IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Privacy Preservation for Outsourced Medical Data with Flexible Access Control

XINGGUANG ZHOU[1], JIANWEI LIU[1], QIANHONG WU[1], AND ZONGYANG ZHANG.[1, 2]

[1]School of Cyber Science and Technology, Beihang University, Beijing, China (e-mail: {zhouxingguang, liujianwei, qianhong.wu, zongyangzhang}@buaa.edu.cn)

[2]State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China

Corresponding author: Zongyang Zhang (e-mail: zongyangzhang@buaa.edu.cn).

This work was supported in part by National Key R&D Program of China (2017YFB1400700), by Beijing Natural Science Foundation (4182033), by the fund of the State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences (No. 2017-MS-02), by Natural Science Foundation of China through projects 61672083, 61370190, 61532021, 61472429, and 61402029.

ABSTRACT Electronic medical records (EMRs) play an important role in healthcare networks. Since these records always contain considerable sensitive information regarding patients, privacy preservation for the EMR system is critical. Current schemes usually authorize a user to read one's EMR if and only if his/her role satisfies the defined access policy. However, these existing schemes allow an adversary to link patients' identities to their doctors. Therefore, classifications of patients' diseases are leaked without adversaries actually seeing patients' EMRs. To address this problem, we present two anonymous schemes. They not only achieve data confidentiality but also realize anonymity for individuals. The first scheme achieves moderate security, where adversaries choose attack targets before obtaining information from the EMR system. The second scheme achieves full security, where adversaries adaptively choose attack targets after interaction with the EMR system. We provide rigorous proof showing the security and anonymity of our schemes. In addition, we propose an approach in which EMR owners can search for their EMRs in an anonymous system. For a better user experience, we apply the "online/offline" approach to speed up data processing. Experimental results show that the time complexity for key generation and EMR encapsulation can be reduced to milliseconds.

INDEX TERMS privacy preservation, security, electronic medical record.

I. INTRODUCTION

Currently, electronic medical records (EMRs) are very prominent in healthcare networks. They enables users to share their health data in a flexible and convenient way. For example, to find one's diagnostic report, a patient or his/her doctor needs only to retrieve the information from a database rather than having to search through numerous physical documents. Health data is very sensitive, and it is a major challenge to securely store and access EMRs in modern EMR systems. As most EMRs are outsourced to the cloud, they are easily exposed to potential threats and vulnerable to leakage, loss, and theft [1]. To prevent EMRs from unauthorized access, a standard solution is to perform an encryption before uploading them to the cloud.

Specifically, an EMR owner encrypts an EMR using a symmetric key, and only authorized medical staff are authorized to access and decrypt it. However, data sharing becomes inflexible in this case. Two potential issues are the complicated key management and repetitive encryption [2]: as patients usually do not know who is allowed to access their EMRs, they encrypt many pieces with distinct session keys and distribute the keys to different medical staff members.

The approach to accessing users' data needs to be flexible enough to address changes in users' roles [3]. Several schemes adopting attribute-based encryption (ABE) have been presented for fine-grained access control [4], [5]. Users with attributes satisfying the access policy can decapsulate the EMR data. In addition, some advanced mechanisms, consisting of a multi-authority model in an outsourcing system [6] and a view-based access control [7] that allows patients to specify a list of authorized/unauthorized users, have recently been proposed. Role-based access control

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2810243, IEEE Access

schemes (RBACs) [8] also allow fine-grained access control. They define a role-based policy for a hierarchical organization with identity-based broadcast encryption (HIBBE). While the above proposals achieve data confidentiality in the EMR system, privacy preservation for patients is still an unresolved issue. For example, an EMR of the patient "Lucy" is uploaded to the cloud, and no attacker can read the encrypted EMR. If the doctor is an expert in the hepatitis disease, an attacker can infer that Lucy may carry hepatitis B without decrypting her EMR. This means that an attacker can obtain her disease-related information by linking Lucy to her doctor, even without seeing the detailed EMR. This means that adversaries possess the capacity that no matter if the EMRs are encrypted or not, adversaries can deduce the EMR owners' diseases based on some experience, such as the acquired identity-related information. Therefore, if there is an anonymous scheme that obfuscates the identity of the patient during an examination, adversaries can only determine that "someone" carries hepatitis B without knowing who it is. Thus, the patient's privacy is preserved.

A. OUR CONTRIBUTIONS

We design two anonymous schemes, denoted as "RBACAnony" and "RBACAnony-F", to preserve patients' privacy in an EMR system with role-based access control. We present competing models and a high level demonstration of rigorous proof. In brief, our schemes have the advantage of data confidentiality, identity anonymity and access control flexibility. Technical details are highlighted as follows.

RBACAnony. This scheme is built on a bilinear group with two subgroups [9], and a patient's identity information is hidden in one of the subgroups. The identity-related element in this subgroup is indistinguishable from a random element chosen from the bilinear group. Therefore, an attacker cannot distinguish a patient's identity from a random string. In addition, an attacker chooses the targeted identities he/she wishes to attack before the system is set up. This means that an attacker of the RBACAnony scheme cannot obtain any experience prior to attacking.

RBACAnony-F. This scheme is built on a bilinear group with four subgroups [10], and a patient's identity information is hidden in one of the composite-order subgroups. The identity-related element in this subgroup cannot be distinguished from an element randomly chosen from the same subgroup. Therefore, an attacker cannot distinguish the patient's identity from a random string. In addition, an attacker adaptively gives out the targeted identities he/she wishes to attack after interacting with the EMR system. This means that an attacker of the RBACAnony-F scheme can accumulate experience before attacking and thus possesses a stronger ability to attack.

Versatile access control. A user encapsulates the EMR using an on-demand access policy. This policy enables one-to-many encryption, where the EMR is encrypted once and different medical staff members are allowed to access it.

Scalable data sharing. Senior medical staff members are allowed to delegate access privileges to their subordinates.

Anonymous search. A patient and his doctors can link themselves to the targeted EMR, but outsiders cannot.

B. RELATED WORK

Access control [11] is widely adopted in the EMR system to protect patients' health data. Access control policies are specified by some pieces of legislation, i.e., health insurance portability and accountability act (HIPAA) [12], electronic documents [13], and company rules or regulations. The legislation regulates who can access and how they can operate the stored EMRs. Two solutions are usually used to support flexible access control. One solution is to use attribute-based encryption [14], [15]. As attributes can be applied to describe users' privileges, data owners determine the access policies. The other solution is to use role-based access control schemes [8], where each user's identity denotes a role and one is allowed to gain access permission if his role belongs to a defined policy. However, there is still a lack of consideration regarding the identity privacy of EMR owners. Anonymization techniques can be used to guarantee users' identity privacy [16]. For example, some anonymous ABE schemes address not only data privacy but also identity privacy [17], [18]. These schemes provide an analysis of confidentiality, anonymity and flexibility.

In practice, an unaddressed challenge to real-world deployment remains: healthcare organizations are usually structured hierarchically, with data being shared among many users. In a previous work, we achieved anonymous role-based access control in this kind of organization with a moderate security level, where an attacker must output the targeted identities before communication with the EMR system [19]. This scheme is denoted as RBACAnony in this paper. We additionally propose a new scheme in the current work, denoted as RBACAnony-F, where an attacker can adaptively output the targeted identities after interaction with the EMR system. Both schemes preserve patients' privacy in a healthcare network. The anonymous algorithms in [10], [20] are used to achieve patient privacy for RBACAnony and RBACAnony-F, respectively.

II. PRELIMINARIES

A. NOTATIONS

We introduce several notations to simplify the illustration of our scheme. For ease of description, we borrow notations from [8], as summarized in Table 1.

B. BILINEAR GROUPS

Let $\mathcal{G}$ be a group generation algorithm that takes a security parameter λ as its input and outputs the description of a bilinear group $(N, \mathbb{G}, \mathbb{G}_T, e)$. In the case where $\mathcal{G}$ outputs $(N = p_1p_2p_3p_4, G, G_T, e)$, where p_1, p_2, p_3, p_4 are distinct prime factors, $\mathbb{G}$ and $\mathbb{G}_T$ are cyclic groups of order $N = p_1p_2p_3p_4$, and $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is an efficient bilinear

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2810243, IEEE Access

TABLE 1: Notations

Notation	Description
λ	Security parameter
ID	Identity of a patient
$\mathcal{R}$	Atom role of medical staff member
$\overrightarrow{\mathcal{R}}$	Role of medical staff member
$S_{\overrightarrow{\mathcal{R}}}$	Atom role set for $\overrightarrow{\mathcal{R}}$
$\mathcal{P}$	Access policy
$S_{\mathcal{P}}$	Atom role set for $\mathcal{P}$
$Pref(\overrightarrow{\mathcal{R}})$	Prefix of $\overrightarrow{\mathcal{R}}$, defined as $\{(\mathcal{R}_1, ..., \mathcal{R}_{d'}) : d' \leq d\}$
$Pref(\mathcal{P})$	Prefix of $\mathcal{P}$, defined as $\bigcup_{\overrightarrow{\mathcal{R}} \in \mathcal{P}} Pref(\overrightarrow{\mathcal{R}})$
MSK	Master secret key
$SK^{\overrightarrow{\mathcal{R}}}$	Secret key for a role $\overrightarrow{\mathcal{R}}$
EMR	Electronic medical record
Hdr	Header of an uploaded EMR
K	Message encapsulation key
CT	Ciphertext for the encapsulated EMR
H	Collision resistant hash function $\{0,1\}^\star \rightarrow \mathbb{Z}_N$
$SymEnc$	Secure symmetric encryption algorithm
$SymDec$	Secure symmetric decryption algorithm
PPT	Probabilistic polynomial time

map satisfying the following two properties: (i) bilinearity: for all $g, h \in \mathbb{G}$ and all $a, b \in \mathbb{Z}_N$, $e(g^a, h^b) = e(g, h)^{ab}$; (ii) 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 subgroups of order p_1, p_2, p_3, p_4 in $\mathbb{G}$ as $\mathbb{G}_{p_1}, \mathbb{G}_{p_2}, \mathbb{G}_{p_3}$ and $\mathbb{G}_{p_4}$. We use $G_{p_i p_j}$ $(1 \leq i, j \leq 4)$ to denote the subgroup of order $p_i p_j$ in $\mathbb{G}$. These four subgroups additionally satisfy the orthogonality property, i.e., $\forall h_i \in \mathbb{G}_{p_i}$ and $h_j \in \mathbb{G}_{p_j}$ for $i \neq j$, $e(h_j, h_j) = 1$.

Composite-order bilinear groups were first introduced in [9] and are widely used as tools for constructing cryptographic primitives [21].

III. SYSTEM MODEL

A. SYSTEM ARCHITECTURE

We describe a typical healthcare network in Figure. 1. It mainly includes three entities: the trusted keying authority (TKA), the patient and the medical staff.

The TKA is trusted in the system and is responsible for generating and distributing system parameters, rooting master keys, and authorizing top-level medical staff and patients.

The patient is identified by his/her name or identity. The patient and his/her responsible medical staff are the EMR owners.

The top-level medical staff member delegates privileges to his subordinates, which forms a tree-like organization. Each staff member is identified by a role vector consisting of ordered atom roles. For instance, the role vector for an intern doctor, consisting of ordered atom roles "chief doctor, associate doctor, intern doctor", is administrated by the associate doctor, whose atoms roles are "chief doctor, associate doctor". We assign the chief doctor, the associate doctor and the intern doctor to one access policy for a certain patient.

Each user can encapsulate the patient's EMR, but only the one whose role satisfies the defined access policy or the patient himself can decapsulate it. We hide all the identity-related information in the system such that adversaries cannot infer patients' personal information. The adversaries include the dishonest internal staff and the malicious external attackers.

