot q$, and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is an efficient bilinear map satisfying the following properties:

- *Bilinearity*: For all $g, h \in \mathbb{G}$ and all $a, b \in \mathbb{Z}_N$, $e(g^a, h^b) = e(g, h)^{ab}$;
- *Non-degeneracy*: There exists at least a generator $g$ in $\mathbb{G}$ such that $e(g, g)$ generates $\mathbb{G}_T$;

We respectively denote the two subgroups of order $p$ and $q$ in $\mathbb{G}$ by $\mathbb{G}_p$ and $\mathbb{G}_q$, and the two subgroups of order $p$ and $q$ in $\mathbb{G}_T$ by $\mathbb{G}_{T,p}$, $\mathbb{G}_{T,q}$, respectively. The generators of the two subgroups $\mathbb{G}_p$ and $\mathbb{G}_q$ are denoted respectively by $g_p$ and $g_q$. These two subgroups additionally satisfy the orthogonality property, i.e., $\forall h_p \in \mathbb{G}_p, h_q \in \mathbb{G}_q, e(h_p, h_q) = 1$.

Composite-order bilinear groups were first introduced by Boneh, Goh and Nissim [7]. This tool is now widely used for constructing cryptographic primitives with abundant security results [8, 18, 19]. We use bilinear groups with composite order in this paper.

### 2.3 Theoretical Assumptions

Our security analysis is based on the following four assumptions.

**$l$-Bilinear Diffie-Hellman Exponent assumption**($l$-BDHE). This assumption was introduced by Boneh, Boyen and Goh [6]. Chooses a random exponent $a \xleftarrow{R} \mathbb{Z}_N$, and a random element $h \xleftarrow{R} \mathbb{G}_p$. The decision $l$-BDHE problem in $\mathbb{G}$ is to determine whether the given element $T_1 \in \mathbb{G}_{T,p}$ equals $e(g_p, h)^{a^l}$, or a random element in $\mathbb{G}_{T,p}$, by taking the input as

$$D_1 \leftarrow \left( \begin{array}{c} (N, \mathbb{G}, \mathbb{G}_T, e), g_p, g_q, h, g_p^{a}, g_p^{a^2}, \cdots, g_p^{a^{l-1}}, \\ g_p^{a^{l+1}}, \cdots, g_p^{a^{2l}} \end{array} \right)$$

The advantage of an algorithm $\mathcal{A}$ that outputs $b \in \{0, 1\}$ in solving the decision $l$-BDHE$^*$ problem is defined as

$$Adv_{\mathcal{A}}(\lambda) = \left| \begin{array}{r} \Pr\left[\mathcal{A}\left(D_1, T_1 \leftarrow e(g_p, h)^{a^l}\right) = 1\right] \\ - \;\Pr\left[\mathcal{A}\left(D_1, T_1 \xleftarrow{R} \mathbb{G}_{T,p}\right) = 1\right] \end{array} \right| - \frac{1}{2}$$

where the probability is over the random bits used by $\mathcal{A}$, the random choice of $T_1 \in \mathbb{G}_{T,p}$.

*Definition 1.* The decision $(t, \epsilon, l)$-BDHE assumption in $\mathbb{G}$ states that no $t$-time algorithm has advantage at least $\epsilon$ in solving the decision $l$-BDHE problem in $\mathbb{G}$.

**Bilinear Subset Decision assumption** (BSD). This assumption was introduced by Boneh, Sahai and Waters [9]. The decision BSD problem in $\mathbb{G}$ is to decide whether the given element $T_2$ is a random element in the subgroup $\mathbb{G}_{T,p}$, or a random element in $\mathbb{G}_T$, by taking the input as

$$D_2 \leftarrow ((N, \mathbb{G}, \mathbb{G}_T, e), g_p, g_q)$$

We define the advantage of an algorithm $\mathcal{A}$ that outputs $b \in \{0, 1\}$ in solving the BSD problem as

$$Adv_{\mathcal{A}}(\lambda) = \left| \begin{array}{l} \Pr\left[\mathcal{A}\left(D_2, T_2 \xleftarrow{R} \mathbb{G}_{T,p}\right) = 1\right] \\ - \Pr\left[\mathcal{A}\left(D_2, T_2 \xleftarrow{R} \mathbb{G}_T\right) = 1\right] \end{array} \right| - \frac{1}{2}$$

where the probability is over the random bits used by $\mathcal{A}$, and the random choice of $T_2 \in \mathbb{G}_T$.

*Definition 2.* The $(t, \epsilon)$-BSD assumption in $\mathbb{G}$ states that there exists no $t$-time algorithm that has advantage at least $\epsilon$ in solving the BSD problem in $\mathbb{G}$.

**$l$-composite Diffie-Hellman assumption** ($l$-cDH). This assumption was introduced by Seo *et al.* [27]. Picks two random exponents $a, b \xleftarrow{R} \mathbb{Z}_N$, and three random elements $R_1, R_2, R_3 \xleftarrow{R} \mathbb{G}_q$. Given the input as

$$D_3 \leftarrow \begin{pmatrix} (N, \mathbb{G}, \mathbb{G}_T, e), g_p, g_q, g_p^a, g_p^{a^2}, \cdots, g_p^{a^l}, \\ g_p^{a^{l+1}} \cdot R_1, g_p^{a^{l+1} \cdot b} \cdot R_2 \end{pmatrix}$$

the decision $l$-cDH problem is to determine whether the given element $T_3$ equals $g_p^b \cdot R_3$, or a random element in $\mathbb{G}$. The advantage of an algorithm $\mathcal{A}$ that outputs $b \in \{0, 1\}$ in solving the decision $l$-cDH problem is defined as

$$Adv_{\mathcal{A}}(\lambda) = \left| \begin{array}{c} \Pr\left[\mathcal{A}\left(D_3, T_3 = g_p^b \cdot R_3\right) = 1\right] \\ -\Pr\left[\mathcal{A}\left(D_3, T_3 \xleftarrow{R} \mathbb{G}\right) = 1\right] \end{array} \right| - \frac{1}{2}$$

where the probability is over the random bits used by $\mathcal{G}$, the random choice of $T_3 \in \mathbb{G}$, and the random bits used by $\mathcal{A}$.

*Definition 3.* The $(t, \epsilon, l)$-decision cDH assumption in $\mathbb{G}$ states that there exists no $t$-time algorithm that has advantage at least $\epsilon$ in solving the decision $l$-cDH problem in $\mathbb{G}$.

**$l$-composite Diffie-Hellman Exponent assumption** ( $l$-cDHE). This assumption is the transformation of $l$-cDH assumption in composite-order bilinear groups . Picks two random exponents $a, b \xleftarrow{R} \mathbb{Z}_N$, and three random elements $R_1, R_2, R_3 \xleftarrow{R} \mathbb{G}_q$. Given the input as

$$D_4 \leftarrow \begin{pmatrix} (N, \mathbb{G}, \mathbb{G}_T, e), g_p, g_q, g_p^a, g_p^{a^2}, \cdots, g_p^{a^l}, \\ g_p^{a^{l+1}} \cdot R_1, g_p^{a^{l+1} \cdot b} \cdot R_2, g_p^{a^{l+2}}, \cdots, g_p^{a^{2l}}, \end{pmatrix}$$

the decision $l$-cDHE problem is to determine whether the given element $T_4$ equals $g_p^b \cdot R_3$, or a random element in $\mathbb{G}$. The advantage of an algorithm $\mathcal{A}$ that outputs $b \in \{0, 1\}$ in solving the decision $l$-cDHE problem is defined as

$$Adv_{\mathcal{A}}(\lambda) = \left| \begin{array}{c} \Pr\left[\mathcal{A}\left(D_4, T_4 = g_p^b \cdot R_3\right) = 1\right] \\ -\Pr\left[\mathcal{A}\left(D_4, T_4 \xleftarrow{R} \mathbb{G}\right) = 1\right] \end{array} \right| - \frac{1}{2}$$

where the probability is over the random bits used by $\mathcal{G}$, the random choice of $T_4 \in \mathbb{G}$, and the random bits used by $\mathcal{A}$.

*Definition 4.* The $(t, \epsilon, l)$-decision cDHE assumption in $\mathbb{G}$ states that there exists no $t$-time algorithm that has advantage at least $\epsilon$ in solving decision $l$-cDHE problem in $\mathbb{G}$.

# 3. SYSTEM MODEL

## 3.1 System Architecture

The system architecture is depicted in Figure 1. In the system, Trusted Keying Authority (TKA) is responsible for generating and distributing system parameters, rooting master keys and authorizing the top-level medical staff and patient. Top-level medical staff delegate keys to his subordinates, which implies a tree-like organization. Each staff is identified by a role consisting of ordered atom roles. For example, the role of a nurse consisting of the ordered atom roles "dept.surgery, chief doctor, head nurse, nurse", is administrated by the head nurse whose role is "dept.surgery, chief doctor, head nurse". The head nurse is then administrated by the chief doctor and so on. We group the chief doctor, the head nurse and the nurse in one Access Policy, where all of them are responsible for a certain patient. The patient is identified by his own identity related information. Each user can encapsulate the patient's EHR, while only the one whose role satisfies the corresponding access policy or the patient himself can decapsulation it. Besides, we blind all the identities in the system such that no one can infer personal information of patients. The system works as follows.

$(PK, MSK) \leftarrow \mathsf{Setup}(\lambda, n)$. The setup algorithm takes as inputs the security parameter $\lambda$ and the maximal size $n$ of users. It outputs a masker key $MSK$ and a public key $PK$.

$SK^{\vec{R}} \leftarrow \mathsf{KeyGenM}(PK, MSK, \vec{\mathcal{R}})$. The medial staff key generation algorithm takes as inputs the public key $PK$, the master key $MSK$, and a role $\vec{\mathcal{R}}$ for a medical staff. It outputs a secret key $SK^{\vec{R}}$ for the medical staff with role $\vec{\mathcal{R}}$.

$SK^{\vec{R}} \leftarrow \mathsf{KeyDelegM}(PK, SK^{\vec{R}'}, \mathcal{R})$. The medical staff key delegation algorithm takes as inputs the public key $PK$, the secret key $SK^{\vec{R}'}$ for a medical staff with role $\vec{\mathcal{R}}'$ and an atom role $\mathcal{R}$. It returns a secret key $SK^{\vec{R}}$ for the medical staff with role $\vec{R} = (\vec{R}', \mathcal{R})$.

$SK^{Id} \leftarrow \mathsf{KeyGenP}(PK, MSK, Id)$. The patient key generation algorithm takes as inputs the public key $PK$, the master key $MSK$, and a patient's identity $Id$. It outputs a secret key $SK^{Id}$ for that patient.

$(Hdr, En) \leftarrow \mathsf{EHREnc}(PK, Id, \mathcal{P}, EHR)$. The EHR encapsulation algorithm takes as inputs the public key $PK$, a patient's identity $Id$, an access policy $\mathcal{P}$ for a group of entitled medical staff, and the EHR file $EHR$. The algorithm outputs the ciphertext $(Hdr, En)$, where $En$ is the encapsulated EHR data by a session key $K$ hidden in the header $Hdr$. We assume that the access policy $\mathcal{P}$ assigned to the EHR file is also included in the header $Hdr$.

$EHR \leftarrow \mathsf{EHRDecM}(PK, Id, (Hdr, En), SK^{\vec{R}})$. The medical staff decapsulation algorithm takes as inputs the public key $PK$, the patient's identity $Id$, the ciphertext $(Hdr, En)$, and the secret key $SK^{\vec{R}}$ for a medical staff with role $\vec{R}$. If $\vec{R} \notin Pref(\mathcal{P})$, the algorithm outputs $\perp$ representing a decapsulation failure. Otherwise, $\vec{R} \in Pref(\mathcal{P})$ so that the secret key $SK^{\vec{R}}$ is able to be used to decapsulate $En$. The algorithm does it by first recovering the session key $K$ from the $Hdr$ and then decapsulating $EHR$ from $En$ with $K$.

$EHR \leftarrow \mathsf{EHRDecP}(PK, Id, (Hdr, En), SK^{Id})$. $\mathsf{EHRDecP}$ algorithm takes as inputs the public key $PK$ , a patient's identity $Id$, a ciphertext $(Hdr, En)$, and a secret key $SK^{Id}$ for that patient. If the ciphertext $(Hdr, En)$ is not for that patient, the algorithm simply outputs $\perp$ to report a decapsulation failure. Otherwise, the ciphertext $(Hdr, En)$ is encapsulated by the identity $Id$. The algorithm recovers the session key $K$ using the secret key $SK^{Id}$. Then, it decapsulates $EHR$ from $En$ with the session key $K$.

## 3.2 Security Model

We use the selective security notion [20] in which the adversary must commit ahead of time the set of roles of medical staff and the identity of a patient it wishes to attack. The security model includes two parts, semantic security model on data confidentiality and anonymity model on identity privacy. We define them by the security games played with a challenger and an adversary $\mathcal{A}$ as followings.

### 3.2.1 Semantic security model

**Init**. The adversary outputs a challenge access policy set $\mathcal{P}$ and a challenge patient's identity $Id$.

**Setup**. The challenger runs Setup algorithm to obtain public key $PK$ and gives it to the adversary $\mathcal{A}$.

**Query Phase 1**. The adversary $\mathcal{A}$ adaptively issues two kinds of queries:

- Secret key query for a medical staff associated with role $\vec{\mathcal{R}}^{\star}$ such that $\vec{\mathcal{R}}^{\star} \notin Pref(\mathcal{P})$. The challenger generates a secret key for $\vec{\mathcal{R}}^{\star}$ and gives it to the adversary.

- Secret key query for a patient with identity $Id^{\star}$ such that $Id^{\star} \neq Id$. The challenger generates a secret key for $Id^{\star}$ and gives it to the adversary.

**Challenge**. When adversary $\mathcal{A}$ decides that it obtains enough secret keys, it outputs two equal-length $EHR_0, EHR_1$ on which it wishes to be challenged. The challenger picks a random bit $b \in \{0,1\}$, and encapsulates the $EHR_b$ under the challenge access policy set $\mathcal{P}$ and the challenge identity $Id$. It gives the challenge ciphertext $(Hdr, En)$ to the adversary $\mathcal{A}$. $En$ is the output of the encapsulation of $EHR_b$.

**Query Phase 2**. Phase 1 is repeated adaptively.

**Guess**. The adversary $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$ and wins the game if $b' = b$.

We require that no polynomial time adversary can distinguish a ciphertext of challenge EHR with the challenge access policy set $\mathcal{P}$ and the challenge patient's identity $Id$, from a ciphertext of random message with the challenge access policy set $\mathcal{P}$ and the challenge patient's identity $Id$.

### 3.2.2 Anonymity model

The phases of **Init**, **Setup**, **Query** are the same as that in Semantic security model. We stress the phases of **Challenge** and **Guess** here.

**Challenge**. When adversary $\mathcal{A}$ decides that it obtains enough secret keys, it outputs two equal-length $EHR_0, EHR_1$ on which it wishes to be challenged. The challenger picks a random bit $b \in \{0,1\}$. If $b = 0$, it generates the header $Hdr$ of ciphertext under the challenge access policy set $\mathcal{P}$ and the challenge identity $Id$, and encapsulates $EHR_0$. If $b = 1$, it generates the header of ciphertext under a random access policy set and a random patient's identity, and encapsulates $EHR_1$. It gives the challenge ciphertext $(Hdr, En)$ to the adversary $\mathcal{A}$. $En$ is the output of the encapsulation of $EHR_b$.

**Guess**. The adversary $\mathcal{A}$ outputs a guess $b'$ and wins the game if $b' = b$.

We require that no polynomial time adversary can distinguish a ciphertext of challenge EHR with the challenge access policy set $\mathcal{P}$ and the challenge patient's identity $Id$, from a ciphertext of challenge EHR with a random access policy set and a random patient's identity.

# 4. PROPOSED APPROACH

## 4.1 The Construction

In this section, we propose an anonymous role-based access control scheme. The scheme is based on the HIBE scheme proposed by Boneh *et al.* [6] and RBAC scheme proposed by Liu *et al.* [20] which offers an efficient approach to support hierarchical access control. The anonymous property is motivated by Seo *et al.* [27], where anonymity is achieved by leveraging bilinear groups with composite order $N = p \cdot q$. Elements in the public parameters are utilized in two separate layers: "key generation layer" and "anonymity layer". Elements in the "key generation layer" are in the subgroup $\mathbb{G}_p$. They provide the secret key and master secret key functionality. Elements in the "anonymity layer" are blinded by the elements in the subgroup $\mathbb{G}_q$. They help to ensure anonymity. In this way, we offer information about the subgroup $\mathbb{G}_p$ in "key generation layer", while keep our scheme's anonymity by the help of "anonymity layer".

$\mathsf{Setup}(\lambda, n)$. It is run by TKA to establish the system. We assume that the patient identity and the medical staff roles are elements in $\mathbb{Z}_N$. A secure symmetric encryption scheme with algorithms $\mathsf{SymEnc}(K, EHR)$ and $\mathsf{SymDec}(K, En)$, and a collision resistant hash $H : \{0,1\}^* \to \mathbb{Z}_N$ are employed in our scheme. TKA picks a random exponent $\alpha \overset{R}{\leftarrow} \mathbb{Z}_N$, random elements $\omega, g_p, g, f, u, g_h, \{h_i\}_{i\in[1,n]} \overset{R}{\leftarrow} \mathbb{G}_p$, and random elements $g_q, R_g, R_f, R_u, R_h, \{R_{h_i}\}_{i\in[1,n]} \overset{R}{\leftarrow} \mathbb{G}_q$. Next, it computes

$$E = e(g, \omega), \quad G = g \cdot R_g, \quad F = f \cdot R_f, \quad U = u \cdot R_u,$$
$$H = g_h \cdot R_h, \quad \{H_i = h_i \cdot R_{h_i}\}_{i\in[1,n]}$$

The public key $PK$ includes the description of composite-order bilinear groups $(N, \mathbb{G}, \mathbb{G}_T, e)$, as well as

$$PK = \{g_p, g_q, G, F, U, H, \{H_i\}_{i\in[1,n]}, E\}$$

The master key is $MSK = \{\omega, p, q, g, f, u, g_h, \{h_i\}_{i\in[1,n]}\}$, which is kept by TKA.

$\mathsf{KeyGenM}(PK, MSK, \vec{\mathcal{R}})$. For a medical staff associated with role $\vec{\mathcal{R}} = (\mathcal{R}_1, ..., \mathcal{R}_d)$, denote $I = \{i : R_i \in S_{\vec{\mathcal{R}}}\}$. When the medical staff wants to join the hospital system, he should be authenticated by TKA firstly. Next, if he is the top-level medical staff, TKA generate a secret key $SK^{\vec{\mathcal{R}}}$ for him. It picks random exponents $r_1, r_2, s_1, s_2, t_1, t_2 \overset{R}{\leftarrow} \mathbb{Z}_N$ satisfying that $s_1 \cdot t_2 - s_2 \cdot t_1 \neq 0 \bmod p$ and $s_1 \cdot t_2 - s_2 \cdot t_1 \neq 0 \bmod q$. If the equations do not hold, TKA picks other random exponents and repeats the procedure. It outputs the secret key $SK^{\vec{\mathcal{R}}}$ that consists of two sub-keys: the sub-key $SK_d^{\vec{\mathcal{R}}}$ is used for decryption and delegation, and the sub-key $SK_r^{\vec{\mathcal{R}}}$ is used for re-randomization.

$$SK_d^{\vec{\mathcal{R}}} = \left\{ \omega \left( u \prod_{i\in I} h_i^{\mathcal{R}_i} \right)^{r_1} f^{r_2}, g^{r_1}, g^{r_2}, g_h^{r_1}, \{h_j^{r_1}\}_{j\in[1,n]\setminus I} \right\} \tag{1}$$

$$SK_r^{\vec{\mathcal{R}}} = \left\{ \begin{array}{l} \left( u \prod_{i\in I} h_i^{\mathcal{R}_i} \right)^{s_1} f^{s_2}, g^{s_1}, g^{s_2}, g_h^{s_1}, \{h_j^{s_1}\}_{j\in[1,n]\setminus I}, \\ \left( u \prod_{i\in I} h_i^{\mathcal{R}_i} \right)^{t_1} f^{t_2}, g^{t_1}, g^{t_2}, g_h^{t_1}, \{h_j^{t_1}\}_{j\in[1,n]\setminus I} \end{array} \right\} \tag{2}$$

Finally TKA outputs $SK^{\vec{\mathcal{R}}} = \left\{SK_d^{\vec{\mathcal{R}}}, SK_r^{\vec{\mathcal{R}}}\right\}$ for the medical staff. Actually, TKA can generate secret keys for any medical staff with authorized roles by running $\mathsf{KeyGenM}$.

$\mathsf{KeyDelegM}(PK, SK^{\vec{\mathcal{R}}'}, \mathcal{R})$. The secret key for a low-level medical staff associated with role $\vec{\mathcal{R}} = (\vec{\mathcal{R}}', \mathcal{R})$ is derived from a given secret key of his supervisor at a higher-level $SK^{\vec{\mathcal{R}}'} = (SK_d^{\vec{\mathcal{R}}'}, SK_r^{\vec{\mathcal{R}}'})$ associated with role $\vec{\mathcal{R}}'$, where

$$SK_d^{\vec{\mathcal{R}}'} = \left\{a_{d,0}, a_{d,1}, a_{d,2}, a_{d,3}, \{b_{d,j}\}_{j\in[1,n]\setminus I'}\right\}$$
$$SK_r^{\vec{\mathcal{R}}'} = \left\{\begin{array}{l} a_{r,0}, a_{r,1}, a_{r,2}, a_{r,3}, \{b_{r,j}\}_{j\in[1,n]\setminus I'}, \\ a'_{r,0}, a'_{r,1}, a'_{r,2}, a'_{r,3}, \{b'_{r,j}\}_{j\in[1,n]\setminus I'} \end{array}\right\}$$

and $I' = \{i : \mathcal{R}_i \in S_{\vec{\mathcal{R}}'}\}$. The high-level medical staff generates a secret key $SK^{\vec{\mathcal{R}}}$ for the low-level one that consists of two parts as well: the decryption part $SK_d^{\vec{\mathcal{R}}}$ and the rerandomization part $SK_r^{\vec{\mathcal{R}}}$.

For the decryption part $SK_d^{\vec{\mathcal{R}}}$, the high-level medical staff picks random exponents $\gamma_1, \delta_1 \xleftarrow{R} \mathbb{Z}_N$ and delegates the secret key for the low-level one by using

$$SK_d^{\vec{\mathcal{R}}} = \left\{d_1, d_2, d_3, d_4, \{d_j\}_{j\in[1,n]\setminus I}\right\} =$$
$$\left\{\begin{array}{c} \left(\left(a_{d,0}(b_{d,i}^{\mathcal{R}})\right) \cdot \left(a_{r,0}(b_{r,i}^{\mathcal{R}})\right)^{\gamma_1} \cdot \left({a'}_{r,0}({b'_{r,i}}^{\mathcal{R}})\right)^{\delta_1}\right)_{i\in I\setminus I'} \\ a_{d,1} \cdot a_{r,1}^{\gamma_1} \cdot {a'}_{r,1}^{\delta_1}, \quad a_{d,2} \cdot a_{r,2}^{\gamma_1} \cdot {a'}_{r,2}^{\delta_1}, \quad a_{d,3} \cdot a_{r,3}^{\gamma_1} \cdot {a'}_{r,3}^{\delta_1} \\ \{b_{d,j} \cdot b_{r,j}^{\gamma_1} \cdot {b'}_{r,j}^{\delta_1}\}_{j\in[1,n]\setminus I} \end{array}\right\}$$

where $I = \{i : \mathcal{R}_i \in S_{\vec{\mathcal{R}}}\}$. Finally, the delegated secret key $SK_d^{\vec{\mathcal{R}}}$ can be attained in the form

$$SK_d^{\vec{\mathcal{R}}} = \left\{d_1, d_2, d_3, d_4, \{d_j\}_{j\in[1,n]\setminus I}\right\}$$
$$= \left\{\omega\left(u \cdot \prod_{i\in I} h_i^{\mathcal{R}_i}\right)^{\widetilde{r}_1} f^{\widetilde{r}_2}, g^{\widetilde{r}_1}, g^{\widetilde{r}_2}, g_h^{\widetilde{r}_1}, \{h_j^{\widetilde{r}_1}\}_{j\in[1,n]\setminus I}\right\}$$

where

$$\begin{pmatrix} \widetilde{r}_1 \\ \widetilde{r}_2 \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix} + \begin{pmatrix} s_1 & t_1 \\ s_2 & t_2 \end{pmatrix} \begin{pmatrix} \gamma_1 \\ \delta_1 \end{pmatrix}$$

It follows that $SK_d^{\vec{\mathcal{R}}}$ is well-formed as if it is generated directly by TKA with the $\mathsf{KeyGenM}$ algorithm.

For delegating $SK_r^{\vec{\mathcal{R}}}$, the high-level medical staff picks random exponents $\gamma_2, \delta_2, \gamma_3, \delta_3 \xleftarrow{R} \mathbb{Z}_N$ which satisfies that $g_p^{\gamma_2\cdot\delta_3-\gamma_3\cdot\delta_2} \neq 1$, $g_q^{\gamma_2\cdot\delta_3-\gamma_3\cdot\delta_2} \neq 1$. Then, he delegates the secret key by using

$$SK_r^{\vec{\mathcal{R}}} = \left\{\begin{array}{l} d_{r,1}, d_{r,2}, d_{r,3}, d_{r,4}, \{d_{r,j}\}_{j\in[1,n]\setminus I}, \\ d'_{r,1}, d'_{r,2}, d'_{r,3}, d'_{r,4}, \{d'_{r,j}\}_{j\in[1,n]\setminus I} \end{array}\right\} =$$
$$\left\{\begin{array}{c} \left(\left(a_{r,0}(b_{r,i}^{\mathcal{R}})\right)^{\gamma_2} \cdot \left({a'}_{r,0}({b'_{r,i}}^{\mathcal{R}})\right)^{\delta_2}\right)_{i\in I\setminus I'}, \\ a_{r,1}^{\gamma_2} \cdot {a'}_{r,1}^{\delta_2},\ a_{r,2}^{\gamma_2} \cdot {a'}_{r,2}^{\delta_2},\ a_{r,3}^{\gamma_2} \cdot {a'}_{r,3}^{\delta_2},\ \{b_{r,j}^{\gamma_2} \cdot {b'}_{r,j}^{\delta_2}\}_{j\in[1,n]\setminus I}, \\ \left(\left(a_{r,0}(b_{r,i}^{\mathcal{R}})\right)^{\gamma_3} \cdot \left({a'}_{r,0}({b'_{r,i}}^{\mathcal{R}})\right)^{\delta_3}\right)_{i\in I\setminus I'}, \\ a_{r,1}^{\gamma_3} \cdot {a'}_{r,1}^{\delta_3},\ a_{r,2}^{\gamma_3} \cdot {a'}_{r,2}^{\delta_3},\ a_{r,3}^{\gamma_3} \cdot {a'}_{r,3}^{\delta_3},\ \{b_{r,j}^{\gamma_3} \cdot {b'}_{r,j}^{\delta_3}\}_{j\in[1,n]\setminus I} \end{array}\right\}$$

Finally, the delegated secret key $SK_r^{\vec{\mathcal{R}}}$ can be written as

$$SK_r^{\vec{\mathcal{R}}} = \left\{\begin{array}{l} d_{r,1}, d_{r,2}, d_{r,3}, d_{r,4}, \{d_{r,j}\}, \\ d'_{r,1}, d'_{r,2}, d'_{r,3}, d'_{r,4}, \{d'_{r,j}\} \end{array}\right\}_{j\in[1,n]\setminus I}$$
$$= \left\{\begin{array}{c} \left(u \cdot \prod_{i\in I} h_i^{\mathcal{R}_i}\right)^{\widetilde{s}_1} f^{\widetilde{s}_2}, g^{\widetilde{s}_1}, g^{\widetilde{s}_2}, g_h^{\widetilde{s}_1}, \{h_j^{\widetilde{s}_1}\}, \\ \left(u \cdot \prod_{i\in I} h_i^{\mathcal{R}_i}\right)^{\widetilde{t}_1} f^{\widetilde{t}_2}, g^{\widetilde{t}_1}, g^{\widetilde{t}_2}, g_h^{\widetilde{t}_1}, \{h_j^{\widetilde{t}_1}\} \end{array}\right\}_{j\in[1,n]\setminus I}$$

where

$$\begin{pmatrix} \widetilde{s}_1 & \widetilde{t}_1 \\ \widetilde{s}_2 & \widetilde{t}_2 \end{pmatrix} = \begin{pmatrix} s_1 & t_1 \\ s_2 & t_2 \end{pmatrix} \begin{pmatrix} \gamma_2 & \gamma_3 \\ \delta_2 & \delta_3 \end{pmatrix}$$

As a conclusion, by running $\mathsf{KeyDelegM}$ algorithm, the delegated the secret key is well formed as if it is generated directly by TKA with the $\mathsf{KeyGenM}$ algorithm.