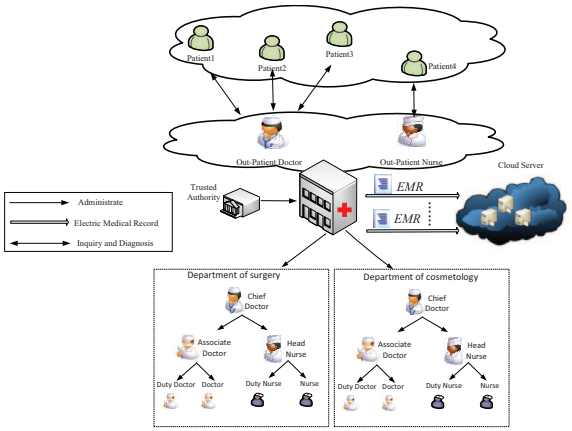


FIGURE 1: System architecture: a typical healthcare network

B. SECURITY REQUIREMENTS

In practice, all entities are likely to attack an EMR system. A dishonest party may try to obtain useful information from encrypted data that it is not authorized to access or to divert instructions from the system regarding benefits (e.g., with false information in medical disputes). Multiple dishonest parties may collude to achieve this goal. In the context of these attacks, the EMR system is expected to meet the following security requirements.

- **Data Confidentiality.** Personal data needs to be encrypted before being uploaded and securely stored on the cloud until an entitled recipient downloads and decrypts it. Specifically, only the users whose roles satisfy the associated access policy have the privilege to access the data, with all other unauthorized entities not able to obtain any useful information from the encrypted data, even if they collude with each other.
- **Identity Anonymity.** Identity-related information needs to be hidden, as individual privacy is vulnerable to loss, theft, and illegal transactions. When a user's identity is hidden in an EMR system, it decreases the possibility of an adversary guessing that user's identity such that hardly any third party can obtain useful patient information.

C. SECURITY MODELS

Our security models include the semantic security model, anonymity model and full anonymity model. The semantic security model is used to meet the requirement of data confidentiality, while the anonymity model and the full anonymity model are used to meet the requirement of identity anonymity. We define them according to the security games played between an adversary $\mathcal{A}$ and a challenger.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2810243, IEEE Access

1) Semantic Security Model

We adopt the selective security notion [8], i.e., an adversary must present the set of medical staff roles and the identity of the patient it wishes to attack before the system is set up.

Init. The adversary $\mathcal{A}$ outputs a challenge access policy set $\mathcal{P}^*$ and a challenge identity ID^*.

Setup. The challenger runs the 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:

- Upon receiving a secret key query for a medical staff member associated with a role $\overrightarrow{\mathcal{R}}$ such that $\overrightarrow{\mathcal{R}} \notin Pref(\mathcal{P}^\star)$, the challenger generates a secret key for $\overrightarrow{\mathcal{R}}$ and gives it to $\mathcal{A}$.
- Upon receiving a secret key query for patients with an identity ID such that $ID \neq ID^\star$, the challenger generates a secret key for ID and gives it to $\mathcal{A}$.

Challenge. When the adversary $\mathcal{A}$ decides that it has obtained enough secret keys, it outputs two equal-length EMR files EMR_0, EMR_1 that it wishes to challenge. The challenger picks a random bit $\beta \in \{0,1\}$ and encapsulates the EMR_β under the challenge access policy set $\mathcal{P}^*$ and the challenge identity ID^*. It gives $\mathcal{A}$ the challenge ciphertext (Hdr, En), where En is the output of the encapsulation of EMR_β.

Query Phase 2. Phase 1 is repeated adaptively.

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

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

2) Anonymity model

The **Init**, **Setup**, and **Query** phases are the same as that in the semantic security model.

Challenge. When an adversary $\mathcal{A}$ decides that it has obtained enough secret keys, it outputs two equal-length EMRs EMR_0, EMR_1 regarding what it wishes to be challenged. The challenger picks a random bit $\beta \in \{0,1\}$. If $\beta = 0$, it generates the header Hdr of the ciphertext under the challenge access policy set $\mathcal{P}^*$ and the challenge identity ID^* and encapsulates EMR_0. If $\beta = 1$, it generates the header of the ciphertext under a random access policy set and a random patient's identity and encapsulates EMR_1. It gives $\mathcal{A}$ the challenge ciphertext (Hdr, En), where En is the output of the encapsulation of EMR_β.

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

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

3) Full Secure Anonymity Model

In the full secure anonymity model, instead of committing the challenge access policy $\mathcal{P}^*$ and the challenge identity ID^* it wishes to attack before the system is set up, the adversary $\mathcal{A}$ can adaptively decide to output the challenge access policy set and identity during the system interaction. Clearly, this model achieves a stronger security level. Specifically, there is no **Init** phase in the full secure anonymity model. The adversary $\mathcal{A}$ outputs a challenge access policy set $\mathcal{P}^*$ and a challenge patient's identity ID^* that it wishes to attack after it issues sufficient key queries in the **Query** phase. The challenge access policy set $\mathcal{P}^*$ and the challenge identity ID^* should satisfy the following: for all the secret key queries for roles $\overrightarrow{\mathcal{R}}$ and identity ID in Query Phase 1, $\overrightarrow{\mathcal{R}} \notin Pref(\mathcal{P}^*)$ and $ID \neq ID^\star$.

IV. RBACANONY CONSTRUCTION

A. OUR PROPOSAL

Our RBACAnony scheme is based on the HIBE scheme proposed by Boneh et al. [22] and the RBAC scheme proposed by Liu et al. [8] and offers an efficient approach to supporting hierarchical access control. The property is motivated by Seo et al. [20] and is achieved by leveraging bilinear groups with composite order $N = pq$. 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 hidden by the elements in the subgroup $\mathbb{G}_q$, which helps to ensure anonymity. In this way, we offer information regarding the subgroup $\mathbb{G}_p$ in the "key generation layer" while maintaining our scheme's anonymity with the help of the "anonymity layer".

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

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

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

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

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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2810243, IEEE Access

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

$$SK_d^{\overrightarrow{\mathcal{R}}} = \left\{\omega\Big(u\prod_{i\in I}h_i^{\mathcal{R}_i}\Big)^{r_1} f^{r_2}, g^{r_1}, g^{r_2}, g_h^{r_1}, \{h_j^{r_1}\}\right\}$$

$$SK_r^{\overrightarrow{\mathcal{R}}} = \left\{\Big(u\prod_{i\in I}h_i^{\mathcal{R}_i}\Big)^{s_1} f^{s_2}, g^{s_1}, g^{s_2}, g_h^{s_1}, \{h_j^{s_1}\}, \Big(u\prod_{i\in I}h_i^{\mathcal{R}_i}\Big)^{t_1} f^{t_2}, g^{t_1}, g^{t_2}, g_h^{t_1}, \{h_j^{t_1}\}\right\}$$

In the above equations, $j \in [1, n]\backslash I$. Finally, the TKA outputs $SK^{\overrightarrow{\mathcal{R}}} = \left\{SK_d^{\overrightarrow{\mathcal{R}}}, SK_r^{\overrightarrow{\mathcal{R}}}\right\}$ for the medical staff.

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

$$SK_d^{\overrightarrow{\mathcal{R}'}} = \left\{a_{d,0}, a_{d,1}, a_{d,2}, a_{d,3}, \{b_{d,j}\}_{j\in[1,n]\backslash I'}\right\}$$

$$SK_r^{\overrightarrow{\mathcal{R}'}} = \left\{a_{r,0}, a_{r,1}, a_{r,2}, a_{r,3}, \{b_{r,j}\}_{j\in[1,n]\backslash I'}, a'_{r,0}, a'_{r,1}, a'_{r,2}, a'_{r,3}, \{b'_{r,j}\}_{j\in[1,n]\backslash I'}\right\}$$

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

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

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

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

$$SK_d^{\overrightarrow{\mathcal{R}}} = \left\{\omega\Big(u\cdot\prod_{i\in I}h_i^{\mathcal{R}_i}\Big)^{\widetilde{r}_1} f^{\widetilde{r}_2}, g^{\widetilde{r}_1}, g^{\widetilde{r}_2}, g_h^{\widetilde{r}_1}, \{h_j^{\widetilde{r}_1}\}\right\}$$

where $j \in [1, n]\backslash I$ and

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

It follows that $SK_d^{\overrightarrow{\mathcal{R}}}$ is well formed, appearing as if it was generated directly by the TKA using the KeyGenM algorithm.

To delegate $SK_r^{\overrightarrow{\mathcal{R}}}$, the high-level medical staff member picks random exponents $\gamma_2, \delta_2, \gamma_3, \delta_3 \xleftarrow{R} \mathbb{Z}_N$ satisfying $g_p^{\gamma_2\cdot\delta_3-\gamma_3\cdot\delta_2} \neq 1$ and $g_q^{\gamma_2\cdot\delta_3-\gamma_3\cdot\delta_2} \neq 1$. Then, he delegates the secret key by using

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

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

$$SK_r^{\overrightarrow{\mathcal{R}}} = \left\{\begin{array}{c}\Big(u\cdot\prod_{i\in I}h_i^{\mathcal{R}_i}\Big)^{\widetilde{s}_1} f^{\widetilde{s}_2}, g^{\widetilde{s}_1}, g^{\widetilde{s}_2}, g_h^{\widetilde{s}_1}, \{h_j^{\widetilde{s}_1}\},\\ \Big(u\cdot\prod_{i\in I}h_i^{\mathcal{R}_i}\Big)^{\widetilde{t}_1} f^{\widetilde{t}_2}, g^{\widetilde{t}_1}, g^{\widetilde{t}_2}, g_h^{\widetilde{t}_1}, \{h_j^{\widetilde{t}_1}\}\end{array}\right\}$$

where $j \in [1, n]\backslash I$ and

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

In conclusion, by running KeyDelegM, the delegated secret key is well formed, appearing as if it was generated directly by the TKA using KeyGenM.

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

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

EMREnc$(PK, ID, \mathcal{P}, EMR)$. For an access policy $\mathcal{P}$, denote $\mathbb{I} = \{i : \mathcal{R}_i \in S_{\mathcal{P}}\}$. When an EMR needs to be encapsulated under a patient's identity ID and an access policy $\mathcal{P}$, the user (the patient or the medical staff member) first picks a random exponent $s \xleftarrow{R} \mathbb{Z}_N$ and random elements

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2810243, IEEE Access

$Z_1, Z_2, Z_3 \overset{R}{\leftarrow} \mathbb{G}_q$. Note that these random elements in $\mathbb{G}_q$ can be chosen by raising g_q to random exponents from $\mathbb{Z}_N$. Next, the user computes the header Hdr as follows:

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

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

$\mathsf{EMRDecM}(PK, ID, (Hdr, En), SK^{\overrightarrow{\mathcal{R}}})$. To retrieve the session key K, the medical staff member with a role satisfying the access policy $\mathcal{P}$ can use his secret key to compute

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

Finally, $EMR = \mathsf{SymDec}(K, En)$ is run to obtain the EMR.

Correctness. Assume that $CT = ((C_1, C_2, C_3), En)$ is a well-formed ciphertext. The medical staff decapsulation algorithm can correctly recover the EMR file with a valid secret key $SK^{\overrightarrow{\mathcal{R}}}$, where $\overrightarrow{\mathcal{R}} \in Pref(\mathcal{P})$ due to the following:

$$\begin{aligned} K =& \frac{e\left(w\left(u \prod_{i \in I} h_i^{\mathcal{R}_i}\right)^{r_1} f^{r_2} \cdot g_h^{r_1 \cdot ID} \cdot \prod_{i \in \mathbb{I} \setminus I} (h_i^{\mathcal{R}_i})^{r_1}, g^s\right)}{e\left(g^{r_1}, \left(u \cdot g_h^{ID} \cdot \prod_{i \in \mathbb{I}} h_i^{\mathcal{R}_i}\right)^s\right) e\left(g^{r_2}, f^s\right)} \\ =& e(g, \omega)^s \end{aligned}$$

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

$\mathsf{EMRDecP}(PK, ID, (Hdr, En), SK^{ID})$. The patient with identity ID can decapsulate his own EMR using his secret key. We denote $\mathbb{I} = \{i : \mathcal{R}_i \in S_{\mathcal{P}}\}$. The patient computes the session key

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

Finally, he runs $\mathsf{SymDec}(K, En)$ to recover his EMR.

Correctness. Assume that $CT = ((C_1, C_2, C_3), En)$ is a well-formed ciphertext. A patient can recover his EMR according to the following equations.

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

B. SECURITY ANALYSIS

We showed that the RBACAnony scheme is selectively secure and anonymous in Sections III-C1 and III-C2, respectively. We now validate these characteristics via the following games between an adversary and a challenger.

- CT_1 of Game_1: $((C_1, C_2, C_3,), En)$
- CT_2 of Game_2: $((C_1, C_2, C_3,), En \cdot R_p)$
- CT_3 of Game_3: $((C_1, C_2, C_3,), En \cdot R = R_{En})$
- CT_4 of Game_4: $((R_1, C_2, C_3,), R_{En})$
- 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 and R_{En} are uniformly distributed in $\mathbb{G}_T$, and R_1, R_2, and R_3 are uniformly distributed in $\mathbb{G}$.

Proof. We develop the proof via contradiction. Assume that a PPT adversary can break the RBACAnony scheme. We then solve a series of difficult-to-solve mathematical assumptions: l-BDHE assumption, BSD assumption, l-cDH assumption and l-cDHE assumption [19]. Since no PPT algorithm can be used to solve these assumptions, we reach a contradiction and conclude that RBACAnony is secure. If the group generator algorithm $\mathbb{G}$ satisfies the BDHE assumption and the BSD assumption, then no PPT adversary can distinguish Game_1 and Game_3. The ciphertext of Game_3 does not leak any information regarding the EMR data since the component corresponding to the EMR is a random group element. If $\mathbb{G}$ satisfies the cDH assumption and the cDHE assumption, then no PPT adversary can distinguish Game_3 and Game_5. The ciphertext of Game_5 does not leak any information regarding the roles of medical staff members and the identity of the patient, as the components related to the roles and identity are random group elements. Concrete proof of this is given in our previous work [19].

V. ACHIEVING FULL SECURE ANONYMITY

A. OUR PROPOSAL

In this section, we show how to achieve full anonymity privilege control in RBACAnony-F. We apply the idea of an anonymous HIBE [10] to our RBAC. A user first chooses an access policy, which can be regarded as a broadcast group with all entitled identities. He only needs to encapsulate the EMR once and allows different medical staff members to decapsulate if their identities belong to this broadcast group. Note that the work in [23] also proposed an anonymous HIBBE scheme. The main difference lies in the fact that the patients are identified individually in our scheme, while they are allowed access to their own EMRs in [23]. Thus, we consider the patients' identities in addition to the access policy group when we design the broadcast encryption algorithm.

$\mathsf{Setup}(\lambda, n)$. The TKA chooses a bilinear group G of order $N = p_1 p_2 p_3 p_4$. Then, it chooses random elements $Y_1, X_1, u_1, ..., u_n, u_P \in G_{p_1}, Y_3 \in G_{p_3}, X_4, Y_4 \in G_{p_4}$, and $\alpha \in Z_N$ and outputs the public key PK

$$\{N, Y_1, Y_3, Y_4, u_P, \{u_i\}_{i \in [1,n]}, x = X_1 X_4, A = e(Y_1, Y_1)^\alpha\}$$

and master secret key $MSK = \{X_1, \alpha\}$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2810243, IEEE Access

KeyGenM$(PK, MSK, \overrightarrow{\mathcal{R}})$. For any medical staff member with a role $\overrightarrow{\mathcal{R}} = (\mathcal{R}_1, ..., \mathcal{R}_d)$, I denotes $\{i : \mathcal{R}_i \in S_{\overrightarrow{\mathcal{R}}}\}$. When a top-level medical staff member wants to join the system, the TKA first authenticates him. It then chooses random elements $r_1, r_2 \in Z_N$, $R_{m,1}, R_{m,2}, \{T_{m,j}\}_{j\in[1,n]\setminus I} \in G_{p_3}$ for $m \in \{1, 2\}$ and $R_{P_1}, R_{P_2} \in G_{p_3}$ and outputs the secret key $SK^{\overrightarrow{\mathcal{R}}} = (SK_d^{\overrightarrow{\mathcal{R}}}, SK_r^{\overrightarrow{\mathcal{R}}})$, where $SK_d^{\overrightarrow{\mathcal{R}}}$ is used for decryption and $SK_r^{\overrightarrow{\mathcal{R}}}$ is used for re-randomization delegation.

$$SK_d^{\overrightarrow{\mathcal{R}}} = \{K_{1,1}, K_{1,2}, \{K_{1,j}\}_{j\in[1,n]\setminus I}, K_{P_1}\} = \left\{ \begin{array}{c} Y_1^{r_1} R_{1,1}, Y_1^{\alpha}(X_1 \prod_{i\in I} u_i^{\mathcal{R}_i})^{r_1} R_{1,2}, \\ \{u_j^{r_1} T_{1,j}\}_{j\in[1,n]\setminus I}, u_P^{r_1} R_{P_1} \end{array} \right\}$$

$$SK_r^{\overrightarrow{\mathcal{R}}} = \{K_{2,1}, K_{2,2}, \{K_{2,j}\}_{j\in[1,n]\setminus I}, K_{P_2} =\} \left\{ Y_1^{r_2} R_{2,1}, (X_1 \prod_{i\in I} u_i^{\mathcal{R}_i})^{r_2} R_{2,2}, \{u_j^{r_2} T_{2,j}\}_{j\in[1,n]\setminus I}, u_P^{r_2} R_{P_2} \right\}$$

KeyDelegM$(PK, SK^{\overrightarrow{\mathcal{R}'}}, \mathcal{R})$. For the low-level medical staff member with the role $\overrightarrow{\mathcal{R}} = (\overrightarrow{\mathcal{R}'}, \mathcal{R})$, his secret key is derived from a given secret key of his supervisor, who is at a higher-level associated with the role $\overrightarrow{\mathcal{R}'}$. We denote $I' = \{i : \mathcal{R}_i \in S_{\overrightarrow{\mathcal{R}'}}\}$. Given a secret key $SK^{\overrightarrow{\mathcal{R}'}}$, the high-level medical staff member picks random components $\widetilde{r}_1, \widetilde{r}_2 \in Z_N$, $\widetilde{R}_{m,1}, \widetilde{R}_{m,2}, \{\widetilde{T}_{m,j}\}_{j\in[1,n]\setminus I} \in G_{p_3}$ for $m \in \{1, 2\}$, and $\widetilde{R}_{P_1}, \widetilde{R}_{P_2} \in G_{p_3}$ and computes

$$SK_d^{\overrightarrow{\mathcal{R}}} = \left\{ \begin{array}{l} K'_{1,1}(K'_{2,1})^{\widetilde{r}_1} \widetilde{R}_{1,1}, \\ K'_{1,2}(K'_{2,2})^{\widetilde{r}_1}((K'_{1,i})^{\mathcal{R}_i}(K'_{2,i})^{\widetilde{r}_1 \mathcal{R}_i})_{i\in I\setminus I'} \cdot \widetilde{R}_{1,2}, \\ \{K'_{1,j}(K'_{2,j})^{\widetilde{r}_1} \widetilde{T}_{1,j}\}_{j\in[1,n]\setminus I}, \quad K'_{P_1}(K'_{P_2})^{\widetilde{r}_1} \widetilde{R}_{P_1} \end{array} \right\}$$

$$SK_r^{\overrightarrow{\mathcal{R}}} = \left\{ \begin{array}{l} (K'_{2,1})^{\widetilde{r}_2} \widetilde{R}_{2,1}, (K'_{2,2})^{\widetilde{r}_2}((K'_{2,i})^{\widetilde{r}_2 \mathcal{R}_i})_{i\in I\setminus I'} \cdot \widetilde{R}_{2,2}, \\ \{(K'_{2,j})^{\widetilde{r}_2} \widetilde{T}_{2,j}\}_{j\in[1,n]\setminus I}, \quad (K'_{P_2})^{\widetilde{r}_2} \widetilde{R}_{P_2} \end{array} \right\}$$

where $I = \{i : \mathcal{R}_i \in S_{\overrightarrow{\mathcal{R}}}\}$. The delegated secret key can be finally attained in the form

$$SK_d^{\overrightarrow{\mathcal{R}}} = \left\{ \begin{array}{c} Y_1^{\hat{r}_1} \hat{R}_{1,1}, Y_1^{\alpha}(X_1 \prod_{i\in I} u_i^{\mathcal{R}_i})^{\hat{r}_1} \hat{R}_{1,2}, \\ \{u_j^{\hat{r}_1} \hat{T}_{1,j}\}_{j\in[1,n]\setminus I}, u_P^{\hat{r}_1} \hat{R}_{P_1} \end{array} \right\}$$

$$SK_r^{\overrightarrow{\mathcal{R}}} = \left\{ \begin{array}{c} Y_1^{\hat{r}_2} \hat{R}_{2,1}, (X_1 \prod_{i\in I} u_i^{\mathcal{R}_i})^{\hat{r}_2} \hat{R}_{2,2}, \\ \{u_j^{\hat{r}_2} \hat{T}_{2,j}\}_{j\in[1,n]\setminus I}, u_P^{\hat{r}_2} \hat{R}_{P_2} \end{array} \right\}$$

The new secret key has the same distributions as if it was computed using KeyGenM with randomness $\hat{r}_1 = r_1 + r_2\widetilde{r}_1$ and $\hat{r}_2 = r_2\widetilde{r}_2$.

KeyGenP(PK, MSK, ID). When a patient with identity ID wants to access his own EMR, the TKA authorizes him and randomly chooses $r'_1 \in Z_N$, $R'_1, \{T_j\}_{j\in[1,n]} \in G_{p_3}$. The TKA then outputs

$$SK^{ID} = \{d_{p_1}, d_{p_2}, \{d_{p_j}\}_{j\in[1,n]}\} = \left\{ Y_1^{r'_1} R'_1, Y_1^{\alpha}(u_P^{ID} X_1)^{r'_1} R'_1, \{u_j^{r'_1} T_j\}_{j\in[1,n]} \right\}$$

EMREnc$(PK, ID, \mathcal{P}, EMR)$. For an access policy $\mathcal{P}$, denote $\mathbb{I} = \{i : \mathcal{R}_i \in S_{\mathcal{P}}\}$. When an EMR file needs to be encapsulated under the access policy $\mathcal{P}$ and the patient's identity ID, the userrandomly picks $s \in Z_N$ and $Z, Z' \in G_{p_4}$ and computes the header Hdr as follows:

$$Hdr = \{C_1, C_2\} = \{(\prod_{i\in\mathbb{I}} u_i^{\mathcal{R}_i} u_P^{ID} x)^s Z, \ \ Y_1^s Z'\}$$

Then, the user generates session key $K = A^s$ and computes $En = \text{SymEnc}(K, EMR)$. Finally, the encapsulated EMR is output as $CT = (Hdr, En) = (C_1, C_2, En)$.

EMRDecM$(PK, ID, (Hdr, En), SK^{\overrightarrow{\mathcal{R}}})$. To retrieve session key K, the medical staff member with the role satisfying the access policy $\mathcal{P}$ can use his secret key to compute

$$K = \frac{e(K_{1,2} \cdot K_{P_1}^{ID} \cdot \prod_{i\in\mathbb{I}\setminus I} K_{1,i}^{\mathcal{R}_i}, C_2)}{e(K_{1,1}, C_1)}$$

Then, he runs $EMR = \text{SymDec}(K, En)$ to recover the EMR.

Correctness. Assume that $CT = ((C_1, C_2), En)$ is a well-formed ciphertext. EMRDecM can correctly recover the EMR file using a valid secret key $SK^{\overrightarrow{\mathcal{R}}}$, where $\overrightarrow{\mathcal{R}} \in Pref(\mathcal{P})$ because

$$K = \frac{e(Y_1, Y_1)^{\alpha s} \cdot e((X_1 \prod_{i\in\mathbb{I}} u_i^{\mathcal{R}_i} u_P^{ID})^{r_1}, Y_1^s)}{e(Y_1^{r_1}, (\prod_{i\in\mathbb{I}} u_i^{\mathcal{R}_i} u_P^{ID} X_1)^s)} = A^s$$

EMRDecP$(PK, \mathcal{P}, (Hdr, En), SK^{ID})$. The patient with identity ID can decapsulate his EMR using his secret key. We denote $\mathbb{I} = \{i : \mathcal{R}_i \in S_{\mathcal{P}}\}$. The patient computes a session key

$$K = \frac{e(d_{p_2} \cdot \prod_{i\in\mathbb{I}} d_{p_i}^{\mathcal{R}_i}, C_2)}{e(d_{p_1}, C_1)}$$

Then, he runs $EMR = \text{SymDec}(K, En)$ to recover the EMR.