$\mathsf{KeyGenP}(PK, MSK, Id)$. When a patient with identity $Id$ wants to access his own EHR, TKA first authorizes him and then assigns a secret key. It picks a random exponent $r'_1, r'_2 \xleftarrow{R} \mathbb{Z}_N$ and outputs

$$SK^{Id} = \left(d'_1, d'_2, d'_3, \{d'_j\}_{j\in[1,n]}\right)$$
$$= \left(\omega(ug_h^{Id})^{r'_1} f^{r'_2}, g^{r'_1}, g^{r'_2}, \{h_j^{r'_1}\}_{j\in[1,n]}\right)$$

$\mathsf{EHREnc}(PK, Id, \mathcal{P}, EHR)$. For an access policy $\mathcal{P}$, denote $\mathbb{I} = \{i : \mathcal{R}_i \in S_{\mathcal{P}}\}$. When EHR needs to be encapsulated under the patient's identity $Id$ and the access policy $\mathcal{P}$, the user (a patient or a medical staff) first picks a random exponent $s \xleftarrow{R} \mathbb{Z}_N$, and random elements $Z_1, Z_2, Z_3 \xleftarrow{R} \mathbb{G}_q$. Note that the random elements in $\mathbb{G}_q$ can be chosen by raising $g_q$ to random exponents from $\mathbb{Z}_N$. Then, the user computes the header $Hdr$ as follows

$$Hdr = \{C_1, C_2, C_3\} = \{G^s\cdot Z_1, F^s\cdot Z_2, \left(UH^{Id}\prod_{i\in\mathbb{I}} H_i^{\mathcal{R}_i}\right)^s Z_3\} \quad (3)$$

Then, the user generates session key $K = E^s$, and computes $En = \mathsf{SymEnc}(K, EHR)$. The encapsulated EHR is output as $CT = (Hdr, En) = (C_1, C_2, C_3, En)$.

$\mathsf{EHRDecM}(PK, Id, (Hdr, En), SK^{\vec{\mathcal{R}}})$. In order to retrieve the session key $K$, a medical staff with role satisfied the access policy $\mathcal{P}$, can use his secret key to compute

$$K = \frac{e\left(d_1 \cdot d_4^{Id} \cdot \left(\prod_{i\in\mathbb{I}\setminus I} d_i^{\mathcal{R}_i}\right), C_1\right)}{e\left(d_2, C_3\right) \cdot e\left(d_3, C_2\right)}$$

It finally runs $EHR = \mathsf{SymDec}(K, En)$ to get the EHR.

$\mathsf{Correctness}$. Assume $CT = ((C_1, C_2, C_3), En)$ is a well-formed ciphertext, the medical staff decapsulation algorithm can correctly decapsulate EHRs with a valid secret key $SK^{\vec{R}}$

with $\vec{R} \in Pref(\mathcal{P})$ since we have that

$$K = \frac{e\left(d_1 \cdot d_4^{Id} \cdot \left(\prod_{i \in \mathbb{I} \setminus I} d_i^{\mathcal{R}_i}\right), C_1\right)}{e\left(d_2, C_3\right) \cdot e\left(d_3, C_2\right)}$$
$$= \frac{e\left(w\left(u \prod_{i \in I} h_i^{\mathcal{R}_i}\right)^{r_1} f^{r_2} \cdot g_h^{r_1 \cdot Id} \cdot \prod_{i \in \mathbb{I} \setminus I} (h_i^{\mathcal{R}_i})^{r_1}, g^s\right)}{e\left(g^{r_1}, \left(u \cdot g_h^{Id} \cdot \prod_{i \in \mathbb{I}} h_i^{\mathcal{R}_i}\right)^s\right) e\left(g^{r_2}, f^s\right)}$$
$$= e(g, \omega)^s$$

The second equality holds because $e(h_p, h_q) = 1$ for all $h_p \in \mathbb{G}_p$ and $h_q \in \mathbb{G}_q$.

EHRDecP$(PK, Id, (Hdr, En), SK^{Id})$. The patient with identity $Id$ can decapsulate his/her own EHRs using his/her secret key. Denote $\mathbb{I} = \{i : \mathcal{R}_i \in S_{\mathcal{P}}\}$. The patient computes the session key

$$K = \frac{e\left(d'_1 \cdot \prod_{i \in \mathbb{I}} {d'}_j^{\mathcal{R}_i}, C_1\right)}{e\left(d'_2, C_3\right) \cdot e\left(d'_3, C_2\right)}$$

It finally runs $EHR = \mathsf{SymDec}(K, En)$ to recover EHR.

Correctness. Assume $CT = ((C_1, C_2, C_3), En)$ is a well-formed ciphertext. The patient can also correctly recover its own EHRs since the following equalities

$$K = \frac{e\left(d'_1 \cdot \prod_{i \in \mathbb{I}} {d'}_j^{\mathcal{R}_i}, C_1\right)}{e(d'_2, C_3) \cdot e(d'_3, C_2)}$$
$$= \frac{e\left(w \cdot u \cdot \left(h_h^{Id}\right)^{{r'}_1} f^{{r'}_2} \cdot \prod_{i \in \mathbb{I}} \left(h_i^{r'_1}\right)^{\mathcal{R}_i}, g^s\right)}{e\left(g^{{r'}_1}, \left(u \cdot g_h^{Id} \cdot \prod_{i \in \mathbb{I}} h_i^{\mathcal{R}_i}\right)^s\right) \cdot e\left(g^{{r'}_2}, f^s\right)}$$
$$= e(g, \omega)^s$$

## 4.2 Security Analysis

The anonymous RBAC scheme achieves selective security and anonymity notion which are described in Section 3.2 based on the decision $l$-BDHE assumption, the BSD assumption , the $l$-cDH assumption and the $l$-cDHE assumption defined in Section 2. Formally, we have the following Theorems.

Theorem 1. *Let $\mathbb{G}$ be a group of composite order $N = p \cdot q$, equipped with an efficient bilinear map. Suppose that the Decision $(n{+}2)$-BDHE assumption and BSD assumption holds in $\mathbb{G}$. Then our proposal is semantic secure under the formal semantic security model defined in Section 3.2.1.*

Theorem 2. *Let $\mathbb{G}$ be a group of composite order $N = p \cdot q$, equipped with an efficient bilinear map. Suppose that $(n{+}1)$-cDH assumption and $(n+1)$-cDHE assumption holds in $\mathbb{G}$. Then our proposal is anonymity secure under the formal anonymity model defined in Section 3.2.2.*

We prove Theorem1 and Theorem2 through the following games between an adversary and a challenger.

- $\mathrm{CT}_1$ of Game$_1$: $((C_1, C_2, C_3,), En)$
- $\mathrm{CT}_2$ of Game$_2$: $((C_1, C_2, C_3,), En \cdot R_p)$
- $\mathrm{CT}_3$ of Game$_3$: $((C_1, C_2, C_3,), En \cdot R = R_{En})$
- $\mathrm{CT}_4$ of Game$_4$: $((R_1, C_2, C_3,), R_{En})$
- $\mathrm{CT}_5$ of Game$_5$: $((R_1, R_2, R_3,), R_{En})$

where $R_p$ is randomly chosen from $\mathbb{G}_{T,p}$; $R$, $R_{En}$ are uniformly distributed in $\mathbb{G}_T$; $R_1$, $R_2$, $R_3$ are uniformly distributed in $\mathbb{G}$.

Lemma 1. *Let $\mathbb{G}$ be a group of composite order $N = p \cdot q$ equipped with an efficient bilinear map. If $\mathbb{G}$ satisfies the $(t, \epsilon_1, n+2)$-BDHE assumption, then there is no polynomial time algorithm that can distinguish Game$_2$ from Game$_1$ with advantage $\epsilon_1$.*

Lemma 2. *Let $\mathbb{G}$ be a group of composite order $N = p \cdot q$ equipped with an efficient bilinear map. If $\mathbb{G}$ satisfies the $(t, \epsilon_2)$-BSD assumption, then there is no polynomial time algorithm that can distinguish Game$_3$ from Game$_2$ with advantage $\epsilon_2$.*

**Proof of Theorem 1**. If group generator algorithm $\mathbb{G}$ satisfies the $(t, \epsilon_1, n+2)$-BDHE assumption and the $(t, \epsilon_2)$-BSD assumption, then Lemma 1 and Lemma 2 show that there is no polynomial time adversary that makes at most $q_s$ key extraction queries to distinguish Game$_1$ and Game$_3$ with advantage $\epsilon_1 + \epsilon_2$. Ciphertext of Game$_3$ does not leak any information about the EHR data since the component regarding to EHR in ciphertext of Game$_3$ is a random group element. Therefore, if the group $\mathbb{G}$ with composite order $N = p \cdot q$ satisfies the $(t, \epsilon_1, n+2)$-BDHE assumption and the $(t, \epsilon_2)$-BSD assumption, then our proposed scheme is semantic secure with advantage $\epsilon_1 + \epsilon_2$.

Lemma 3. *Let $\mathbb{G}$ be a group of composite order $N = p \cdot q$ equipped with an efficient bilinear map. If $\mathbb{G}$ satisfies the $(t, \epsilon_3, n+1)$-cDH assumption, then there is no polynomial time algorithm that can make at most $q_s$ key extraction queries and distinguish Game$_4$ from Game$_3$ with advantage $\epsilon_3/(1 - \frac{p+q-1}{N})^{q_s}$.*

Lemma 4. *Let $\mathbb{G}$ be a group of composite order $N = p \cdot q$ equipped with an efficient bilinear map. If $\mathbb{G}$ satisfies the $(t, \epsilon_4, n+1)$-cDHE assumption, then there is no polynomial time algorithm that can make at most $q_s$ key extraction queries and distinguish Game$_5$ from Game$_4$ with advantage $\epsilon_4/(1 - \frac{p+q-1}{N})^{q_s}$.*

**Proof of Theorem 2**. If group generator algorithm $\mathbb{G}$ satisfies the $(t, \epsilon_3, n+1)$-cDH assumption and the $(t, \epsilon_4, n+1)$-cDHE assumption, then Lemma 3 and Lemma 4 show that there is no polynomial time adversary that makes at most $q_s$ key extraction queries to distinguishe Game$_3$ and Game$_5$ with advantage $\epsilon_3/(1 - \frac{p+q-1}{N})^{q_s} + \epsilon_4/(1 - \frac{p+q-1}{N})^{q_s}$. Ciphertext of Game$_5$ does not leak any information about the roles of medical staff and the identity of patient since the components regarding to roles and identity in ciphertext of Game$_5$ are random group elements. Therefore, if the group $\mathbb{G}$ with composite order $N = p \cdot q$ satisfies the Decision $(n+2)$-BDHE, BSD, $(n{+}1)$-cDH and $(n{+}1)$-cDHE assumptions, then our proposed scheme is anonymous with advantage $\epsilon_1 + \epsilon_2 + \epsilon_3/(1 - \frac{p+q-1}{N})^{q_s} + \epsilon_4/(1 - \frac{p+q-1}{N})^{q_s}$.

In Appendix, we provide concrete proof steps through five games.

### 4.3 Anonymous Search

Above section illustrates that the encapsulated EHR is secure enough that no one can get any identity related information from it, even for the EHR owners. However, EHR system may receive the query from the patient or medical staff to search for someone's EHR. In order to perform a search responsive to the query, we set up an approach that links EHR owners to their encapsulated EHR. We tag two labels, $Id'$ and $\mathcal{P}'$ with each ciphertext $CT$, formed as $(CT_i, Id'_i, \mathcal{P}'_i)$. Assume the total number of stored EHRs is $m$, $i \in [1, m]$. $Id'$ and $\mathcal{P}'$ represent the blinded identity for patient and the blinded roles for medical staff respectively so that outsiders cannot identify them. For the patient and medical staff, following operations show how they can find out the targeted EHR.

**SearchInitial**. In this phase, we generate some parameters prepared for the searching work later. Let $G_0$ be a bilinear group of prime order $p$, and $g$ be a generator of $G_0$. When each ciphertext $CT_i$ generated, the $i$th patient with identity $Id_i$ randomly chooses an element $x_{Id_i} \leftarrow G_0$ and the $i$th group of medical staff in access policy $\mathcal{P}_i$ randomly choose an element $x_{R_i} \leftarrow G_0$. Then they compute a session key $SK_i$: $SK_i \leftarrow g^{x_{Id_i} \cdot x_{R_i}} \bmod n$. $n$ is a large prime number. The session key is only owned by the patient with identity $Id_i$ and his responsible medical staff in access policy $\mathcal{P}_i$.

**SearchLabelCreate**. In this phase, we create the searching labels: $Id'_i$ and $\mathcal{P}'_i$. $Id'_i$ can be obtained by applying hash function on $Id_i$: $Id'_i \leftarrow H(Id_i)$. $\mathcal{P}'_i$ can be obtained by applying the symmetric encryption algorithm $SymEnc$ with the session key $SK_i$ on the atom roles $\{\mathcal{R}_{ij}\}$ in $\mathcal{P}_i$: $\{\mathcal{R}'_{ij} \leftarrow SymEnc(\mathcal{R}_{ij}, SK_i)\}$, $j \in \{j : \mathcal{R}_{ij} \in S_{\mathcal{P}_i}\}$. $\{\mathcal{R}'_{ij}\}$ constitute the atom roles for $\mathcal{P}'_i$. Then the labels $Id'_i$ and $\mathcal{P}'_i$ are tagged with $CT_i$, as the format of $(CT_i, Id'_i, \mathcal{P}'_i)$.