Correctness. Assuming that $CT = ((C_1, C_2), En)$ is a well-formed ciphertext, a patient can correctly recover his EMR using the following equalities:

$$K = \frac{e(Y_1, Y_1)^{\alpha s} \cdot e((\prod_{i\in\mathbb{I}} u_i^{\mathcal{R}_i} u_P^{ID} X_1)^{r'_1}, Y_1^s)}{e(Y_1^{r'_1}, (\prod_{i\in\mathbb{I}} u_i^{\mathcal{R}_i} u_P^{ID} X_1)^s)} = A^s$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2810243, IEEE Access

B. SECURITY ANALYSIS

In this subsection, we provide a security analysis to demonstrate that RBACAnony-F is fully anonymous. We apply the dual system encryption technique introduced by Lewko [24], which has been used as a powerful tool for security analysis. In the proof, ciphertexts (CTs) and secret keys (SKs) can take one of two indistinguishable forms: normal form and semi-functional form, with the correlation shown in Table 2. Since the two kinds of ciphertexts and keys are indistinguishable, a simulator is able to replace the normal key and ciphertext with the semi-functional ones in security games. When both the ciphertext and key are semi-functional, an adversary can obtain no information regarding the challenge ciphertext since the given key is not able to decapsulate the challenge ciphertext.

TABLE 2: Normal/semi-functional key and ciphertext

	Normal CT	Semi CT
Normal SK	decryption allowed	decryption allowed
Semi SK	decryption allowed	decryption not allowed

Semi-functional Ciphertext. The users run the EMREnc algorithm to construct a normal ciphertext (C'_1, C'_2, En'). Then, they choose random exponents $x, z_c \in Z_N$ and set $C_1 = C'_1 g_2^{x z_c}, C_2 = C'_2 g_2^{x}, En' = En$.

Semi-functional Key for Medical Staff. For the medical staff member with role $\overrightarrow{\mathcal{R}}$, TKA runs KeyGenM to generate normal keys $SK'^{\overrightarrow{\mathcal{R}}}_d = \{K'_{1,1}, K'_{1,2}, \{K'_{1,j}\}_{j\in[1,n]\setminus I}, K'_{P_1}\}$ and $SK'^{\overrightarrow{\mathcal{R}}}_r = \{K'_{2,1}, K'_{2,2}, \{K'_{2,j}\}_{j\in[1,n]\setminus I}, K'_{P_2}\}$. Then, it chooses random exponents $z, \gamma, z_k, z_{P_1}, z_{P_2} \in Z_N$ and $\{z_{m,j}\}_{j\in[1,n]\setminus I} \in Z_N$ for $m \in \{1, 2\}$. The semi-functional key can be set as

$$SK^{\overrightarrow{\mathcal{R}}}_d = \left\{ \begin{array}{c} K'_{1,1} \cdot g_2^{\gamma},\ K'_{1,2} \cdot g_2^{\gamma \cdot z_k}, \\ \{K'_{1,j} \cdot g_2^{\gamma z_{1,j}}\}_{j\in[1,n]\setminus I}, K'_{P_1} \cdot g_2^{\gamma \cdot z_{P_1}} \end{array} \right\}$$

$$SK^{\overrightarrow{\mathcal{R}}}_r = \left\{ \begin{array}{c} K'_{2,1} \cdot g_2^{z \cdot \gamma},\ K'_{2,2} \cdot g_2^{z \cdot \gamma \cdot z_k}, \\ \{K'_{1,j} \cdot g_2^{z \cdot \gamma z_{2,j}}\}_{j\in[1,n]\setminus I}, K'_{P_2} \cdot g_2^{z \cdot \gamma \cdot z_{P_2}} \end{array} \right\}$$

It can be seen that the EMRDecM algorithm will correctly output the EMR when decrypting a semi-functional ciphertext using a semi-functional key since the added elements in G_{p_2} can be cleared due to the orthogonality property. However, the blinding factor will be multiplied by an additional term $e(g_2, g_2)^{\gamma x (z_k + z_{P_1} ID + \sum_{i\in\mathbb{I}\setminus I} z_{1,i}\mathcal{R}_i - z_c)}$. If $z_c = z_k + z_{P_1} ID + \sum_{i\in\mathbb{I}\setminus I} z_{1,i}\mathcal{R}_i$, then decryption still works. Here, we regard the key for the medical staff as nominally semi-functional.

Semi-functional Key for Patient. For the patient with the identity ID, the TKA runs the KeyGenP algorithm to generate the normal key $SK^{ID} = \{d'_{p_1}, d'_{p_2}, \{d'_{p_j}\}_{l\in[1,n]}\}$. Then, it chooses random exponents $\gamma, \widetilde{z}_k, \{\widetilde{z}_j\}_{j\in[1,n]} \in Z_N$. The semi-functional key can be set as

$$SK^{ID} = \left\{ d'_{p_1} \cdot g_2^{\gamma}, d'_{p_2} \cdot g_2^{\gamma \cdot \widetilde{z}_k}, \{d'_{p_j} \cdot g_2^{\gamma \widetilde{z}_j}\}_{j\in[1,n]} \right\}$$

The EMRDecP algorithm will correctly output the EMR when decrypting a semi-functional ciphertext using a semi-functional key. The blinding factor will be multiplied by an additional term $e(g_2, g_2)^{\gamma x (\widetilde{z}_k + \sum_{i\in\mathbb{I}} \widetilde{z}_i \mathcal{R}_i - z_c)}$. If $z_c = \widetilde{z}_k + \sum_{i\in\mathbb{I}} \widetilde{z}_i \mathcal{R}_i$, then decryption still works. In this case, we regard the key for the patient as nominally semi-functional.

We can verify the security and anonymity via a series of games.

$Game_{real}$: This game is a real game for RBACAnony-F, which describes the real interaction between the adversary and the EMR system.

$Game_{real'}$: This game is the same as $Game_{real}$ except that all the secret key queries are answered by the secret key generation algorithm, not by the secret key delegation algorithm.

$Game_{restrict}$: This game is the same as $Game_{real'}$ except that the adversary cannot query secret keys for the roles that are prefixes of the challenge role modulo p_2. Namely, for any queried role $\overrightarrow{\mathcal{R}} = (\mathcal{R}_1, \mathcal{R}_2, \cdots, \mathcal{R}_d)$, $\exists\ \overrightarrow{\mathcal{R}^*} = (\mathcal{R}^*_1, \mathcal{R}^*_2, \cdots, \mathcal{R}^*_{d'}) \in Pref(\mathcal{P}^*)$ with $d' \leq d$, $s.t.\ \forall i \in [1, d'], \mathcal{R}_i = \mathcal{R}^*_i\ mod\ p_2$, where $\mathcal{P}^*$ is the challenge access policy, is not allowed.

$Game_k$: This game is identical to $Game_{restrict}$ except that the challenge ciphertext given to the adversary is semi-functional and the first k keys are semi-functional ($0 \leq k \leq q$). We note that in $Game_0$, only the challenge ciphertext is semi-functional; in $Game_q$, all secret keys and ciphertext are semi-functional.

$Game_{final'}$: This game is identical to $Game_q$ except that the challenge ciphertext is a semi-functional encapsulation, with the component corresponding to the EMR being a random message in $\mathbb{G}_T$. Thus, the ciphertext is independent from the messages provided by the adversary.

$Game_{final}$: This game is identical to $Game_{final'}$ except that the challenge ciphertext is semi-functional, with the components related to the roles and identity being random group elements in the subgroup $\mathbb{G}_{p_1 p_2 p_4}$. Thus, the ciphertext is independent from the roles and identity provided by the adversary.

Proof. In the appendix, we show that no polynomial time adversary can distinguish Game_{real} and Game_{final}. The ciphertext of $\text{Game}_{final'}$ does not leak any information regarding the EMR file. The ciphertext of Game_{final} does not leak any information regarding the roles of the medical staff and the identity of the patient. Thus, data confidentiality and identity anonymity are achieved.

VI. ANONYMOUS SEARCH

The EMR system may receive queries from the patient or the medical staff to search for someone's EMR. To respond to a search query, we set up an approach that links the EMR owners to their encapsulated EMR. We tag two labels, ID'

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2810243, IEEE Access

and $\mathcal{P}'$, with each ciphertext CT, forming $(CT_i, ID'_i, \mathcal{P}'_i)$. Assume that the total number of stored EMRs is m, $i \in [1, m]$. ID' and $\mathcal{P}'$ represent the hidden identity of the patient and the hidden roles of the medical staff, respectively, such that outsiders cannot identify them. Regarding the patient and medical staff, the following operations show how they can determine their EMR.

SearchInitial. In this phase, we generate some parameters necessary for the subsequent searching work. Let G_0 be a bilinear group of prime order p and g be a generator of G_0. For a generated ciphertext CT_i, the ith patient with identity ID_i randomly chooses an element $x_{ID_i} \leftarrow G_0$, and the ith group of the medical staff in access policy $\mathcal{P}_i$ randomly chooses 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 owned only 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 search labels: ID'_i and $\mathcal{P}'_i$. ID'_i which can be obtained by applying a hash function to 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 to 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, yielding $(CT_i, ID'_i, \mathcal{P}'_i)$.

Search. When a patient with identity $\mathcal{ID}$ tries to search for his EMR (or when one of his doctors tries to do this), he first hashes the identity $\mathcal{ID}$ and obtains $H(\mathcal{ID})$. Then, he searches through the various ID'_i in all patients' labels and pinpoints the one whose value equals $H(\mathcal{ID})$. When he obtains the index i, he uses his session key to decrypt the roles for the 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 a medical staff member and his identity, he can decapsulate CT_i using the corresponding secret key.

VII. IMPROVING USER EXPERIENCE

To achieve the perfect user experience, we speed up the data processing in the key generation and EMR encapsulation procedures. We apply online/offline cryptography [25] to our scheme. The online/offline technique was initiated by Goldreich and Micali [26] for signature schemes. Guo et al. [27] extended the offline algorithm to the identity-based encryption system. Briefly, the online/offline technique splits the encryption or key generation process into two phases: the offline phase, in which most of the complex computations are first executed by assuming a set of random identities, and the online phase, in which only simple computations are performed to produce the ciphertext or secret key once the identities are available. In this way, we show how to move the computational work for key generation and EMR encapsulation offline. The following offline/online algorithms are based on the RBACAnony scheme, while the algorithm for the RBACAnony-F scheme is omitted due to it having similar procedures and results.

Offline.KeyGenM(PK, MSK). The offline KeyGenM algorithm takes as its input the public parameters and master key, excluding the medical staff role. We assume a random role $\overrightarrow{\mathcal{R}_B}$ with bound B on the maximum number of atom roles, which can be used to generate a secret key. Denote $\overrightarrow{\mathcal{R}_B} = (x_1, x_2, ..., x_B)$ and $I_B = \{i : x_i \in S_{\overrightarrow{\mathcal{R}_B}}\}$, where x_i are randomly chosen from $\mathbb{Z}_N$ and regarded as intermediate atom roles. The algorithm picks random exponents $r_1, r_2, s_1, s_2, t_1, t_2 \xleftarrow{R} \mathbb{Z}_N$ satisfying $s_1 \cdot t_2 - s_2 \cdot t_1 \neq 0 \bmod p$ and $\bmod q$. Then, it generates the intermediate secret key $SK^{\overrightarrow{\mathcal{R}_B}}$, which consists of two subkeys: $SK_d^{\overrightarrow{\mathcal{R}_B}}$ and $SK_r^{\overrightarrow{\mathcal{R}_B}}$. $SK_d^{\overrightarrow{\mathcal{R}_B}}$ can be written in the 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. $SK_r^{\overrightarrow{\mathcal{R}_B}}$ has a form similar to that of $SK_d^{\overrightarrow{\mathcal{R}_B}}$, but it is not used for EMR encapsulation. We can view the procedure as key generation for the intermediate role $\overrightarrow{\mathcal{R}_B} = (x_1, x_2, ..., x_B)$. The work done in the offline phase is roughly equivalent to the work carried out for the regular KeyGenM algorithm.

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

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

The subkey $SK_r^{\overrightarrow{\mathcal{R}}}$ is output with a form similar to that of $SK_d^{\overrightarrow{\mathcal{R}}}$ but without the elements $\{K_i\}_{i\in I}$. The dominant cost in the online phase is $||\overrightarrow{\mathcal{R}}||$ 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 procedure as that in the KeyGenM phase, we omit the details of that process. The dominant cost in the online key delegation procedure is only one multiplication.

Offline.EMREnc(PK). The offline EMREnc algorithm takes as its input only the public parameters. 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 selects $y \xleftarrow{R} \mathbb{Z}_N$, which is assumed

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2810243, IEEE Access

to be the intermediate patient identity. Then, the algorithm picks a random element $s \xleftarrow{R} \mathbb{Z}_N$ and random elements $Z_1, Z_2, Z_3 \xleftarrow{R} \mathbb{G}_q$. Finally, it computes the intermediate header Hdr_{IT} as follows:

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

The header generated in the offline phase is roughly equivalent to the work of the regular EMREnc algorithm.

Online.EMREnc$(Hdr_{IT}, Id, \mathcal{P}, EMR)$. The online EMREnc algorithm takes as its input the intermediate header Hdr_{IT} from the offline EMREnc algorithm, a patient identity ID, an access policy $\mathcal{P}$ and the EMR. Denote $\mathbb{I} = \{i : \mathcal{R}_i \in S_{\mathcal{P}}\}$, and note that $\mathbb{I} \subseteq \mathbb{I}_B$ since we have assumed the maximum bound B on the number of atom roles. 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 the symmetric encryption time $En =$ SymEnc(K, EMR) is relatively fast, the cost of EMR encapsulation can be ignored. The dominant cost in the online phase is $(||\mathcal{P}|| + 1)$ 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 EMR can be correctly decapsulated after the online/offline algorithm is applied. The encapsulation key K is calculated using

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

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

VIII. PERFORMANCE ANALYSIS

A. THEORETICAL ANALYSIS

Table 3 shows the efficiency of our proposed scheme in detail. The system parameters, the master secret key and the other secret keys (for the medical staff and patients) are linearly proportional to the maximum number of atom roles. The header contains only three group elements in $\mathbb{G}$, achieving ciphertext with a constant size and being independent of the maximal depth of the hierarchy for the access policy set $\|\mathcal{P}\|$. In Table 3, we denote t_e as one exponent operation time in $\mathbb{G}$, t_m as one multiplication operation time in $\mathbb{G}$ and t_p as one pairing operation time. In the procedures of KeyGenM, KeyDelegM, KeyGenP, and EMREnc, exponentiations can be pre-computed by choosing random exponents.

Table 4 compares four schemes in terms of anonymity, order of the bilinear group and performance. We denote R-BACAnony as "Ours & Scheme-I", RBACAnony-F as "Ours & Scheme-II", and our schemes with the user experience improvement as "Ours & Improved".

B. EXPERIMENTAL PERFORMANCE

We conduct the experiment using an Intel Core i7 processor with 8 GB of RAM and a 2.6 GHZ CPU clock. We use an elliptic curve type A1 with the expression $y^2 = x^3 + x$ for the Tate symmetric pairing. The group order of $\mathbb{Z}_N$ is set to 512 bits, and the element size in $\mathbb{G}$ is also configured to 512 bits. The experiment is executed using the jPBC library (http://gas.dia.unisa.it/projects/jpbc/index.html).

We test the operational time required for key generation, key delegation, EMR encapsulation and decapsulation for medical staff. We show the performance results in Figure. 2(a)-(e). Figure. 2(f) and Figure. 2(g) show the operational time after the user experience is improved.

IX. CONCLUSION

In this paper, we propose two anonymous RBAC schemes for the EMR system. We achieve flexible access control such that the EMR data can be encapsulated according to an on-demand access policy, with only users whose roles satisfy the access policy being able to decapsulate it. Patients' privacy is preserved using a bilinear group, where all the identity-related information is hidden in a subgroup. Based on the chosen bilinear group assumptions, we prove that our proposed models have the property of semantic security and anonymity. We apply the "online/ offline" approach to achieve a better user experience.

.

APPENDIX A PROOF OF SECURITY OF RBACANONY-F

The security proof is based on the following assumptions.

$Assumption1$. Given group generator $\mathcal{G}$, we define the following distribution:

$$\begin{aligned} \mathbb{G} =& (N = p_1p_2p_3p_4, G, G_T, e) \xleftarrow{R} \mathcal{G} \\ g_1, A_1 \xleftarrow{R} G_{p_1}, A_2, B_2 \xleftarrow{R} G_{p_2}, g_3 \xleftarrow{R} G_{p_3}, g_4, B_4 \xleftarrow{R} G_{p_4} \\ D =& (\mathbb{G}, g_1, g_3, g_4, A_1A_2, B_2B_4) \end{aligned}$$

Then, this assumption determines whether the given element $T \xleftarrow{R} G_{p_1p_2p_4}$ or $T \xleftarrow{R} G_{p_1p_4}$.

The advantage of an algorithm $\mathcal{A}$ that outputs $\beta \in \{0, 1\}$ in breaking Assumption 1 is defined as

$$Adv1_{\mathcal{A}}(\lambda) = \left| \begin{aligned} &\Pr\left[\mathcal{A}\left(D, T \xleftarrow{R} G_{p_1p_2p_4}\right) = 1\right] \\ -\ &\Pr\left[\mathcal{A}\left(D, T \xleftarrow{R} G_{p_1p_4}\right) = 1\right] \end{aligned} \right| - \frac{1}{2}$$

Definition I. $\mathcal{G}$ satisfies Assumption 1 if $Adv1_{\mathcal{A}}(\lambda)$ is negligible for any polynomial-time algorithm $\mathcal{A}$.

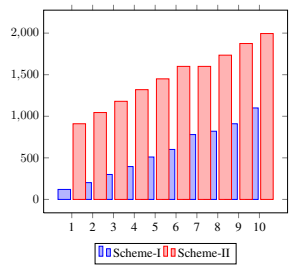
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2810243, IEEE Access

TABLE 3: Efficiency of the proposed schemes

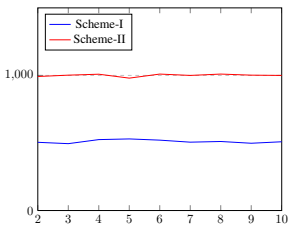
	RBACAnony with n atom roles	RBACAnony-F scheme with n atom roles
MSK size	$n+7$	2
$SK^{\overrightarrow{\mathcal{R}}}$ size	$3 \cdot (n+4-\|\overrightarrow{\mathcal{R}}\|)$	$2 \cdot (n+3-\|\overrightarrow{\mathcal{R}}\|)$
SK^{Id} size	$n+3$	$n+2$
Hdr size	3	2
KeyGenM time	$3 \cdot (n+5)t_e + (3\|\overrightarrow{\mathcal{R}}\|+4)t_m$	$(n+6)t_e + (2n+7)t_m$
KeyDelegM time	$(31+6n-6\|\overrightarrow{\mathcal{R}}\|)t_e +$ $(23+4n-4\|\overrightarrow{\mathcal{R}}\|)t_m$	$(8+2n-2\|\overrightarrow{\mathcal{R}}\|)t_e +$ $(2+3n-3\|\overrightarrow{\mathcal{R}}\|)t_m$
KeyGenP time	$(n+5)t_e + 3t_m$	$(n+4)t_e + (n+4)t_m$
EMREnc time	$(\|\mathcal{P}\|+5)t_e + (\|\mathcal{P}\|+4)t_m$	$(\|\mathcal{P}\|+3)t_e + (\|\mathcal{P}\|+3)t_m$
EMRDecM time	$(1+\|\mathcal{P}\|-\|\overrightarrow{\mathcal{R}}\|)(t_e+t_m)+3t_p+t_m$	$(1+\|\mathcal{P}\|-\|\overrightarrow{\mathcal{R}}\|)(t_e+t_m)+2t_p$
EMRDecP time	$(\|\mathcal{P}\|)(t_e+t_m)+3t_p+t_m$	$(\|\mathcal{P}\|)(t_e+t_m)+2t_p$

TABLE 4: Comparison with related work

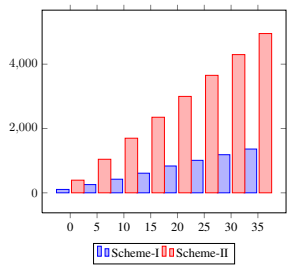
	Anonymity	Order of Bilinear Group	Key Generation Time	Key Delegation Time	EMR Enc Time	Number of pairings in EMR Dec
[8]	×	prime order	$(n+6)t_e +$ $(\|\overrightarrow{\mathcal{R}}\|+1)t_m$	$(n+6)t_e +$ $(n+5)t_m$	$(\|\mathcal{P}\|+4)t_e +$ $(\|\mathcal{P}\|+3)t_m + t_h$	2
[20]	√	composite order	$3 \cdot (n+4)t_e +$ $(3\|\overrightarrow{\mathcal{R}}\|+4)t_m$	$(25+6n-6\|\overrightarrow{\mathcal{R}}\|)t_e +$ $(18+4n-4\|\overrightarrow{\mathcal{R}}\|)t_m$	$(\|\mathcal{P}\|+4)t_e +$ $(\|\mathcal{P}\|+4)t_m$	3
Ours & Scheme-I	√	composite order	$3 \cdot (n+5)t_e +$ $(3\|\overrightarrow{\mathcal{R}}\|+4)t_m$	$(31+6n-6\|\overrightarrow{\mathcal{R}}\|)t_e +$ $(23+4n-4\|\overrightarrow{\mathcal{R}}\|)t_m$	$(\|\mathcal{P}\|+5)t_e +$ $(\|\mathcal{P}\|+4)t_m$	3
Ours & Scheme-II	√	composite order	$(n+6)t_e +$ $(2n+7)t_m$	$(8+2n-2\|\overrightarrow{\mathcal{R}}\|)t_e +$ $(2+3n-3\|\overrightarrow{\mathcal{R}}\|)t_m$	$(\|\mathcal{P}\|+3)t_e +$ $(\|\mathcal{P}\|+3)t_m$	2
Ours & Improved	√	composite order	$\|\|\overrightarrow{\mathcal{R}}\| \cdot t_m$	$1 \cdot t_m$	$(\|\mathcal{P}\|+1)t_m$	3



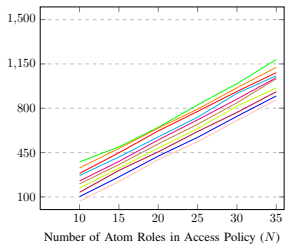
(a) Secret key generation time (ms)



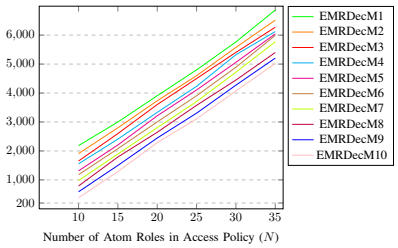
(b) Secret key delegation time (ms)



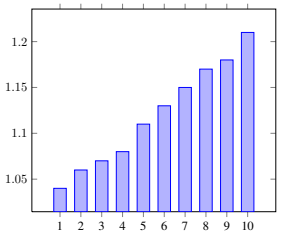
(c) EMR encapsulation time (ms)



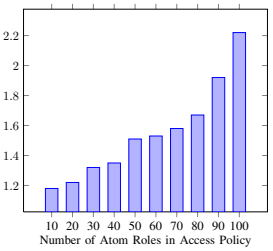
(d) Scheme-I: EMR decapsulation (ms)



(e) Scheme-II: EMR decapsulation (ms)



(f) Improved KeyGenM time (ms)



(g) Improved encapsulation time (ms)

FIGURE 2: Experimental results for the proposed system

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2810243, IEEE Access

*Assumption*2. Given group generator $\mathcal{G}$, we define the following distribution:

$$\mathbb{G} = (N = p_1p_2p_3p_4, G, G_T, e) \xleftarrow{R} \mathcal{G},$$
$$g_1, A_1 \xleftarrow{R} G_{p_1}, A_2, B_2 \xleftarrow{R} G_{p_2}, g_3, B_3 \xleftarrow{R} G_{p_3}, g_4 \xleftarrow{R} G_{p_4}$$
$$D = (\mathbb{G}, g_1, g_3, g_4, A_1A_2, B_2B_3)$$

Then, this assumption determines whether the given element $T \xleftarrow{R} G_{p_1p_2p_3}$ or $T \xleftarrow{R} G_{p_1p_3}$.