**Search**. When a patient with identity $\mathcal{ID}$ tries to search his EHR (or one of his doctors tries to do this), he first applies hash function on the identity $\mathcal{ID}$ and get $H(\mathcal{ID})$. Then he looks through $Id'_i$ in all the patients' labels and pinpoint the one whose value equals $H(\mathcal{ID})$. When he get the index $i$, he can use his session key to decrypt the roles for medical staff: $\{\mathcal{R}_{ij} \leftarrow SymDec(\mathcal{R}'_{ij}, SK_i)\}$. $\{\mathcal{R}_{ij}\}$ are the atom roles in access policy $\mathcal{P}_i$. When the patient knows the access policy $\mathcal{P}_i$ of medical staff and his identity, he can decapsulate $CT_i$ with corresponding secret key.

## 5. IMPROVING USER EXPERIENCE

For the purpose of better user experience, we speed up the data processing in the procedure of key generation, key delegation and EHR encapsulation. We apply online/offline [15] cryptography to our scheme. Online/offline technique is initiated by Goldreich and Micali [28] for signature scheme. Guo et al. [12] extents the offline algorithm to the identity-based encryption system. Briefly speaking, online/offline technique splits the encryption or key generation process into two phases: the offline phase first executes most of heavy computations by assuming a set of random identities, and then the online phase only performs light computations to produce the ciphertext or secret key once the identities are available. In this way, we illustrate we show how to move the computational work of key generation and EHR encapsulation offline. The following offline/online algorithms are based on our construction in Section 4.1.

Offline.KeyGenM$(PK, MSK)$. The offline KeyGenM algorithm takes in the public parameters and master key, excluding role for medical staff. We assume a random role $\vec{\mathcal{R}}_B$ with bound B on the maximum number of atom roles, which can be used to generate secret key. Denote $\vec{\mathcal{R}}_B = (x_1, x_2, ..., x_B)$ and $I_B = \{i : x_i \in S_{\vec{\mathcal{R}}_B}\}$, where $x_i$ are randomly chosen from $\mathbb{Z}_N$ and regarded as intermediate atom roles. The algorithm picks random exponents $r_1, r_2, s_1, s_2, t_1, t_2 \overset{R}{\leftarrow} \mathbb{Z}_N$ satisfying that $s_1 \cdot t_2 - s_2 \cdot t_1 \neq 0 \bmod p$ and $\bmod q$. Then it generates the intermediate secret key $SK^{\vec{\mathcal{R}}_B}$ that consists of two sub-keys: $SK_d^{\vec{\mathcal{R}}_B}$ and $SK_r^{\vec{\mathcal{R}}_B}$. $SK_d^{\vec{\mathcal{R}}_B}$ can be written as following form

$$\left\{ \omega \left( u \prod_{i \in I_B} h_i^{x_i} \right)^{r_1} f^{r_2}, g^{r_1}, g^{r_2}, g_h^{r_1}, \{h_j^{r_1}\}_{j \in [1,n] \setminus I_B} \right\}$$

$\{h_j^{r_1}\}_{j \in [1,n]}$ can be pre-computed here. $SK_r^{\vec{\mathcal{R}}_B}$ has the similar form as $SK_d^{\vec{\mathcal{R}}_B}$ but not used for EHR encapsulation. We can view the procedure as key generation for the intermediate role $\vec{\mathcal{R}}_B = (x_1, x_2, ..., x_B)$. The work done in the offline phase is roughly equivalent to the work of the regular KeyGenM algorithm, as equations (1) and (2).

Online.KeyGenM$(SK^{\vec{\mathcal{R}}_B}, \vec{\mathcal{R}})$. The online KeyGenM algorithm takes in the intermediate secret key $SK^{\vec{\mathcal{R}}_B}$ from offline KeyGenM algorithm and a real role of medical staff $\vec{\mathcal{R}} = (\mathcal{R}_1, ..., \mathcal{R}_{d \leq B})$. Denote $I = \{i : \mathcal{R}_i \in S_{\vec{\mathcal{R}}}\}$. The algorithm computes the "correction factors" $K_i = r_1 \cdot (\mathcal{R}_i - x_i) \bmod N$ for $i \in I$. The sub-key $SK_d^{\vec{\mathcal{R}}}$ for the medical staff is output as the form of

$$\left\{ \omega \left( u \prod_{i \in I} h_i^{x_i} \right)^{r_1} f^{r_2}, g^{r_1}, g^{r_2}, g_h^{r_1}, \{h_j^{r_1}\}_{j \in [1,n] \setminus I}, \{K_i\}_{i \in I} \right\}$$
$$= \{d_1, d_2, d_3, d_4, \{d_j\}_{j \in [1,n] \setminus I}, \{K_i\}_{i \in I}\}$$

The sub-key $SK_r^{\vec{\mathcal{R}}}$ is output with the form similar to $SK_d^{\vec{\mathcal{R}}}$ but without the elements $\{K_i\}_{i \in I}$. The dominant cost in online phase is $||\vec{\mathcal{R}}||$ multiplications for generating $\{K_i = r_1 \cdot (\mathcal{R}_i - x_i)\}_{i \in I}$.

Since the offline/online algorithm of key delegation follows the same way as that in KeyGenM phase, we skip its detailed process. The dominant cost in online key delegation procedure is one multiplication only.

Offline.EHREnc$(PK)$. The offline EHREnc algorithm takes in the public parameters only. We assume a random access policy $\mathcal{P}_B$ with bound $B$ on the maximum number of atom roles, which can be used to generate a ciphertext. Denote $\mathbb{I}_B = \{i : z_i \in S_{\mathcal{P}_B}\}$, where $z_i$ are randomly chosen from $\mathbb{Z}_N$ and regarded as intermediate atom roles. The algorithm picks up $y \overset{R}{\leftarrow} \mathbb{Z}_N$ and it is assumed as the intermediate patient identity. Then, the algorithm picks a random element $s \overset{R}{\leftarrow} \mathbb{Z}_N$ and random elements $Z_1, Z_2, Z_3 \overset{R}{\leftarrow} \mathbb{G}_q$. Finally it computes the intermediate header $Hdr_{IT}$ as following

$$\begin{aligned} Hdr_{IT} &= \{C_1, C_2, C_3\} \\ &= \left\{ G^s \cdot Z_1, F^s \cdot Z_2, \left( UH^y \prod_{i \in \mathbb{I}_B} H_i^{z_i} \right)^s Z_3 \right\} \end{aligned}$$

The generated header in offline phase is roughly equivalent to the work of regular EHREnc algorithm, as equation (3).

$\mathsf{Online.EHREnc}(Hdr_{IT}, Id, \mathcal{P}, EHR)$. Online EHREnc algorithm takes in the intermediate header $Hdr_{IT}$ from offline EHREnc algorithm, a patient identity $Id$, an access policy $\mathcal{P}$ and $EHR$. Denote $\mathbb{I} = \{i : \mathcal{R}_i \in S_{\mathcal{P}}\}$ and we should notice that $\mathbb{I} \subseteq \mathbb{I}_B$ since we have assumed the maximum bound $B$ on atom role numbers. The algorithm computes the "correction factors" $C_{4,i} = s \cdot (\mathcal{R}_i - z_i)$ and $C_5 = s \cdot (Id - y)$ for $i \in \mathbb{I}$. Then it outputs the ciphertext header

$$\begin{aligned} Hdr &= \{C_1, C_2, C_3, \{C_{4,i}\}_{i \in \mathbb{I}}, C_5\} \\ &= \left\{ G^s \cdot Z_1, F^s \cdot Z_2, \left( UH^y \prod_{i \in \mathbb{I}} H_i^{z_i} \right)^s Z_3, \{C_{4,i}\}_{i \in \mathbb{I}}, C_5 \right\} \end{aligned}$$

As symmetric encryption time $En = \mathsf{SymEnc}(K, EHR)$ is relative fast, the cost for EHR encapsulation can be ignored. The dominant cost in online phase are $(||\vec{\mathcal{P}}|| + 1)$ multiplications in $\mathbb{Z}_N$, for generating $\{C_{4,i} = s \cdot (\mathcal{R}_i - z_i)\}_{i \in \mathbb{I}}$ and $C_5 = s \cdot (Id - y)$.

Finally, we should verify that the EHR can be correctly decapsulated after online/ offline algorithm applied. The message encapsulation key $K$ is calculated by

$$K = \frac{e\left( d_1 \cdot \prod_{i \in I} h_i^{K_i} \cdot d_4^{Id} \cdot \left( \prod_{i \in \mathbb{I} \setminus I} d_i^{\mathcal{R}_i} \right), C_1 \right)}{e\left( d_2, C_3 \cdot \prod_{i \in \mathbb{I}} H_i^{C_{4,i}} \cdot H^{C_5} \right) \cdot e\left(d_3, C_2\right)}$$

$K$ can be extracted as $K = e(g, \omega)^s$ from above expression. Finally, EHR can be exactly recovered by running $EHR = SymDec(K, En)$.

# 6. PERFORMANCE ANALYSIS

## 6.1 Theoretical Analysis

Table 2 shows the efficiency of our proposed scheme in details. The system parameter, the master secret key and secret keys (for medical staff and patients) are linear with the maximum atom role number. The header only contains three group elements in $\mathbb{G}$, which achieves constant size ciphertext and independent of the maximal depth of the hierarchy for the access policy set $\|\mathcal{P}\|$. In Table 2, we denote $t_e$ as one exponent operation time in $\mathbb{G}$, $t_m$ one multiplication operation time in $\mathbb{G}$ and $t_p$ one pairing operation time. We ignore symmetric encryption and decryption time in the efficiency analysis as they are relative fast. In the procedures of KeyGenM, KeyDelegM, KeyGenP, EHREnc, exponentiations can be pre-computed by choosing random exponents.

Table 3 compares four schemes in terms of anonymity, orders of bilinear group and performance. We denote "Ours & Improved" as our scheme with user experience improvement. Other symbols have the same meaning as those in Table 2.

## 6.2 Experimental Performance

We conduct experiment on an Intel Core i7 processor with 8GB RAM and 2.6GHZ CPU clock speed. We use elliptic curve type A1 with elliptic curve expression $y^2 = x^3 + x$ for the Tate symmetric pairing. The group order of $\mathbb{Z}_N$ is set 1024 bits, and the element size in $\mathbb{G}$ is configured 1024 bits as well. The experiment is executed with the jPBC library (http://gas.dia.unisa.it/projects/jpbc/index.html).

We test the operational time for key generation, key delegation, EHR encapsulation and decapsulation for medical staff. Most exponentiations can be pre-computed by choosing the random exponents. As illustrated in Figure 3a, as the number of atom roles increases, the time consumptions of key generation increase gradually. When the number of related atom roles approaches to 10, the key generation time exceeds 5 seconds. As depicted in Figure 3b, with the number of atom role increasing, the time consumption for delegation procedure is almost unchanged, keeping around 2.8 seconds, if necessary pre-computations are done. The test result is equivalent to the analysis in KeyDelegM procedure, where the number of atom role is not variable. The time consumption for EHR encapsulation and decapsulation are tested by setting the number of roles that satisfies the designed access policy. As shown in Figure 3c, when the number of roles involved in access policy approaches to 35, the EHR encapsulation time is beyond 7 seconds. Figure 3d displays the time consumption for EHR decapsulation. The dominant time consumption in our anonymity scheme is introduced by exponentiations and paring operations in composite-order bilinear groups. Figure 3e and Figure 3f show the operational time after user experience is improved. The time consumptions for key generation and EHR encapsulation remain in milliseconds. For the procedure of EHR decapsulation, this is usually executed by users on the desktop computers which own a powerful data processing ability, so we do not focus user experience on it.

Table 2: The Efficiency of the Proposed Scheme

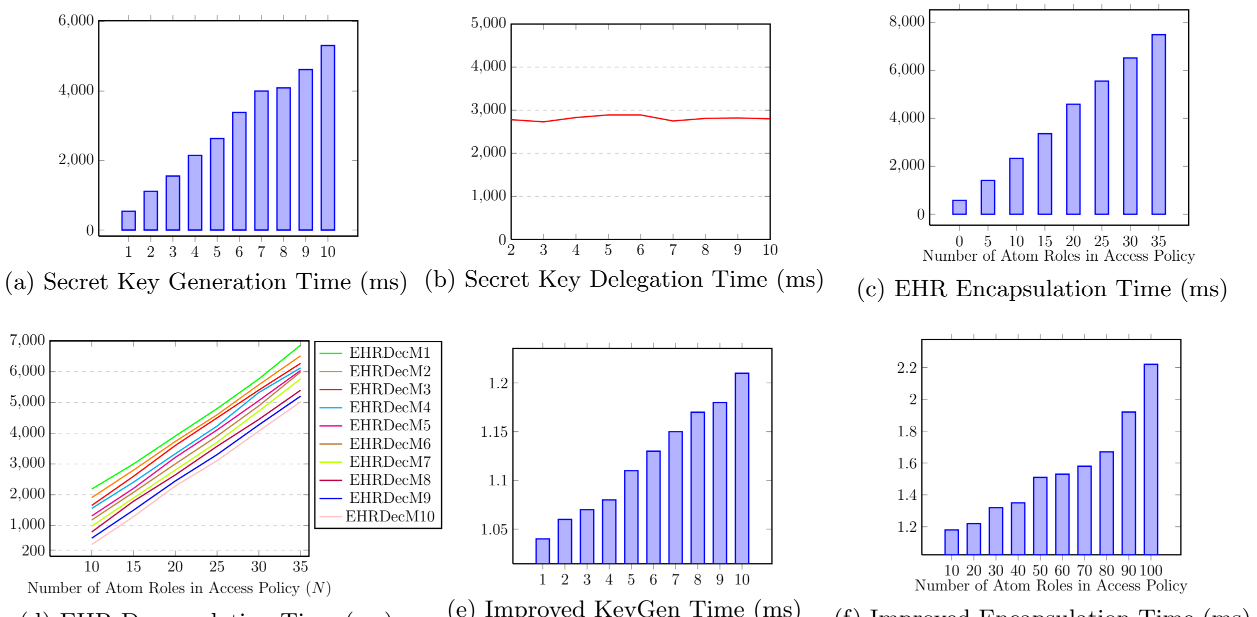
| | Proposed scheme with $n$ Atom Roles |
|---|---|
| $MSK$ Size | $n + 7$ |
| $SK^{\vec{\mathcal{R}}}$ Size | $3 \cdot (n + 4 - \|\vec{\mathcal{R}}\|)$ |
| $SK^{Id}$ Size | $n + 3$ |
| Hdr Size | 3 |
| KeyGenM Time | $3 \cdot (n + 5)t_e + (3\|\vec{\mathcal{R}}\| + 4)t_m$ |
| KeyDelegM Time | $(31 + 6n - 6\|\vec{\mathcal{R}}\|)t_e +$<br>$(23 + 4n - 4\|\vec{\mathcal{R}}\|)t_m$ |
| KeyGenP Time | $(n + 5)t_e + 3$ |
| EHREnc Time | $(\|\mathcal{P}\| + 5)t_e + (\|\mathcal{P}\| + 4)t_m$ |
| EHRDecM Time | $(1 + \|\mathcal{P}\| - \|\vec{\mathcal{R}}\|)(t_e + t_m) + 3t_p + t_m$ |
| EHRDecP Time | $(\|\mathcal{P}\|)(t_e + t_m) + 3t_p + t_m$ |

# 7. CONCLUSIONS

In this paper, we propose an anonymous RBAC scheme to secure identity privacy in EHR system. We achieve flexible access control, where EHR data can be encapsulated according to an on-demand access policy, while only the users whose role satisfies the access policy can decapsulate it. Patient privacy is preserved by using bilinear group with two subgroups, where one of the subgroups is used for blinding identities and the other one is used for key generation. Based on the decisional bilinear group assumptions, we prove that the proposed model has the property of both semantic security and anonymity. Besides, we provide an approach for anonymous search so that patient and his doctors can find out their own EHR in the anonymous system. To achieve better user experience, we apply "online/ offline" approach to speed up the data processing in the procedures of key generation and EHR encapsulation. Experimental results show that the online performance of our scheme reaches to millisecond-level.