The advantage of an algorithm $\mathcal{A}$ that outputs $\beta \in \{0,1\}$ in breaking Assumption 2 is defined as

$$Adv2_{\mathcal{A}}(\lambda) = \left| \begin{array}{l} \Pr\left[\mathcal{A}\left(D, T \xleftarrow{R} G_{p_1p_2p_3}\right) = 1\right] \\ - \Pr\left[\mathcal{A}\left(D, T \xleftarrow{R} G_{p_1p_3}\right) = 1\right] \end{array} \right| - \frac{1}{2}$$

Definition II. $\mathcal{G}$ satisfies Assumption 2 if $Adv2_{\mathcal{A}}(\lambda)$ is negligible for any polynomial-time algorithm $\mathcal{A}$.

*Assumption*3. Given group generator $\mathcal{G}$, we define the following distribution:

$$\mathbb{G} = (N = p_1p_2p_3p_4, G, G_T, e) \xleftarrow{R} \mathcal{G},$$
$$\alpha, s, r \xleftarrow{R} Z_N, g_1 \xleftarrow{R} G_{p_1}, g_2, A_2, B_2 \xleftarrow{R} G_{p_2}, g_3 \xleftarrow{R} G_{p_3},$$
$$g_4 \xleftarrow{R} G_{p_4}, D = (\mathbb{G}, g_1, g_2, g_3, g_4, g_1^{\alpha}A_2, g_1^{s}B_2, g_2^{r}, A_2^{r})$$

Then, this assumption determines whether the given element $T \leftarrow e(g_1, g_1)^{\alpha s}$ or $T \leftarrow G_T$.

The advantage of an algorithm $\mathcal{A}$ that outputs $\beta \in \{0,1\}$ in breaking Assumption 3 is defined as

$$Adv3_{\mathcal{A}}(\lambda) = \left| \begin{array}{l} \Pr\left[\mathcal{A}\left(D, T \leftarrow e(g_1, g_1)^{\alpha s}\right) = 1\right] \\ - \Pr\left[\mathcal{A}\left(D, T \leftarrow G_T\right) = 1\right] \end{array} \right| - \frac{1}{2}$$

Definition III. $\mathcal{G}$ satisfies Assumption 3 if $Adv3_{\mathcal{A}}(\lambda)$ is negligible for any polynomial-time algorithm $\mathcal{A}$.

*Assumption*4. Given group generator $\mathcal{G}$, we define the following distribution:

$$\mathbb{G} = (N = p_1p_2p_3p_4, G, G_T, e) \xleftarrow{R} \mathcal{G},$$
$$s, \hat{r} \xleftarrow{R} Z_N, g_1, U, A_1 \xleftarrow{R} G_{p_1}, g_2, A_2, B_2, D_2, F_2 \xleftarrow{R} G_{p_2},$$
$$g_3 \xleftarrow{R} G_{p_3}, g_4, A_4, B_4, D_4 \xleftarrow{R} G_{p_4}, A_{24}, B_{24}, D_{24} \xleftarrow{R} G_{p_2p_4}$$
$$D =$$
$$(\mathbb{G}, g_1, g_2, g_3, g_4, U, U^{s}A_{24}, U^{\hat{r}}, A_1A_4, A_1^{\hat{r}}A_2, g_1^{\hat{r}}B_2, g_1^{s}B_{24})$$

Then, this assumption determines whether the given element $T \leftarrow A_1^{s}D_{24}$ or $T \leftarrow G_{p_1p_2p_4}$.

The advantage of an algorithm $\mathcal{A}$ that outputs $\beta \in \{0,1\}$ in breaking Assumption 4 is defined as

$$Adv4_{\mathcal{A}}(\lambda) = \left| \begin{array}{l} \Pr\left[\mathcal{A}\left(D, T \leftarrow A_1^{s}D_{24}\right) = 1\right] \\ - \Pr\left[\mathcal{A}\left(D, T \leftarrow G_{p_1p_2p_4}\right) = 1\right] \end{array} \right| - \frac{1}{2}$$

Definition IV. $\mathcal{G}$ satisfies Assumption 4 if $Adv4_{\mathcal{A}}(\lambda)$ is negligible for any polynomial-time algorithm $\mathcal{A}$.

Now, we provide the proof showing that Game_{real}, $\text{Game}_{real'}$, $\text{Game}_{restrict}$, Game_k, $\text{Game}_{final'}$, and Game_{final} are indistinguishable from each other.

Lemma 5.1. For any algorithm $\mathcal{A}$, it holds that $\text{Game}_{Real}Adv_{\mathcal{A}} = \text{Game}_{Real'}Adv_{\mathcal{A}}$.

Proof. We note that the secret keys are identically distributed whether they are generated by the key generation algorithm or by the key delegation algorithm. Therefore, there is no difference between $\text{Game}_{Real}Adv_{\mathcal{A}}$ and $\text{Game}_{Real'}Adv_{\mathcal{A}}$ from the adversary's perspective.

Lemma 5.2. Suppose that there is a PPT algorithm $\mathcal{A}$ such that $\text{Game}_{Real}Adv_{\mathcal{A}}$ - $\text{Game}_{Restricted}Adv_{\mathcal{A}}$= ϵ_1. We can build a PPT algorithm $\mathcal{B}$ with the advantage $\epsilon_1/3$ in breaking Assumption 1.

Proof. If there exists an adversary $\mathcal{A}$ who can distinguish $\text{Game}_{Restricted}$ from Game_{Real} with advantage ϵ_1, then based on the definition of $\text{Game}_{Restricted}$, $\mathcal{A}$ knows that it submits its own secret key query for the role $\overrightarrow{\mathcal{R}} = (\mathcal{R}_1, \mathcal{R}_2, \cdots, \mathcal{R}_d)$ from others satisfying $\exists\ \overrightarrow{\mathcal{R}^*} = (\mathcal{R}_1^*, \mathcal{R}_2^*, \cdots, \mathcal{R}_{d'}^*) \in Pref(\mathcal{P}^*)$ with $d' \leq d$, $s.t.\ \forall i \in [1, d'], \mathcal{R}_i = \mathcal{R}_i^*\ mod\ p_2$. Then, the factor of N can be extracted by computing $gcd(\mathcal{R}_i - \mathcal{R}_i^*, N)$, from which we can build an algorithm similar to that introduced in the proof of Lemma 3.3 in [10] that can break Assumption 1 with advantage $\epsilon_1/3$. We omit the details to avoid repetition.

Lemma 5.3. Suppose that there is a PPT algorithm $\mathcal{A}$ such that $\text{Game}_{Restricted}Adv_{\mathcal{A}}$ - $\text{Game}_0Adv_{\mathcal{A}}$= ϵ_2. We can build a PPT algorithm $\mathcal{B}$ with the advantage ϵ_2 in breaking Assumption 1.

Proof. $\mathcal{B}$ receives $(\mathbb{G}, g_1, g_3, g_4, A_1A_2, B_2B_4, T)$ of Assumption 1, and it needs to determine whether T is in $G_{p_1p_4}$ or in $G_{p_1p_2p_4}$. $\mathcal{B}$ chooses random exponents $\alpha, \{a_i\}_{i\in[1,n]}, a, b, c \in Z_N$ and sets $Y_1 = g_1, Y_3 = g_4, Y_4 = g_3, X_4 = Y_4^{c}, X_1 = Y_1^{b}, u_P = Y_1^{a}$ and $u_i = Y_1^{a_i}$ for $i \in [1, n]$. Then, $\mathcal{B}$ gives the public key $PK = (N, Y_1, Y_3, Y_4, x = X_1X_4, \{u_i\}_{i\in[1,n]}, u_P, A = e(Y_1, Y_1)^{\alpha})$ to adversary $\mathcal{A}$. $\mathcal{B}$ knows the master key $MSK = (X_1, \alpha)$ and thus can answer all queries from $\mathcal{A}$ in the secret key query phase.

In the challenge phase, $\mathcal{A}$ sends $\mathcal{B}$ two equal-length EMRs EMR_0, EMR_1 with a challenge access policy $\mathcal{P}^*$ and a change identity ID^*. $\mathcal{B}$ flips a random coin $\beta \in \{0,1\}$ and returns the challenge ciphertext

$$\{C_1^*, C_2^*, En^*\} =$$
$$\{T^{\sum_{i\in\mathbb{I}^*} a_i\mathcal{R}_i^* + aID^* + b}, T, SymEnc(e(T, Y_1)^{\alpha}, EMR_{\beta})\}$$

If $G \xleftarrow{R} G_{p_1p_4}$, then T can be written as $Y_1^{s_1}Y_3^{s_3}$ with random $s_1, s_3 \leftarrow Z_N$. In this case, (C_1^*, C_2^*, En^*) is a normal ciphertext, and $\mathcal{B}$ simulates $\text{Game}_{Restricted}$. If $G \xleftarrow{R} G_{p_1p_2p_4}$, then T can be written as $Y_1^{s_1}g_2^{s}Y_3^{s_3}$. In this case, (C_1^*, C_2^*, En^*) is a semi-functional ciphertext according to its definition, and $\mathcal{B}$ simulates Game_0.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2810243, IEEE Access

Lemma 5.4. Suppose that there is a PPT algorithm $\mathcal{A}$ such that Game$_{k-1}Adv_{\mathcal{A}}$ - Game$_kAdv_{\mathcal{A}}$= ϵ_3. We can build a PPT algorithm $\mathcal{B}$ with the advantage ϵ_3 in breaking Assumption 2.

Setup. $\mathcal{B}$ receives $(\mathbb{G}, g_1, g_3, g_4, A_1A_2, B_2B_3, T)$ of Assumption 2, and it needs to determine whether T is in $G_{p_1p_3}$ or in $G_{p_1p_2p_3}$. $\mathcal{B}$ chooses random exponents $\alpha, \{a_i\}_{i\in[1,n]}, a, b, c \in Z_N$ and sets $Y_1 = g_1, Y_3 = g_3, Y_4 = g_4, X_4 = Y_4^c, X_1 = Y_1^b, u_P = Y_1^a$ and $u_i = Y_1^{a_i}$ for $i \in [1, n]$. Then, $\mathcal{B}$ gives the public key $PK = (N, Y_1, Y_3, Y_4, x = X_1X_4, \{u_i\}_{i\in[1,n]}, u_P, A = e(Y_1, Y_1)^\alpha)$ to adversary $\mathcal{A}$. The master key $MSK = (X_1, \alpha)$ is kept by $\mathcal{B}$.

Secret Key Query for Medical Staff. When $\mathcal{A}$ requests the l^{th} secret key for the medical staff member with the role $\overrightarrow{\mathcal{R}_l} = (\mathcal{R}_{l,1}, ..., \mathcal{R}_{l,d})$, where $I = \{i : \mathcal{R}_{l,i} \in S_{\overrightarrow{\mathcal{R}}}\}$, we need to consider three cases: $l < k$, $l > k$ and $l = k$.

- When $l < k$, $\mathcal{B}$ creates a semi-functional secret key. It selects random exponents $r_1, r_2, f, z, w, w_1, w_2, w_{P_1}, w_{P_2}$ and $\{w_{m,j}\}_{j\in[1,n]\setminus[I]}$ for $m \in \{1, 2\}$ from Z_N.

$$SK_d^{\overrightarrow{\mathcal{R}}} = \left\{ \begin{array}{l} Y_1^{r_1}(B_2B_3)^f, Y_1^\alpha(B_2B_3)^w(\prod_{i\in I} u_i^{\mathcal{R}_{l,j}}X_1)^{r_1}Y_3^{w_1} \\ \{u_j^{r_1}(B_2B_3)^{w_{1,j}}\}_{j\in[1,n]\setminus I}, u_P^{r_1}(B_2B_3)^{w_{P_1}} \end{array} \right\}$$

$$SK_r^{\overrightarrow{\mathcal{R}}} = \left\{ \begin{array}{l} Y_1^{r_2}(B_2B_3)^{zf}, (B_2B_3)^{zw}(\prod_{i\in I} u_i^{\mathcal{R}_{l,j}}X_1)^{r_2}Y_3^{w_2} \\ \{u_j^{r_2}(B_2B_3)^{w_{2,j}}\}_{j\in[1,n]\setminus I}, u_P^{r_2}(B_2B_3)^{w_{P_2}} \end{array} \right\}$$

  We consider B_2 as g_2^s for random $s \leftarrow Z_N$. It can be seen that the generated secret key is semi-functional because $\gamma = s \cdot f, \gamma \cdot z_k = s \cdot w$ and $\gamma \cdot z_{P_m} = s \cdot w_{P_m}$ for $m \in \{1, 2\}$.
- When $l > k$, $\mathcal{B}$ creates a normal secret key by calling the **KeyGenM** algorithm.
- When $l = k$, $\mathcal{B}$ creates the kth secret key. $\mathcal{B}$ lets $z_k = \sum_{i\in I} a_iR_{k,i} + b$, chooses random exponents $r_2', w_1, w_2, w_{P_1}, w_{P_2} \in Z_N$ and $\{w_{m,j}\} \in Z_N$ for $j \in [1, n]\setminus I$,$m \in \{1, 2\}$, and sets

$$SK_d^{\overrightarrow{\mathcal{R}}} = \{T, Y_1^\alpha T^{z_k}Y_3^{w_1}, \{T^{a_j}Y_3^{w_{1,j}}\}, T^aY_3^{w_{P_1}}$$
$$SK_r^{\overrightarrow{\mathcal{R}}} = \{T^{r_2'}, T^{r_2'\cdot z_k}Y_3^{w_2}, \{T^{r_2'\cdot a_j}Y_3^{w_{2,j}}\}, T^{r_2'\cdot a}Y_3^{w_{P_2}}\}$$

  If $T \xleftarrow{R} G_{p_1p_3}$, then all components in this secret key are in $G_{p_1p_3}$. Hence, it is a normal secret key. If $T \xleftarrow{R} G_{p_1p_2p_3}$, then T can be written as $Y_1^{r_1'}g_2^sY_3^{r_3}$ with $s, r_3 \in Z_N$. Hence, it is a semi-functional secret key with $\gamma = s, z = r_2', r_1 = r_1', r_2 = r_1'r_2'$.

Secret Key Query for Patient. When $\mathcal{A}$ requests the l^{th} secret key for the patient with identity ID_l, we need to consider three cases: $l < k$, $l > k$ and $l = k$.

- When $l < k$, $\mathcal{B}$ creates a semi-functional secret key. It does this by selecting random exponents $r_1', w, \widetilde{w}, \{w_j\}_{j\in[1,n]}$ from Z_N.

$$SK^{ID} = \left\{ \begin{array}{l} TY_1^{r_1'}(B_2B_3)^f, Y_1^\alpha(B_2B_3)^w(u_P^{ID_l}X_1)^{r_1'}Y_3^{\widetilde{w}}, \\ \{u_j^{r_1'}(B_2B_3)^{w_j}\}_{j\in[1,n]} \end{array} \right\}$$

  We consider B_2 as g_2^s for random $s \leftarrow Z_N$. It can be seen that the generated secret key is semi-functional because $\gamma = s\cdot f, \gamma\cdot\widetilde{z}_k = s\cdot w$ and $\{\gamma\cdot\widetilde{z}_j = s\cdot w_j\}_{j\in[1,n]}$.
- When $l > k$, $\mathcal{B}$ creates a normal secret key by calling the **KeyGenP** algorithm.
- When $l = k$, $\mathcal{B}$ creates the kth secret key. $\mathcal{B}$ lets $\widetilde{z}_k = a\cdot ID$, chooses random exponents $w, \{w_j\}_{j\in[1,n]} \in Z_N$, and sets $SK^{ID} = \{T, Y_1^\alpha T^{\widetilde{z}_k}Y_3^w, \{T^{a_j}Y_3^{w_j}\}_{j\in[1,n]}\}$. If $T \xleftarrow{R} G_{p_1p_3}$, then all components in this secret key are in $G_{p_1p_3}$. Hence, it is a normal secret key. If $T \xleftarrow{R} G_{p_1p_2p_3}$, then T can be written as $Y_1^{r_1'}g_2^sY_3^{r_3}$ with $s, r_3 \in Z_N$. Hence, it is a semi-functional secret key with $\gamma = s, r_1' = r_1'$.

Challenge. $\mathcal{A}$ sends $\mathcal{B}$ two equal-length EMRs EMR_0, EMR_1 with a challenge access policy $\mathcal{P}^*$ and a change identity ID^*. $\mathcal{B}$ flips a random coin $\beta \in \{0, 1\}$ and returns to $\mathcal{A}$ the challenge ciphertext $C^* = \{C_1^*, C_2^*, En^*\}$, where $C_1^* = (A_1A_2)^{\sum_{i\in\mathbb{I}^*} a_i\mathcal{R}_i^* + aID^* + b}Y_4^z, A_1A_2Y_4^{z'}$, $C_2^* = SymEnc(e(A_1A_2, Y_1)^\alpha$, $C_3^* = EMR_\beta)$, and $\mathbb{I}^* = \{i : \mathcal{R}_i^* \in S_{\mathcal{P}^*}\}$.

This ciphertext is semi-functional, with $z_c = \sum_{i\in\mathbb{I}^*} a_i\mathcal{R}_i + aID^* + b$. Since the role associated with the kth secret key for the medical staff is not a prefix of the challenge role $\mathcal{R}^*$ modulo p_2 and the identity associated with the kth secret key for the patient is not the challenge identity ID^* modulo p_2, $z_k + \widetilde{z}_k$ and z_c will appear to be randomly distributed to adversary $\mathcal{A}$. If $\mathcal{B}$ tries to test whether the kth key is semi-functional or not via the above procedure by creating a semi-functional ciphertext for $\overrightarrow{\mathcal{R}_k} \in Pref(\mathcal{P})$ and ID_k, then we will have $z_c = z_k + \widetilde{z}_k + \sum_{i\in\mathbb{I}\setminus I} a_i\mathcal{R}_i$, where $I = \{i : \mathcal{R}_i \in S_{\overrightarrow{\mathcal{R}}}\}$ and $\mathbb{I} = \{i : \mathcal{R}_i \in S_{\mathcal{P}}\}$; thus, the decryption also works. In other words, simulator $\mathcal{B}$ can create only a nominally semi-functional key for the kth key query.

Guess. If $T \xleftarrow{R} G_{p_1p_3}$, all components in the kth secret key generated by $\mathcal{B}$ are in $G_{p_1p_3}$. Hence, it is a normal secret key. In this case, $\mathcal{B}$ simulates Game$_{k-1}$. Otherwise, $T \xleftarrow{R} G_{p_1p_2p_3}$; hence, the kth secret key is semi-functional. In this case, $\mathcal{B}$ simulates Game$_k$. If $\mathcal{A}$ has the advantage ϵ_3 when distinguishing the two games, $\mathcal{B}$ can distinguish $T \xleftarrow{R} G_{p_1p_3}$ from $T \xleftarrow{R} G_{p_1p_2p_3}$ with advantage ϵ_3.

Lemma 5.5. Suppose that there is a PPT algorithm $\mathcal{A}$ such that Game$_qAdv_{\mathcal{A}}$ - Game$_{final'}Adv_{\mathcal{A}}$= ϵ_4. We can build a PPT algorithm $\mathcal{B}$ with the advantage ϵ_4 in breaking Assumption 3.

Setup. $\mathcal{B}$ receives $(\mathbb{G}, g_1, g_2, g_3, g_4, g_1^\alpha A_2, g_1^sB_2, g_2^r, A_2^r, T)$ of Assumption 3, and it determines whether $T \leftarrow e(g_1, g_1)^{\alpha s}$ or $T \xleftarrow{R} G_T$. $\mathcal{B}$ chooses random exponents $\{a_i\}_{i\in[1,n]}, a, b, c \in Z_N$ and sets $Y_1 = g_1, Y_3 = g_3, Y_4 = g_4, X_4 =$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2810243, IEEE Access

$Y_4^c, X_1 = Y_1^b, u_P = Y_1^a$ and $u_i = Y_1^{a_i}$ for $i \in [1,n]$. Then, $\mathcal{B}$ gives the public key $PK = (N, Y_1, Y_3, Y_4, x = X_1X_4, \{u_i\}_{i\in[1,n]}, u_P, A = e(g_1^\alpha A_2, Y_1) = e(Y_1, Y_1)^\alpha)$ to adversary $\mathcal{A}$. The master key $MSK = (X_1, \alpha)$ is kept by $\mathcal{B}$.

Secret Key Query for Medical Staff. When $\mathcal{A}$ requests a secret key for the medical staff member with the role $\overrightarrow{\mathcal{R}} = (\mathcal{R}_1, ..., \mathcal{R}_d)$, where $I = \{i : \mathcal{R}_i \in S_{\overrightarrow{\mathcal{R}}}\}$, $\mathcal{B}$ creates a semi-functional key by choosing random exponents $r_1, r_2, z, z', z_{P_1}, z_{P_2}, w_{P_1}, w_{P_2} \in Z_N$ and $w_{m,1}, w_{m,2}, \{w_{m,j}, z_{m,j}\}_{j\in[1,n]\setminus I} \in Z_N$ for $m \in \{1,2\}$.

$$SK_d^{\overrightarrow{\mathcal{R}}} = \left\{ \begin{array}{c} Y_1^{r_1} g_2^z Y_3^{w_{1,1}}, (g_1^\alpha A_2) g_2^{z'} (\prod_{i\in I} u_i^{\mathcal{R}_{l,j}} X_1)^{r_1} Y_3^{w_{1,2}} \\ \{u_j^{r_1} g_2^{z_{1,j}} Y_3^{w_{1,j}}\}_{j\in[1,n]\setminus I}, u_P^{r_1} g_2^{z_{P_1}} Y_3^{w_{P_1}} \end{array} \right\}$$

$$SK_r^{\overrightarrow{\mathcal{R}}} = \left\{ \begin{array}{c} Y_1^{r_2} (g_2^r)^z Y_3^{w_{2,1}}, A_2^r (g_2^r)^{z'} (\prod_{i\in I} u_i^{\mathcal{R}_{l,j}} X_1)^{r_2} Y_3^{w_{2,2}} \\ \{u_j^{r_2} g_2^{z_{2,j}} Y_3^{w_{2,j}}\}_{j\in[1,n]\setminus I}, u_P^{r_2} g_2^{z_{P_2}} Y_3^{w_{P_2}} \end{array} \right\}$$

We note that this secret key is semi-functional.

Secret Key Query for Patient. When $\mathcal{A}$ requests a secret key for the patient with identity ID, $\mathcal{B}$ creates a semi-functional key by choosing random exponents $r_1, z, z', \widetilde{w}_1, \widetilde{w}_2 \in Z_N$ and $\{w_j, z_j\}_{j\in[1,n]} \in Z_N$.

$$SK^{ID} = \left\{ \begin{array}{c} Y_1^{r_1} g_2^z Y_3^{\widetilde{w}_1}, (g_1^\alpha A_2) g_2^{z'} (u_P^{ID} X_1)^{r_1} Y_3^{\widetilde{w}_2}, \\ \{u_j^{r_1} g_2^{z_{1,j}} Y_3^{w_{1,j}}\}_{j\in[1,n]} \end{array} \right\}$$

We note that this secret key is semi-functional.