Table 3: Comparison with Related Work

| | Anonymity | Order of bilinear group | Key Generation Time | Key Delegation Time | EHR Enc Time | Number of paring in EHR Dec |
|---|---|---|---|---|---|---|
| [20] | × | prime order | $(n+6)t_e+$<br>$(\lVert\vec{\mathcal{R}}\rVert+1)t_m$ | $(n+6)t_e+$<br>$(n+5)t_m$ | $(\lVert\mathcal{P}\rVert+4)t_e+$<br>$(\lVert\mathcal{P}\rVert+3)t_m+t_h$ | 2 |
| [27] | √ | composite order | $3\cdot(n+4)t_e+$<br>$(3\lVert\vec{\mathcal{R}}\rVert+4)t_m$ | $(25+6n-6\lVert\vec{\mathcal{R}}\rVert)t_e+$<br>$(18+4n-4\lVert\vec{\mathcal{R}}\rVert)t_m$ | $(\lVert\mathcal{P}\rVert+4)t_e+$<br>$(\lVert\mathcal{P}\rVert+4)t_m$ | 3 |
| Ours | √ | composite order | $3\cdot(n+5)t_e+$<br>$(3\lVert\vec{\mathcal{R}}\rVert+4)t_m$ | $(31+6n-6\lVert\vec{\mathcal{R}}\rVert)t_e+$<br>$(23+4n-4\lVert\vec{\mathcal{R}}\rVert)t_m$ | $(\lVert\mathcal{P}\rVert+5)t_e+$<br>$(\lVert\mathcal{P}\rVert+4)t_m$ | 3 |
| Ours & Improved | √ | composite order | $\lVert\vec{\mathcal{R}}\rVert\cdot t_m$ | $1\cdot t_m$ | $(\lVert\vec{\mathcal{P}}\rVert+1)t_m$ | 3 |



(a) Secret Key Generation Time (ms)

(b) Secret Key Delegation Time (ms)

(c) EHR Encapsulation Time (ms)

(d) EHR Decapsulation Time (ms)

(e) Improved KeyGen Time (ms)

(f) Improved Encapsulation Time (ms)

Figure 2: Experimental result for the Proposed System

## 8. ACKNOWLEDGMENTS

This paper is supported by National Key Basic Research Program (973 program) through the project 2012CB315905, by Natural Science Foundation through the projects 61272501, 61370190 and 61003214, by Fundamental Research Funds for the Central Universities through the project No.YWF-15-GJSYS-059, and by Beijing Natural Science Foundation through the project 4132056.

## 9. REFERENCES

[1] Health insurance portability and accountability act. U.S. Government Printing Office.

[2] Recommendations for the interpretation and application of the personal information protection and electronic documents act (s.c.2000, c.5) in the health research context. In *Institutes of Health Research*.

[3] J. A. Akinyele, C. U. Lehmann, M. D. Green, M. W. Pagano, Z. N. J. Peterson, and A. D. Rubin. Self-protecting electronic medical records using attribute-based encryption. Cryptology ePrint Archive, Report 2010/565, 2010. http://eprint.iacr.org/.

[4] M. Atallah, M. Blanton, N. Fazio, and K. Frikken. Dynamic and efficient key management for access hierarchies. *ACM Transactions on Information and System Security*, 12(3), 2009.

[5] M. Barua, X. Liang, R. Lu, and X. Shen. Peace: An efficient and secure patient-centric access control scheme for ehealth care system. In *INFOCOM WKSHPS '11*, pages 970–975. IEEE, 2011.

[6] D. Boneh, X. Boyen, and E. J. Goh. Hierarchical identity based encryption with constant size ciphertext. In *EUROCRYPT '05*, pages 440–456. Springer Berlin Heidelberg, 2005.

[7] D. Boneh, E. J. Goh, and K. Nissim. Evaluating 2-dnf formulas on ciphertexts. In *TCC '05*, pages 325–341. Springer.

[8] D. Boneh, E.-J. Goh, and K. Nissim. Conjunctive, subset, and range queries on encrypted data. In *TCC '07*, pages 535–554. Springer Berlin Heidelberg, 2007.

[9] D. Boneh, A. Sahai, and B. Watersn. Fully collusion resistant traitor tracing with short ciphertexts and private keys. In *EUROCRYPT '06*, pages 573–592. Springer Berlin Heidelberg, 2006.

[10] M. S. E. Ciampi. A view-based acces control model for ehr systems. In *Intelligent Distributed Computing VIII*, pages 443–452. Springer Berlin Heidelberg, 2015.

[11] M. S. Esposito. An access control model for easy management of patient privacy in ehr systems. In *Internet Technology and Secured Transactions'2013*, pages 463–470. IEEE, 2013.

[12] G. Fuchun and Y. Mu. Identity-based online/offline encryption. *Intel It*, 2012.

[13] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *ACM CCS'06*, pages 89–98. Proc of Acmccs, 2006.

[14] L. Guo, C. Zhang, J. Sun, and Y. Fang. Paas: A privacy-preserving attribute-based authentication system for ehealth networks. In *ICDCS '12*, pages 224–233. IEEE, 2012.

[15] S. Hohenberger and B. Waters. Online/offline attribute-based encryption. In *PKC '14*, pages 293–310. Springer Berlin Heidelberg, 2014.

[16] J. Huang, M. Sharaf, and C. T. Huang. A hierarchical framework for secure and scalable ehr sharing and access control in multi-cloud. In *ICPPW '12*, pages 279–287. IEEE, 2012.

[17] lakovidis I. Towards personal health record: Current situation, obstacles and trends in implementation of electronic healthcare record in europe. *International Journal of Medical Informatics*, 53(1-3):105–115, 1998.

[18] A. Lewko and B. Waters. New techniques for dual system encryption and fully secure hibe with short ciphertexts. In *TCC '10*, pages 455–479. Springer Berlin Heidelberg, 2010.

[19] A. Lewko and B. Waters. New proof methods for attribute-based encryption: Achieving full security through selective techniques. In *CRYPTO '12*, pages 180–198. Springer Berlin Heidelberg, 2012.

[20] W. Liu, X. Liu, J. Liu, Q. Wu, J. Zhang, and Y. Li. Auiting and revocation enabled role-based access contrl over outsourced private ehrs. In *HPCC '15*. IEEE, 2015.

[21] H. Löhr, A.-R. Sadeghi, and M. Winandy. Securing the e-health cloud. In *IHI '10*, pages 220–229. ACM, 2010.

[22] S. Narayan, M. Gagné, and R. Safavi-Naini. Privacy preserving ehr system using attribute-based infrastructure. In *CCSW '10*, pages 47–52. ACM, 2010.

[23] A. Ross. Technical perspective a chilly sense of security. In *ACM '09*, pages 90–90. Commun., 2009.

[24] L. Røstad and O. Nytrø. Personalized access control for a personally controlled health record. In *CSAW '08*, pages 9–16. ACM, 2008.

[25] S. Sabitha and M. Rajasree. Anonymous-cpabe: Privacy preserved content disclosure for data sharing in cloud. In *ARCS 2015*, pages 146–157. Springer International Publishing, 2015.

[26] J. Sedayao. Enhancing cloud security using data anonymization. *Cryptology*, 9(1):35–67, 1996.

[27] J. H. Seo, T. Kobayashi, M. Ohkubo, and K. Suzuki. Anonymous hierarchical identity-based encryption with constant size ciphertexts. In *PKC '09*, pages 215–234. Springer Berlin Heidelberg, 2009.

[28] E. Shimon and S. Micali. On-line/off-line digital signatures. *Cryptology*, 9(1):35–67, 1996.

[29] J. Sun and Y. Fang. Cross-domain data sharing in distributed electronic health record systems. *IEEE Transactions on Parallel and Distributed Systems*, 21(6):754–764, 2010.

[30] J. Sun, X. Zhu, C. Zhang, and Y. Fang. Hcpp: Cryptography based secure ehr system for patient privacy and emergency healthcare. In *ICDCS '11*, pages 373–382. IEEE, 2011.

[31] J. Taeho, X. Li, Z. Wan, and W. Meng. Control cloud data access privilege and anonymity with fully anonymous attribute-based encryption. In *Transaction '15*, pages 190–199. IEEE, 2015.

# APPENDIX

## A. PROOF OF SECURITY

In this section, we provide concrete proof steps for Theorem 1 and Theorem 2. We extend the Anonymous Hierarchical Identity-Based Encryption technique introduced by Seo [27] to our security proof.

**Indistinguishability between Game$_1$ and Game$_2$.**
*Proof of Lemma* 1. Assume that there exists an adversary $\mathcal{A}$ that can distinguish between Game$_1$ and Game$_2$ with advantage $\epsilon_1$. Then there is an simulator $\mathcal{B}$ that can solve the $(n+2)$-BDHE problem with the same advantage $\epsilon_1$. The input of simulator $\mathcal{B}$ is the challenge tuple $(D_1, T_1)$ of the decision BDHE problem with

$$D_1 = \begin{pmatrix} (N, \mathbb{G}, \mathbb{G}_T, e), g_p, g_q, h, g_p^{a}, g_p^{a^2}, \cdots, g_p^{a^{n+1}}, \\ g_p^{a^{n+3}}, \cdots, g_p^{a^{2n+4}} \end{pmatrix}$$

Simulator $\mathcal{B}$ needs to decide whether $T_1 = e(g_p, h)^{a^{n+2}}$ or $T_1 \xleftarrow{R} \mathbb{G}_{T,p}$. Let $A_i = g_p^{a^i}$ for $1 \le i \le 2n+4$.

**Init**. Adversary $\mathcal{A}$ outputs an access policy $\mathcal{P}$ containing roles for medical staffs that $\mathcal{A}$ may decide to be challenged. Also, adversary $\mathcal{A}$ outputs an identity $Id$ for a patient that it may decide to be challenged. We denote the challenge roles as $\{\vec{\mathcal{R}}_i \in \mathcal{P}\}$, the challenge atom roles as $\{\mathcal{R}_i \in S_{\mathcal{P}}\}$, and $\mathbb{I} = \{i : \mathcal{R}_i \in S_{\mathcal{P}}\}$. $\mathcal{A}$ sends $\mathcal{P}$ and $Id$ to simulator $\mathcal{B}$.

**Setup**. To generate the system parameter $PK$, simulator $\mathcal{B}$ requests an instance of the Decision $(n+2)$-BDHE assumption. $\mathcal{B}$ chooses random integers $\gamma, x, y, z, \{x_i\}_{i \in [1,n+1]} \xleftarrow{R} \mathbb{Z}_N$ and random elements $R_g, R_f, R_u, R_h, \{R_{h_i}\}_{i \in [1,n]} \xleftarrow{R} \mathbb{G}_q$. Then it sets

$$E = e(A_1, A_{n+1} \cdot g_p^{\gamma}), \quad G = g_p \cdot R_g, \quad F = g_p^{z} \cdot R_f$$

$$U = (g_p^{y} \cdot A_{n+1}^{Id} \cdot \prod_{i \in \mathbb{I}} (A_{n-i+1})^{\mathcal{R}_i}) R_u$$

$$H = (g_p^{x_{n+1}} / A_{n+1}) \cdot R_h, \ \{H_i = (g_p^{x_i} / A_{n-i+1}) R_{h,i}\}_{i \in [1,n]}$$

Finally, $\mathcal{B}$ gives the system parameters

$$PK = \{g_p, g_q, G, F, U, H, \{H_i\}_{i \in [1,n]}, E\}$$

to the adversary $\mathcal{A}$. The master key $\omega$ corresponding to the system parameters is $(A_{n+1} \cdot g_p^{\gamma})^a = A_{n+2} \cdot A_1^{\gamma}$. Note that $\mathcal{B}$ does not know the master key as it does not have $A_{n+2}$.

**Phase 1**. Adversary $\mathcal{A}$ can adaptively issue secret key query for medical staff with role $\vec{\mathcal{R}}^\star$ and secret key query for a patient with identity $Id^\star$.

**I**. When $\mathcal{A}$ issues secret key query for a medical staff with role $\vec{\mathcal{R}}^\star$, the only restriction is that $\vec{\mathcal{R}}^\star \notin Pref(\mathcal{P})$. This ensures $\vec{\mathcal{R}}^\star$ contains at least one atom role $\mathcal{R}_k^\star \in S_{\vec{\mathcal{R}}^\star}$ such that $\mathcal{R}_k^\star \notin S_{\vec{\mathcal{R}}}$, where $k \leq n$. Let $k$ be the smallest index satisfying this condition. To response the query, simulator $\mathcal{B}$ first generates secret key for the medical staff with role $\vec{\mathcal{R}}_k^\star = (\mathcal{R}_1^\star, \cdots, \mathcal{R}_k^\star)$, from which $\mathcal{B}$ can then derive $\mathsf{KeyDelegM}$ algorithm for $\vec{\mathcal{R}}^\star$. Denote $I^\star = \{i : \mathcal{R}_i^\star \in S_{\vec{\mathcal{R}}_k^\star}\}$. $\mathcal{B}$ randomly choose integers $r_1, r_2 \in \mathbb{Z}_N$ to compute $SK^{\vec{\mathcal{R}}^\star} = \left\{SK_d^{\vec{\mathcal{R}}^\star}, SK_r^{\vec{\mathcal{R}}^\star}\right\}$. For $SK_d^{\vec{\mathcal{R}}^\star}$, we have

$$\left\{\omega\left(u \prod_{i \in I^\star} h_i^{\mathcal{R}_i^\star}\right)^{\hat{r}_1} f^{r_2}, g^{\hat{r}_1}, g^{r_2}, g_h^{\hat{r}_1}, \{h_j^{\hat{r}_1}\}_{j \in [1,n] \setminus I^\star}\right\}$$

where $\hat{r}_1 = r_1 + \frac{a^{k+1}}{\mathcal{R}_k^\star - \mathcal{R}_k}$. We observe the first component:

$$\omega(u \prod_{i \in I^\star} h_i^{\mathcal{R}_i^\star})^{\hat{r}_1} f^{r_2} = \omega(u \prod_{i \in I^\star} h_i^{\mathcal{R}_i^\star})^{r_1} f^{r_2} (u \prod_{i \in I^\star} h_i^{\mathcal{R}_i^\star})^{\frac{a^{k+1}}{\mathcal{R}_k^\star - \mathcal{R}_k}}$$

Since $u$ is $g_p^y A_{n+1}^{Id} \prod_{i \in \mathbb{I}} (A_{n-i+1})^{\mathcal{R}_i}$, $f$ is $g_p^z$, and $h_i$ is $g_p^{x_i} / A_{n-i+1}$ which can be obtained by removing $R_u$, $R_f$, $R_{h,i}$ from $U$, $F$ and $H_i$ respectively, and $r_1$, $r_2$ are known to $\mathcal{B}$, so the key point for simulator $\mathcal{B}$ is to compute $\omega \cdot (u \prod_{i \in I^\star} h_i^{\mathcal{R}_i^\star})^{\frac{a^{k+1}}{\mathcal{R}_k^\star - \mathcal{R}_k}}$. It equals to

$$A_{n+2} \cdot A_1^{\gamma} \cdot \begin{pmatrix} g_p^y \cdot A_{n+1}^{Id} \cdot \prod_{i \in \mathbb{I}} (A_{n-i+1})^{\mathcal{R}_i} \cdot \\ \prod_{i \in I^\star} (g_p^{x_i} / A_{n-i+1})^{\mathcal{R}_i^\star} \end{pmatrix}^{\frac{a^{k+1}}{\mathcal{R}_k^\star - \mathcal{R}_k}}$$

$$= A_1^{\gamma} \cdot \begin{pmatrix} A_{k+1}^y \cdot A_{n+k+2}^{Id} \cdot \\ \prod_{i \in \mathbb{I}, i \notin I^\star} (A_{n+k-i+2})^{\mathcal{R}_i} \cdot \prod_{i \in I^\star} A_{k+1}^{x_i \mathcal{R}_i^\star} \end{pmatrix}^{\frac{1}{\mathcal{R}_k^\star - \mathcal{R}_k}}$$

where all the terms in the above expression do not involve $A_{n+2}$ and not exceed $A_{2n+4}$, so $\mathcal{B}$ can compute it. Similar, the remaining components can be be computed by $\mathcal{B}$ due to they do not involve $A_{n+2}$. Since $\hat{r}_1 = r_1 + \frac{a^{k+1}}{\mathcal{R}_k^\star - \mathcal{R}_k}$ and $r_2$ are uniformly and independently distributed in $\mathbb{Z}_N$, $SK_d^{\vec{\mathcal{R}}^\star}$ has the same distribution as that of the actual key distribution.

To generate $SK_r^{\vec{\mathcal{R}}^\star}$, $\mathcal{B}$ randomly choose $s_1, s_2, t_1, t_2 \in \mathbb{Z}_N$. It is easier for $\mathcal{B}$ to compute $SK_r^{\vec{\mathcal{R}}^\star}$ than $SK_d^{\vec{\mathcal{R}}^\star}$ due to there is no component associate with $\omega$ in $SK_r^{\vec{\mathcal{R}}^\star}$. The only restriction is that $s_1, s_2, t_1, t_2$ must satisfy equation $s_1 \cdot t_2 - s_2 \cdot t_1 \neq 0$ mod $p$ and mod $q$. $\mathcal{B}$ picks up $s_1, s_2, t_1, t_2$ until they satisfy the equation. The iteration will finish as soon as the equation holds without probability $\frac{p+q-1}{N}$. Therefore $SK^{\vec{\mathcal{R}}^\star}$ has the same distribution as that of the actual key distribution.

**II**.When $\mathcal{A}$ issues secret key query for a patient with identity $Id^\star$, the restriction is that $Id^\star \neq Id$. To response the query, $\mathcal{B}$ randomly choose $r_1', r_2' \in \mathbb{Z}_N$. Then it computes

$$SK^{Id^\star} = \left(\omega(ug_h^{Id^\star})^{\hat{r}_1'} f^{r_2'}, g^{\hat{r}_1'}, g^{r_2'}, \{h_j^{\hat{r}_1'}\}_{j \in [1,n]}\right)$$

where $\hat{r}_1' = \frac{a}{Id^\star - Id} + r_1'$. We observe the first component of $SK^{Id^\star}$:

$$\omega(ug_h^{Id^\star})^{\hat{r_1}'} f^{r_2'} = \omega \cdot (u \cdot g_h^{Id^\star})^{r_1'} f^{r_2'} \cdot (ug_h^{Id^\star})^{\frac{a}{Id^\star - Id}}$$

Since $u, g_h, f$ can be obtained by removing $R_u, R_h, R_f$ from $U, H, F$ respectively, and $r_1', r_2'$ are chosen by simulator itself, the key point for $\mathcal{B}$ is to compute $\omega(u \cdot g_h^{Id^\star})^{\frac{a}{Id^\star - Id}}$. It equals

$$A_{n+2} A_1^{\gamma} \cdot \begin{pmatrix} g_p^y \cdot A_{n+1}^{Id} \cdot \prod_{i \in \mathbb{I}} (A_{n-i+1})^{\mathcal{R}_i} \cdot \\ (g_p^{x_{n+1}} / A_{n+1})^{Id^\star} \end{pmatrix}^{\frac{a}{Id^\star - Id}}$$

$$= A_1^{\gamma} \cdot (A_1^y \cdot \prod_{i \in \mathbb{I}} (A_{n-i+2})^{\mathcal{R}_i} \cdot A_1^{x_{n+1} Id^\star})^{\frac{1}{Id^\star - Id}}$$

which can be computed by $\mathcal{B}$ since it knows all terms involved in above expression. All the other components in $SK^{Id^\star}$ can also be computed by $\mathcal{B}$ since they do not involve $A_{n+2}$. Finally, simulator $\mathcal{B}$ generates secret key for $Id^\star$ and responds it to adversary $\mathcal{A}$.

**Challenge**. Adversary $\mathcal{A}$ outputs two equal-length EHRs $EHR_0, EHR_1$. Simulator $\mathcal{B}$ chooses random elements $Q_1, Q_2$ and $Q_3$ from $\mathbb{G}_q$, picks a random coin $b \in \{0, 1\}$ and returns the challenge ciphertext

$$(\mathrm{Hdr}, En) = (C_1, C_2, C_3, En)$$
$$= \begin{pmatrix} h \cdot Q_1, h^z \cdot Q_2, h^{y + x_{n+1} \cdot Id + \sum_{i \in \mathbb{I}} \mathcal{R}_i \cdot x_i} \cdot Q_3, \\ SymEnc(T_1 \cdot e(A_1, h^{\gamma}), EHR_b) \end{pmatrix}$$

where $h$ and $T_1$ are given from challenge tuple $(D_1, T_1)$. We consider $h$ as $g_p^c$ for some unknown $c \in \mathbb{Z}_N$. Observe each component in the challenge ciphertext and we have

$$C_1 = g_p^c \cdot Q_1 = G^c \cdot Q_1', \; C_2 = (g_p^z)^c \cdot Q_2 = F^c \cdot Q_2'$$
$$C_3 = (g_p^{y + x_{n+1} \cdot Id + \sum_{i \in \mathbb{I}} \mathcal{R}_i \cdot x_i})^c \cdot Q_3$$
$$= \left(g_p^y \cdot A_{n+1}^{Id} \cdot H^{Id} \cdot \prod_{i \in \mathbb{I}} H_i^{\mathcal{R}_i} A_{n-i+1}^{\mathcal{R}_i}\right)^c \cdot Q_3'$$
$$= \left(U \cdot H^{Id} \cdot \prod_{i \in \mathbb{I}} H_i^{R_i}\right)^c \cdot Q_3'$$

For the component $SymEnc(T_1 \cdot e(A_1, h^{\gamma}), EHR_b)$, if $T_1 = e(g_p, h)^{a^{n+2}}$, then

$$T_1 \cdot e(A_1, h^{\gamma}) = e(g_p, g_p^c)^{a^{n+2}} \cdot e(g_p^a, g_p^{c\gamma})$$
$$= e(g_p^a, g_p^{a^{n+1}})^c \cdot e(g_p^a, g_p^{\gamma})^c$$
$$= e(A_1, A_{n+1} \cdot g_p^{\gamma})^c = E^c$$

Therefore it is a valid session key for $EHR_b$ and $(\mathrm{Hdr}, En)$ is a valid ciphertext of $\mathrm{Game}_1$. Otherwise, $EHR_b$ is encapsulated by a random element in $\mathbb{G}_{T,p}$ and $SymEnc(T_1 \cdot e(A_1, h^{\gamma}), EHR)$ is random from the adversarial perspective. In that case, $(\mathrm{Hdr}, En)$ is a ciphertext of $\mathrm{Game}_2$.

**Phase 2**. Repeat Phase 1.

**Guess**. Adversary $\mathcal{A}$ outputs a guess that it is in Game$_1$ or Game$_2$. Simulator $\mathcal{B}$ guess $T_1 = e(g_p, h)^{a^{n+2}}$ if $\mathcal{A}$ decides it is in Game$_1$. Otherwise $\mathcal{B}$ outputs $T_1 \xleftarrow{R} \mathbb{G}_{T,p}$. If $\mathcal{A}$ has the advantage $\epsilon_1$ to distinguish Game$_1$ and Game$_2$, $\mathcal{B}$ can solve the Decision $(n+2)$-BDHE problem with the same advantage $\epsilon_1$, which completes the proof of Lemma 1.

**Indistinguishability between Game$_2$ and Game$_3$.**

*ProofofLemma* 2. Assume that there exists an adversary $\mathcal{A}$ that can distinguish between Game$_2$ and Game$_3$ with advantage $\epsilon_2$. Then there is an simulator $\mathcal{B}$ that can solve the BSD problem with the same advantage $\epsilon_2$. The input of simulator $\mathcal{B}$ is the challenge tuple $(D_2, T_2)$ of the BSD assumption with

$$D_2 = (N, \mathbb{G}, \mathbb{G}_T, e)$$

$\mathcal{B}$ needs to decide whether $T_2 \xleftarrow{R} \mathbb{G}_{T,p}$ or $T_2 \xleftarrow{R} \mathbb{G}_T$.

**Init**. $\mathcal{A}$ send an access policy $\mathcal{P}$ containing roles for medical staffs and an identity $Id$ for a patient to $\mathcal{B}$.

**Setup**. $\mathcal{B}$ chooses all random elements from $\mathbb{G}_p$ and $\mathbb{G}_q$ to generate system parameters as the actual setup algorithm.

**Phase 1**. Adversary $\mathcal{A}$ can adaptively issue secret key query for medical staff with role $\vec{\mathcal{R}}^\star$ and secret key query for a patient with identity $Id^\star$. $\mathcal{B}$ responds to queries as the actual key generation algorithm.

**Challenge**. Adversary $\mathcal{A}$ outputs two equal-length EHRs $EHR_0, EHR_1$. Simulator $\mathcal{B}$ picks a random coin $b \in \{0,1\}$ and outputs a normal ciphertext with the exception that the $EHR_b$ is encapsulated as $En = SymEnc(T_2, EHR_b)$. It is a normal encapsulated EHR of Game$_2$ if $T_2 \xleftarrow{R} \mathbb{G}_{T,p}$. Otherwise $T_2 \xleftarrow{R} \mathbb{G}_T$ and $En$ is an encapsulated EHR of Game$_3$.

**Phase 2**. Repeat Phase 1.

**Guess**. Adversary $\mathcal{A}$ outputs a guess that it is in Game$_2$ or Game$_3$. Simulator $\mathcal{B}$ guess $T_2 \xleftarrow{R} \mathbb{G}_{T,p}$ if $\mathcal{A}$ decides it is in Game$_2$. Otherwise $\mathcal{B}$ outputs $T_2 \xleftarrow{R} \mathbb{G}_T$. If $\mathcal{A}$ has the advantage $\epsilon_2$ to distinguish Game$_2$ and Game$_3$, $\mathcal{B}$ can solve the Decision BSD problem with the same advantage $\epsilon_2$, which completes the proof of Lemma 2.

**Indistinguishability between Game$_3$ and Game$_4$.**

*ProofofLemma* 3. Assume that there exists an adversary $\mathcal{A}$ that can distinguish between Game$_3$ and Game$_4$ with advantage $\epsilon_3'$. Then there is an simulator $\mathcal{B}$ that can solve the $(n+1)$-cDH problem with the same advantage $\epsilon_3' \cdot (1 - \frac{p+q-1}{N})^{q_s}$. The input of simulator $\mathcal{B}$ is the challenge tuple $(D_3, T_3)$ of the decision cDH problem with

$$D_3 = \begin{pmatrix} (N, \mathbb{G}, \mathbb{G}_T, e), g_p, g_q, g_p^{a}, g_p^{a^2}, \cdots, g_p^{a^n}, g_p^{a^{n+1}}, \\ g_p^{a^{n+2}} \cdot R_1, g_p^{a^{n+2} \cdot b} \cdot R_2 \end{pmatrix}$$

Simulator $\mathcal{B}$ needs to decide whether $T_3 = g_p^b \cdot R_3$ or $T_3 \xleftarrow{R} \mathbb{G}$. Let $A_i = g_p^{a^i}, B = A_{n+2} \cdot R_1'$ and $C = A_{n+2}^b \cdot R_2'$ where $g_p^{a^i}, R_1'$ and $R_2'$ are defined in $D_3$ for $1 \le i \le n+2$.

**Init**. Adversary $\mathcal{A}$ outputs an access policy $\mathcal{P}$ containing roles for medical staffs that $\mathcal{A}$ may decide to be challenged. Also, algorithm $\mathcal{A}$ outputs an identity $Id$ for a patient that it may decide to be challenged. We denote the challenge roles as $\{\vec{\mathcal{R}}_i \in \mathcal{P}\}$, the challenge atom roles as $\{\mathcal{R}_i \in S_{\mathcal{P}}\}$, and $\mathbb{I} = \{i : \mathcal{R}_i \in S_{\mathcal{P}}\}$. $\mathcal{A}$ sends $\mathcal{P}$ and $Id$ to simulator $\mathcal{B}$.

**Setup**. To generate system parameter $PK$, simulator $\mathcal{B}$ requests an instance of the Decision $(n+1)$-cDH assumption. $\mathcal{B}$ choose random integers $\gamma, x, y, z, \{x_i\}_{i \in [1, n+1]} \xleftarrow{R} \mathbb{Z}_N$, $\omega \xleftarrow{R} \mathbb{G}_p$ and random elements $R_g, R_f, R_u, R_h, \{R_{h_i}\}_{i \in [1,n]} \xleftarrow{R} \mathbb{G}_q$. Then it sets

$$E = e(B^x, \omega), \quad G = B^x \cdot R_g, \quad F = g_p^z \cdot R_f$$

$$H = A_{n+1}^{x_{n+1}} \cdot R_h, \{H_i = A_i^{x_i} R_{h,i}\}_{i \in [1,n]}$$

$$U = (g_p^y / (H^{Id} \cdot \prod_{i \in \mathbb{I}} {H_i}^{\mathcal{R}_i})) R_u$$

Finally, $\mathcal{B}$ gives the system parameters

$$PK = \{g_p, g_q, G, F, U, H, \{H_i\}_{i \in [1,n]}, E\}$$

to the adversary $\mathcal{A}$.

**Phase 1**. Adversary $\mathcal{A}$ can adaptively issue secret key query for medical staff with role $\vec{\mathcal{R}}^\star$ and secret key query for a patient with identity $Id^\star$.

**I**. When $\mathcal{A}$ issues secret key query for a medical staff with role $\vec{\mathcal{R}}^\star$, we also have the restriction that $\vec{\mathcal{R}}^\star \notin Pref(\mathcal{P})$. This ensures $\vec{\mathcal{R}}^\star$ contains at least one atom role $\mathcal{R}_k^\star \in S_{\vec{\mathcal{R}}^\star}$ such that $\mathcal{R}_k^\star \notin S_{\vec{\mathcal{R}}}$, where $k \le n$. Let $k$ be the smallest index satisfying this condition. To response the query, simulator $\mathcal{B}$ first generates secret key for the medical staff with role $\vec{\mathcal{R}}_k^\star = (\mathcal{R}_1^\star, \cdots, \mathcal{R}_k^\star)$, from which $\mathcal{B}$ can then derive KeyDelegM algorithm for $\vec{\mathcal{R}}^\star$. Denote $I^\star = \{i : \mathcal{R}_i^\star \in S_{\vec{\mathcal{R}}_k^\star}\}$. $\mathcal{B}$ randomly choose integers $r_1, r_2 \in \mathbb{Z}_N$ to compute $SK^{\vec{\mathcal{R}}} = \left\{SK_d^{\vec{\mathcal{R}}^\star}, SK_r^{\vec{\mathcal{R}}^\star}\right\}$, where we posit $\hat{r}_1 = \frac{z}{a^k} r_1 + \frac{z}{a^{k+1}} r_2, \hat{r}_2 = -\frac{y}{a^k} r_1 - (\frac{x_k(\mathcal{R}_k^\star - \mathcal{R}_k)}{a} + \frac{y}{a^{k+1}}) r_2$. For the first component of $SK_d^{\vec{\mathcal{R}}^\star}$, since $u, f$ and $h_i$ can be obtained by removing blind factors, it can be rewritten as

$$\omega((g_p^y / A_{n+1}^{x_{n+1} \cdot Id} \cdot \prod_{i \in \mathbb{I}} A_i^{x_i \mathcal{R}_i}) \cdot \prod_{i \in I^\star} A_i^{x_i \cdot \mathcal{R}_i^\star})^{\hat{r}_1} g_p^{z \cdot \hat{r}_2}$$

We focus on the exponent of $g_p$ in above expression and have

$$(y - a^{n+1} x_{n+1} Id - \sum_{i \in \mathbb{I}} a^i x_i \mathcal{R}_i + \sum_{i \in I^\star} a^i x_i \mathcal{R}_i^\star) \hat{r}_1 + z \hat{r}_2$$

$$= \begin{pmatrix} y + a^k x_k (\mathcal{R}_k^\star - \mathcal{R}_k) - \\ \sum\limits_{i \in \mathbb{I}, i \notin I^\star} a^i x_i \mathcal{R}_i - a^{n+1} x_{n+1} Id \end{pmatrix} \hat{r}_1 + z \hat{r}_2$$

$$= \begin{pmatrix} x_k (\mathcal{R}_k^\star - \mathcal{R}_k) - \\ \sum\limits_{i \in \mathbb{I}, i \notin I^\star} a^{i-k} x_i \mathcal{R}_i - a^{n-k+1} x_{n+1} Id \end{pmatrix} z \cdot r_1 +$$

$$\left( \sum_{i \in \mathbb{I}, i \notin I^\star} a^{i-k-1} x_i \mathcal{R}_i + a^{n-k} x_{n+1} Id \right) z \cdot r_2$$

All the terms in above expression are not associated with $A_{n+2}$. For the the index $i$, it satisfies $i \in \mathbb{I}, i \notin I^\star$ ($I^\star = \{i : \mathcal{R}_i^\star \in S_{\vec{\mathcal{R}}_k^\star}\}$) and $k$ is the smallest index satisfying $\mathcal{R}_k^\star \in S_{\vec{\mathcal{R}}^\star}$, so we can get $i > k$. Hence, above expressions can be computed by $\mathcal{B}$. The remaining components in $SK_d^{\vec{\mathcal{R}}^\star}$ can be computed by $\mathcal{B}$ as well due to they are not associated with $A_{n+2}$.

Next, $\mathcal{B}$ generates $SK_r^{\vec{\mathcal{R}}^\star}$ in a similar manner to generating $SK_d^{\vec{\mathcal{R}}^\star}$. The details of this procedure are highly similar to those of $SK_d^{\vec{\mathcal{R}}^\star}$, so they are skipped. We just highlight the chosen parameters here. $\mathcal{B}$ randomly choose $s_1, s_2, t_1, t_2 \in \mathbb{Z}_N$ and let

$$\hat{s}_1 = \frac{z}{a^k}s_1 + \frac{z}{a^{k+1}}s_2,\ \hat{t}_1 = \frac{z}{a^k}t_1 + \frac{z}{a^{k+1}}t_2$$
$$\hat{s}_2 = -\frac{y}{a^k}s_1 - (\frac{x_k(\mathcal{R}_k^\star - \mathcal{R}_k)}{a} + \frac{y}{a^{k+1}})s_2$$
$$\hat{t}_2 = -\frac{y}{a^k}t_1 - (\frac{x_k(\mathcal{R}_k^\star - \mathcal{R}_k)}{a} + \frac{y}{a^{k+1}})t_2$$

$\hat{s}_1, \hat{s}_2, \hat{t}_1, \hat{t}_2$ must satisfy equation $\hat{s}_1 \cdot \hat{t}_2 - \hat{s}_2 \cdot \hat{t}_1 \neq 0$ mod $p$ and mod $q$ with probability $1 - \frac{p+q-1}{N}$. Therefore $SK^{\vec{\mathcal{R}}^\star}$ has the same distribution and structure as that of the actual key distribution with probability $1 - \frac{p+q-1}{N}$.

**II**.When $\mathcal{A}$ issues secret key query for a patient with identity $Id^\star$, the restriction is that $Id^\star \neq Id$. To response the query, $\mathcal{B}$ randomly choose $\hat{r}'_1, \hat{r}'_2 \in \mathbb{Z}_N$ and let $\hat{r}'_1 = \frac{z}{a}r'_1 + \frac{z}{a}r'_2$ and $\hat{r}'_2 = -\frac{y}{a}r'_1 - \frac{y}{a}r'_2$. Then it computes

$$SK^{Id^\star} = \left(\omega(ug_h^{Id^\star})^{\hat{r}'_1} f^{\hat{r}'_2}, g^{\hat{r}'_1}, g^{\hat{r}'_2}, \{h_j^{\hat{r}'_1}\}_{j\in[1,n]}\right)$$

As $u, g_h, f$ can be obtained by removing $R_u, R_h, R_f$ from $U, H, F$ respectively, the first component can be rewritten as $\omega((g_p^y/A_{n+1}^{x_{n+1}\cdot Id} \cdot \prod_{i\in\mathbb{I}} A_i^{x_i\mathcal{R}_i}) \cdot A_{n+1}^{x_{n+1}Id^\star})^{\hat{r}_1} \cdot g_p^{z\hat{r}'_2}$.

We focus on the exponent of $g_p^z$ and get

$$(y - a^{n+1}x_{n+1}Id - \sum_{i\in\mathbb{I}} a^i x_i\mathcal{R}_i + a^{n+1}x_{n+1}Id^\star)\hat{r}'_1 + z\hat{r}'_2$$
$$= \left(x_{n+1}a^n(Id^\star - Id) - \sum_{i\in\mathbb{I}} a^{i-1}x_i\mathcal{R}_i\right) z \cdot r'_1 +$$
$$\left(x_{n+1}a^n(Id^\star - Id) - \sum_{i\in\mathbb{I}} a^{i-1}x_i\mathcal{R}_i\right) z \cdot r'_2$$

which can be computed by $\mathcal{B}$ since it knows all terms involved in above expression. All the other components in $SK^{Id^\star}$ can also be computed by $\mathcal{B}$ since they do not involve $A_{n+2}$. Finally, simulator $\mathcal{B}$ generates secret key for $Id^\star$ and responds it to adversary $\mathcal{A}$.

**Challenge**. Adversary $\mathcal{A}$ outputs two equal-length EHRs $EHR_0, EHR_1$. Simulator $\mathcal{B}$ ignores them and selects random element $R_{En}$ from $\mathbb{G}_T$. $\mathcal{B}$ also selects random elements $Q_1, Q_2, Q_3$ from $\mathbb{G}_q$. Then $\mathcal{B}$ sends the challenge ciphertext to adversary $\mathcal{A}$

$$\begin{aligned}(\text{Hdr}, En) &= (C_1, C_2, C_3, R_{En}) \\ &= (C^x Q_1, {T_3}^z Q_2, {T_3}^y Q_3, R_{En})\end{aligned}$$

where $T_3$ are given from challenge tuple $(D_3, T_3)$. Observe each component in the challenge ciphertext and we have $C_1 = (A_{n+2}^b R'_2)^x Q_1 = G^b Q'_1$. If $T_3 = g_p^b \cdot R_3$, then

$$C_2 = (g_p^b R_3)^z Q_2 = F^b Q'_2$$
$$C_3 = (g_p^b R_3)^y Q_3 = (UH^{Id}\prod_{i\in\mathbb{I}} H_i^{\mathcal{R}_i})^b Q'_3$$

If $T_3 \overset{R}{\leftarrow} \mathbb{G}$, $T_3$ can be written as $g_p^r \widetilde{R}_3$ where r is random integer chosen from $\mathbb{Z}_N$ and $\widetilde{R}_3$ is a random element chosen from $\mathbb{G}_q$. Then

$$C_2 = (g_p^r \widetilde{R_3})^z Q_2 = F^r Q'_2$$
$$C_3 = (g_p^r \widetilde{R_3})^y Q_3 = (UH^{Id}\prod_{i\in\mathbb{I}} H_i^{\mathcal{R}_i})^r Q'_3$$

In this case, $C_2$ and $C_3$ share the same random $r$, while $C_1$ uses random $b$ independent from $r$. Both $b$ and $r$ first appear in ciphertext from adversarial viewpoint and they are uniformly and independently chosen from $\mathbb{Z}_N$. Hence, $C_1$ is a random element from the adversarial viewpoint and $(\text{Hdr}, En)$ is the ciphertext of Game$_4$.

**Phase 2**. Repeat Phase 1.

**Guess**. Adversary $\mathcal{A}$ outputs a guess. i.e. if $\mathcal{A}$ decides to output 0 (Game$_3$ ), simulator $\mathcal{B}$ output $T_3 = g_p^b \cdot R_3$; if $\mathcal{A}$ decides to output 1 (Game$_4$), $\mathcal{B}$ output $T_3 \overset{R}{\leftarrow} \mathbb{G}$. If $\mathcal{A}$ has the advantage $\epsilon'_3$ to distinguish Game$_3$ and Game$_4$, $\mathcal{B}$ can solve the Decision cDH problem with the advantage $\epsilon'_3 \cdot (1 - \frac{p+q-1}{N})^{q_s}$, which completes the proof of Lemma 3.

**Indistinguishability between Game$_4$ and Game$_5$.**

*ProofofLemma* 4. Assume that there exists an adversary $\mathcal{A}$ that can distinguish between Game$_4$ and Game$_5$ with advantage $\epsilon'_4$. Then there is an simulator $\mathcal{B}$ that can solve the $(n+1)$-cDHE problem with the same advantage $\epsilon'_4 \cdot (1 - \frac{p+q-1}{N})^{q_s}$. The input of simulator $\mathcal{B}$ is the challenge tuple $(D_4, T_4)$ of the decision cDHE problem with

$$D_4 \leftarrow \begin{pmatrix} (N, \mathbb{G}, \mathbb{G}_T, e), g_p, g_q, g_p^a, g_p^{a^2}, \cdots, g_p^{a^n}, g_p^{a^{n+1}}, \\ g_p^{a^{n+2}} \cdot R_1, g_p^{a^{n+2}\cdot b} \cdot R_2, g_p^{a^{n+3}}, \cdots, g_p^{a^{2n+2}}, \end{pmatrix}$$

Simulator $\mathcal{B}$ needs to decide whether $T_4 = g_p^b \cdot R_3$ or $T_4 \overset{R}{\leftarrow} \mathbb{G}$. Let $A_i = g_p^{a^i}$, $B = A_{n+2} \cdot R'_1$ and $C = A_{n+2}^b \cdot R'_2$ where $g_p^{a^i}$, $R'_1$ and $R'_2$ are defined in $D_4$ for $1 \le i \le 2n+2$.

**Init**. Adversary $\mathcal{A}$ outputs an access policy $\mathcal{P}$ containing roles for medical staff that $\mathcal{A}$ may decide to be challenged. Also, algorithm $\mathcal{A}$ outputs an identity $Id$ for a patient that it may decide to be challenged. We denote the challenge roles as $\{\vec{\mathcal{R}}_i \in \mathcal{P}\}$, the challenge atom roles as $\{\mathcal{R}_i \in S_\mathcal{P}\}$, and $\mathbb{I} = \{i : \mathcal{R}_i \in S_\mathcal{P}\}$. $\mathcal{A}$ sends $\mathcal{P}$ and $Id$ to simulator $\mathcal{B}$.

**Setup**. To generate system parameter $PK$, simulator $\mathcal{B}$ requests an instance of the Decision $(n+1)$-cDHE assumption. $\mathcal{B}$ chooses random integers $\gamma, x, y, z, \{x_i\}_{i\in[1,n+1]} \overset{R}{\leftarrow} \mathbb{Z}_N$, $\omega \overset{R}{\leftarrow} \mathbb{G}_p$ and random elements $R_g, R_f, R_u, R_h, \{R_{h_i}\}_{i\in[1,n]} \overset{R}{\leftarrow} \mathbb{G}_q$. Then it sets

$$E = e(g_p^x, \omega), \quad G = g_p^x \cdot R_g, \quad F = B^z \cdot R_f$$
$$H = A_{n+1}^{x_{n+1}} \cdot R_h, \{H_i = A_{n+1-i}^{x_i} R_{h,i}\}_{i\in[1,n]}$$
$$U = (g_p^y/(H^{Id} \cdot \prod_{i\in\mathbb{I}} {H_i}^{\mathcal{R}_i}))R_u$$

$\mathcal{B}$ gives $PK = \{g_p, g_q, G, F, U, H, \{H_i\}_{i\in[1,n]}, E\}$ to $\mathcal{A}$.

**Phase 1**. Adversary $\mathcal{A}$ can adaptively issue secret key query for medical staff with role $\vec{\mathcal{R}}^\star$ and secret key query for a patient with identity $Id^\star$.

**I**. When $\mathcal{A}$ issues secret key query for a medical staff with role $\vec{\mathcal{R}}^\star$, we also have the restriction that $\vec{\mathcal{R}}^\star \notin Pref(\mathcal{P})$. This ensures $\vec{\mathcal{R}}^\star$ contains at least one atom role $\mathcal{R}_k^\star \in S_{\vec{\mathcal{R}}^\star}$

such that $\mathcal{R}_k^\star \notin S_{\vec{\mathcal{R}}}$, where $k \leq n$. Let $k$ be the smallest index satisfying this condition. To response the query, simulator $\mathcal{B}$ first generates secret key for the medical staff with role $\vec{\mathcal{R}}_k^\star = (\mathcal{R}_1^\star, \cdots, \mathcal{R}_k^\star)$, from which $\mathcal{B}$ can then derive $\mathsf{KeyDelegM}$ algorithm for $\vec{\mathcal{R}}^\star$. Denote $I^\star = \{i : \mathcal{R}_i^\star \in S_{\vec{\mathcal{R}}_k^\star}\}$. $\mathcal{B}$ randomly choose integers $r_1, r_2 \in \mathbb{Z}_N$ to compute $SK^{\vec{\mathcal{R}}} = \left\{SK_d^{\vec{\mathcal{R}}^\star}, SK_r^{\vec{\mathcal{R}}^\star}\right\}$, where we posit $\hat{r}_1 = a^{k+1}zr_1 + r_2, \hat{r}_2 = -x_k(\mathcal{R}_k^\star - \mathcal{R}_k)r_1$. We observe the first component of $SK_d^{\vec{\mathcal{R}}^\star}$, $\omega(u \prod_{i\in I^\star} h_i^{\mathcal{R}_i^\star})^{\hat{r}_1} f^{\hat{r}_2}$. Since $u, f$ and $h_i$ can be obtained by removing blind factors from $U, F$ and $H_i$ respectively, it is $\omega((g_p^y / A_{n+1}^{x_{n+1}\cdot Id} \cdot \prod_{i\in\mathbb{I}} A_{n+1-i}^{x_i \mathcal{R}_i}) \cdot \prod_{i\in I^\star} A_{n+1-i}^{x_i \cdot \mathcal{R}_i^\star})^{\hat{r}_1} A_{n+2}^{z\cdot\hat{r}_2}$. We focus on the exponent of $g_p$ in above expression and have

$$\begin{pmatrix} y - a^{n+1}x_{n+1}Id - \\ \sum_{i\in\mathbb{I}} a^{n+1-i}x_i\mathcal{R}_i + \sum_{i\in I^\star} a^{n+1-i}x_i\mathcal{R}_i^\star \end{pmatrix} \hat{r}_1 + a^{n+2}z\hat{r}_2$$

$$= \begin{pmatrix} y + a^{n+1-k}x_k(\mathcal{R}_k^\star - \mathcal{R}_k) - \\ \sum_{i\in\mathbb{I}, i\notin I^\star} a^{n+1-i}x_i\mathcal{R}_i - a^{n+1}x_{n+1}Id \end{pmatrix} \cdot (a^{k+1}zr_1 + r_2)$$

$$+ a^{n+2}z \cdot \big(-x_k(\mathcal{R}^\star - \mathcal{R}_k)r_1\big)$$

$$= \begin{pmatrix} ya^{k+1}z - \\ \sum_{i\in\mathbb{I}, i\notin I^\star} a^{n+k+2-i}x_i\mathcal{R}_i - a^{n+k+2}x_{n+1}Id \end{pmatrix} z \cdot r_1 +$$

$$\begin{pmatrix} y - \sum_{i\in\mathbb{I}, i\notin I^\star} a^{n+1-i}x_i\mathcal{R}_i + \\ a^{n+1-k}x_k(\mathcal{R}_k^\star - \mathcal{R}_k) - a^{n+1}x_{n+1}Id \end{pmatrix} \cdot r_2$$

All the terms in above expression are not associated with $A_{n+2}$ and do not exceed $A_{2n+2}$, so $\mathcal{B}$ can compute the first component of $SK_d^{\vec{\mathcal{R}}^\star}$. Similarly, $\mathcal{B}$ can compute the rest components in $SK_d^{\vec{\mathcal{R}}^\star}$.

Next, $\mathcal{B}$ generates $SK_r^{\vec{\mathcal{R}}^\star}$ in a similar manner to generating $SK_d^{\vec{\mathcal{R}}^\star}$. The details of this procedure are highly similar to those of $SK_d^{\vec{\mathcal{R}}^\star}$, so they are skipped. $\mathcal{B}$ randomly choose $s_1, s_2, t_1, t_2 \in \mathbb{Z}_N$ and let

$$\hat{s}_1 = a^{k+1}zs_1 + s_2, \quad \hat{s}_2 = -x_k(\mathcal{R}_k^\star - \mathcal{R}_k)s_1$$
$$\hat{t}_1 = a^{k+1}zt_1 + t_2, \quad \hat{t}_2 = -x_k(\mathcal{R}_k^\star - \mathcal{R}_k)t_1$$

$\hat{s}_1, \hat{s}_2, \hat{t}_1, \hat{t}_2$ must satisfy equation $\hat{s}_1 \cdot \hat{t}_2 - \hat{s}_2 \cdot \hat{t}_1 \neq 0$ mod $p$ and mod $q$ with probability $1 - \frac{p+q-1}{N}$. Therefore $SK^{\vec{\mathcal{R}}^\star}$ has the same distribution and structure as that of the actual key distribution with probability $1 - \frac{p+q-1}{N}$.

**II**.When $\mathcal{A}$ issues secret key query for a patient with identity $Id^\star$, the restriction is that $Id^\star \neq Id$. To response the query, $\mathcal{B}$ randomly choose $\hat{r}_1', \hat{r}_2' \in \mathbb{Z}_N$ and let $\hat{r}_1' = azr_1' + r_2'$ and $\hat{r}_2' = -x_{n+1}(Id^\star - Id)r_1'$. Then it computes

$$SK^{Id^\star} = \left(\omega(ug_h^{Id^\star})^{\hat{r}_1'} f^{\hat{r}_2'}, g^{\hat{r}_1'}, g^{\hat{r}_2'}, \{h_j^{\hat{r}_1'}\}_{j\in[1,n]}\right)$$

As $u, g_h, f$ can be obtained by removing $R_u, R_h, R_f$ from $U, H, F$ respectively, the first component can be rewritten as $\omega((g_p^y / A_{n+1}^{x_{n+1}\cdot Id} \cdot \prod_{i\in\mathbb{I}} A_{n+1-i}^{x_i\mathcal{R}_i}) \cdot A_{n+1}^{x_{n+1}Id^\star})^{\hat{r}_1} \cdot A_{n+2}^{z\hat{r}_2'}$.

We focus on the exponent of $g_p^z$ and get

$$\begin{pmatrix} y - a^{n+1}x_{n+1}Id - \\ \sum_{i\in\mathbb{I}} a^{n+1-i}x_i\mathcal{R}_i + a^{n+1}x_{n+1}Id^\star \end{pmatrix} \hat{r}_1 + a^{n+2}z \cdot \hat{r}_2'$$

$$= (y - \sum_{i\in\mathbb{I}} a^{n+1-i}x_i\mathcal{R}_i)az \cdot r_1'$$

$$+ (y + a^{n+1}x_{n+1}(Id^\star - Id) - \sum_{i\in\mathbb{I}} a^{n+1-i}x_i\mathcal{R}_i)r_2'$$

All the terms in above expression are not associated with $A_{n+2}$ and do not exceed $A_{2n+2}$, so $\mathcal{B}$ can compute the first component of $SK^{Id^\star}$. Similarly, $\mathcal{B}$ can compute the rest components in $SK^{Id^\star}$.Finally, simulator $\mathcal{B}$ generates secret key for $Id^\star$ and responds it to adversary $\mathcal{A}$.

**Challenge**. Adversary $\mathcal{A}$ outputs two equal-length EHRs $EHR_0, EHR_1$. Simulator $\mathcal{B}$ ignores them and selects $R_{En}$ from $\mathbb{G}_T$. $\mathcal{B}$ also selects $R_1$ from $\mathbb{G}$ and random elements $Q_1, Q_2$ from $\mathbb{G}_q$. Then $\mathcal{B}$ sends the challenge ciphertext to adversary $\mathcal{A}$

$$(\mathrm{Hdr}, En) = (R_1, C_2, C_3, R_{En}) = (R_1, C^z Q_1, {T_4}^y Q_2, R_{En})$$

where $T_4$ are given from challenge tuple $(D_4, T_4)$. Observe each component in the challenge ciphertext and we have

$$C_2 = (A_{n+2}^b R_2')^z Q_1 = F^b Q_1'$$

If $T_4 = g_p^b \cdot R_3$, then $C_3 = (g_p^b R_3)^y Q_2 = (UH^{Id} \prod_{i\in\mathbb{I}} H_i^{\mathcal{R}_i})^b Q_2'$

If $T_4 \overset{R}{\leftarrow} \mathbb{G}$, $T_4$ can be written as $g_p^r \widetilde{R}_3$ where $r$ is random integer chosen from $\mathbb{Z}_N$ and $\widetilde{R}_3$ is a random element chosen from $\mathbb{G}_q$. Then $C_3 = (g_p^r \widetilde{R_3})^y Q_2 = (UH^{Id} \prod_{i\in\mathbb{I}} H_i^{\mathcal{R}_i})^r Q_2'$. In this case, $C_2$ uses random integer $b$, while $C_3$ uses random integer $r$ which is independent from $b$. Both $b$ and $r$ first appear in ciphertext from adversarial viewpoint and they are uniformly and independently chosen from $\mathbb{Z}_N$. Hence, $C_2$ and $C_3$ are random elements from the adversarial viewpoint and $(\mathrm{Hdr}, En)$ is the ciphertext of $\mathrm{Game}_5$.

**Phase 2**. Repeat Phase 1.

**Guess**. Adversary $\mathcal{A}$ outputs a guess. i.e. if $\mathcal{A}$ decides to output 0 ($\mathrm{Game}_4$ ), simulator $\mathcal{B}$ output $T_4 = g_p^b \cdot R_3$; if $\mathcal{A}$ decides to output 1 ($\mathrm{Game}_5$), $\mathcal{B}$ outputs $T_4 \overset{R}{\leftarrow} \mathbb{G}$. If $\mathcal{A}$ has the advantage $\epsilon_4'$ to distinguish $\mathrm{Game}_4$ and $\mathrm{Game}_5$, $\mathcal{B}$ can solve the Decision $(n+1)$-cDHE problem with the advantage $\epsilon_4' \cdot (1 - \frac{p+q-1}{N})^{q_s}$, which completes the proof of Lemma 4.