Challenge. $\mathcal{A}$ sends $\mathcal{B}$ two equal-length EMRs EMR_0, EMR_1 with a challenge access policy $\mathcal{P}^*$ and a change identity ID^*. $\mathcal{B}$ flips a random coin $\beta \in \{0,1\}$ and returns to $\mathcal{A}$ the semi-functional ciphertext $CT^* = \{C_1^*, C_2^*, En^*\}$, where $C_1^* = (g_1^s B_2)^{\sum_{i\in\mathbb{I}^*} a_i \mathcal{R}_i^* + aID^* + b} Y_4^z$, $C_2^* = g_1^s B_2 Y_4^{z'}$, $C_3^* = SymEnc(T, EMR_\beta)$, and $\mathbb{I}^* = \{i : \mathcal{R}_i^* \in S_{\mathcal{P}^*}\}$. We implicitly set $z_c = \sum_{i\in\mathbb{I}^*} a_i \mathcal{R}_i^* + aID^* + b$.

Guess. If $T \leftarrow e(g_1, g_1)^{\alpha s}$, then $\mathcal{B}$ simulates Game_q since CT^* is a semi-functional ciphertext of the EMR EMR_β. If $T \xleftarrow{R} G_T$, CT^* is a semi-functional ciphertext of a random message that is independent of EMR_β. In this case, $\mathcal{B}$ simulates $\text{Game}_{final'}$. Hence, if $\mathcal{A}$ has the advantage ϵ_4 in distinguishing Game_q and $\text{Game}_{final'}$, then $\mathcal{B}$ has the advantage ϵ_4 in distinguishing the distribution of T.

Lemma 5.6. Suppose that there exists a PPT algorithm $\mathcal{A}$ such that $\text{Game}_{final'} Adv_{\mathcal{A}}$ - $\text{Game}_{final} Adv_{\mathcal{A}} = \epsilon_5$. Then, we can build a polynomial-time algorithm $\mathcal{B}$ with the advantage ϵ_5 in breaking Assumption 4.

Proof. Assume that we are simulating the games for an adversary who can distinguish a ciphertext of the challenge EMR with the challenge access policy set $\mathcal{P}^*$ and the challenge patient identity ID^* from a ciphertext of the challenge EMR with a random access policy set and a random patient's identity.

Setup. $\mathcal{B}$ receives $(\mathbb{G}, g_1, g_2, g_3, g_4, U, U^s A_{24}, U^{\hat{r}}, A_1 A_4, A_1^{\hat{r}} A_2, g_1^{\hat{r}} B_2, g_1^s B_{24}, T)$ of Assumption 4, and it needs to determine whether $T \leftarrow A_1^s D_{24}$ or $T \xleftarrow{R} G_{p_1p_2p_4}$. $\mathcal{B}$ chooses random exponents $\{a_i\}_{i\in[1,n]}, a \in Z_N$ and sets $Y_1 = g_1, Y_3 = g_3, Y_4 = g_4, x = A_1A_4, u_P = U^a$ and $u_i = U^{a_i}$ for $i \in [1,n]$. $\mathcal{B}$ gives public key $PK = (N, Y_1, Y_3, x, \{u_i\}_{i\in[1,n]}, u_P, A = e(Y_1, Y_1)^\alpha)$ to $\mathcal{A}$.

Secret Key Query for Medical Staff. When $\mathcal{A}$ requests a secret key for the medical staff member with the role $\overrightarrow{\mathcal{R}} = (\mathcal{R}_{l,1}, ..., \mathcal{R}_{l,d})$, where $I = \{i : \mathcal{R}_i \in S_{\overrightarrow{\mathcal{R}}}\}$, $\mathcal{B}$ creates a semi-functional key by choosing random exponents $r_1, r_2, w_{P_1}, w_{P_2}, z_{P_1}, z_{P_2} \in Z_N$ and $w_{m,1}, w_{m,2}, \{w_{m,j}, z_{m,j}\}_{j\in[1,n]\setminus I} \in Z_N$ for $m \in \{1,2\}$.

$$SK_d^{\overrightarrow{\mathcal{R}}} = \left\{ \begin{array}{c} (g_1^{\hat{r}} B_2)^{r_1} Y_3^{w_{1,1}}, Y_1^\alpha ((U^{\hat{r}})^{\sum_{i\in I} a_i \mathcal{R}_i} (A_1^{\hat{r}} A_2))^{r_1} Y_3^{w_{1,2}} \\ \{(U^{\hat{r}})^{r_1 a_j} Y_2^{z_{1,j}} Y_3^{w_{1,j}}\}_{j\in[1,n]\setminus I}, (U^{\hat{r}})^{r_1 a} Y_2^{z_{P_1}} Y_3^{w_{P_1}} \end{array} \right\}$$

$$SK_r^{\overrightarrow{\mathcal{R}}} = \left\{ \begin{array}{c} (g_1^{\hat{r}} B_2)^{r_2} Y_3^{w_{2,1}}, Y_1^\alpha ((U^{\hat{r}})^{\sum_{i\in I} a_i \mathcal{R}_i} (A_1^{\hat{r}} A_2))^{r_2} Y_3^{w_{2,2}} \\ \{(U^{\hat{r}})^{r_2 a_j} Y_2^{z_{2,j}} Y_3^{w_{2,j}}\}_{j\in[1,n]\setminus I}, (U^{\hat{r}})^{r_2 a} Y_2^{z_{P_2}} Y_3^{w_{P_2}} \end{array} \right\}$$

We note that this secret key is semi-functional.

Secret Key Query for Patient. When $\mathcal{A}$ requests a key for the patient with identity ID, $\mathcal{B}$ creates a semi-functional key by choosing exponents $r_1, \widetilde{w}_1, \widetilde{w}_2$ and $\{w_j, z_j\}_{j\in[1,n]} \in Z_N$.

$$SK^{ID} = \left\{ \begin{array}{c} (g_1^{\hat{r}} B_2)^{r_1} Y_3^{\widetilde{w}_1}, Y_1^\alpha ((U^{\hat{r}})^{a\cdot ID} (A_1^{\hat{r}} A_2))^{r_1} Y_3^{\widetilde{w}_2}, \\ \{(U^{\hat{r}})^{r_1 a_j} Y_2^{z_j} Y_3^{w_j}\}_{j\in[1,n]} \end{array} \right\}$$

We note that this secret key is semi-functional.

Challenge. $\mathcal{A}$ sends $\mathcal{B}$ two equal-length EMRs EMR_0, EMR_1 with a challenge access policy $\mathcal{P}^*$ and a change identity ID^*. $\mathcal{B}$ chooses a random $En^* \in G_T$, flips a random coin $\beta \in \{0,1\}$, and returns to $\mathcal{A}$ the semi-functional ciphertext $T^* = \{C_1^*, C_2^*, En^*\}$ as

$$\{T(U^s A_{24})^{\sum_{i\in\mathbb{I}^*} a_i \mathcal{R}_i^* + aID^*}, g_1^s B_{24}, En^*\}$$

where $\mathbb{I}^* = \{i : \mathcal{R}_i^* \in S_{\mathcal{P}^*}\}$.

Guess. If $T \leftarrow A_1^s D_{24}$, then the adversary $\mathcal{A}$ generates a semi-functional ciphertext of a random message En^* and a header Hdr^* under the challenge access policy $\mathcal{P}^*$ and the challenge identity ID^*. In this case, $\mathcal{B}$ simulates $\text{Game}_{final'}$. If $T \xleftarrow{R} G_{p_1p_2p_4}$, the adversary $\mathcal{A}$ generates a semi-functional ciphertext of a random message En^* and a header Hdr^* under the implicit random access policy set and a random patient's identity. In this case, $\mathcal{B}$ simulates Game_{final}.

If $\mathbb{G}$ satisfies the four assumptions with advantages $\epsilon_1', \epsilon_2', \epsilon_3', \epsilon_4'$, and ϵ_5', then the above lemmas show that no PPT adversary can distinguish Game_{real} and Game_{final} with advantage $3\epsilon_1' + \epsilon_2' + \epsilon_3' + \epsilon_4' + \epsilon_5'$. The ciphertext of $\text{Game}_{final'}$ does not leak any information regarding the EMR data since the component corresponding to the EMR in the ciphertext is

a random group element. The ciphertext of Game$_{final}$ does not leak any information regarding the roles of the medical staff and the identity of the patient since the components corresponding to the roles and identity in the ciphertext are random group elements.

REFERENCES

[1] M. J. Atallah, M. Blanton, and K. B. Frikken, "Dynamic and efficient key management for access hierarchies," ACM Trans. Inf. Syst. Secur., vol. 12, no. 3, 2009.
[2] 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 2012. IEEE, 2012, pp. 279–287.
[3] M. C. Mont, P. Bramhall, and K. Harrison, "A flexible role-based secure messaging service: Exploting ibe technology for privacy in health care," IEEE Computer Society, vol. 432, 2003.
[4] J. A. Akinyele, M. W. Pagano, M. D. Green, C. U. Lehmann, Z. N. Peterson, and A. D. Rubin, "Securing electronic medical records using attribute-based encryption on mobile devices," in SPSM 2011. ACM, 2011, pp. 75–86.
[5] S. Narayan and R. Safavi-Naini, "Privacy preserving ehr system using attribute-based infrastructure," in CCSW'10. ACM, 2010, pp. 47–52.
[6] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, "Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption," IEEE Trans. Parallel Distrib. Syst., vol. 24, no. 1, pp. 131–143, 2013.
[7] M. Sicuranza, A. Esposito, and M. Ciampi, "A view-based acces control model for EHR systems," in IDC 2014. Springer, 2014, pp. 443–452.
[8] W. Liu, X. Liu, J. Liu, Q. Wu, J. Zhan, and Y. Li, "Auiting and revocation enabled role-based access contrl over outsourced private ehrs," in HPCC 2015. IEEE, 2015, pp. 336–341.
[9] D. Boneh, E. J. Goh, and K. Nissim, "Evaluating 2-dnf formulas on ciphertexts," in TCC 2005. Springer, 2005, pp. 325–341.
[10] A. D. Caro, V. Iovino, and G. Persiano, "Fully secure anonymous hibe and secret-key anonymous ibe with short ciphertexts," in International Conference on Pairing-Based Cryptography. Springer, 2010, pp. 347–366.
[11] R. J. Anderson, "Technical perspective - A chilly sense of security," Commun. ACM, vol. 52, no. 5, p. 90, 2009.
[12] Centers for Medicare & Medicaid Services, "Health insurance portability and accountability act," 1996.
[13] C. I. of Health Research, 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. Canadian Institutes of Health Research, 2001.
[14] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in CCS 2006. ACM, 2006, pp. 89–98.
[15] L. Guo, C. Zhang, J. Sun, and Y. Fang, "Paas: A privacy-preserving attribute-based authentication system for ehealth networks," in ICDCS 2012. IEEE, 2012, pp. 224–233.
[16] J. Sedayao, "Enhancing cloud security using data anonymization," White Paper, Intel Coporation, 2012.
[17] T. Jung, X. Li, Z. Wan, and M. Wan, "Control cloud data access privilege and anonymity with fully anonymous attribute-based encryption," IEEE Trans. Inf. Forensics Security, vol. 10, no. 1, pp. 190–199, 2015.
[18] S. Sabitha and M. Rajasree, "Anonymous-cpabe: Privacy preserved content disclosure for data sharing in cloud," in ARCS 2015. Springer, 2015, pp. 146–157.
[19] X. Zhou, J. Liu, W. Liu, and Q. Wu, "Anonymous role-based access control on e-health records," in AsiaCCS 2016. ACM, 2016, pp. 559–570.
[20] J. H. Seo, T. Kobayashi, M. Ohkubo, and K. Suzuki, "Anonymous hierarchical identity-based encryption with constant size ciphertexts," in PKC 2009. Springer, 2009, pp. 215–234.
[21] A. Lewko and B. Waters, "New proof methods for attribute-based encryption: Achieving full security through selective techniques," in CRYPTO 2012. Springer, 2012, pp. 180–198.
[22] D. B. X. Boyen and E. J. Goh, "Hierarchical identity based encryption with constant size ciphertext," in EUROCRYPT 2005. Springer, 2005, pp. 440–456.
[23] M. H. Ameri, J. Mohajeri, and M. Salmasizadeh, "Efficient and provable secure anonymous hierarchical identity-based broadcast encryption (hibbe) scheme without random oracle," ia.cr/2016/780, 2016.
[24] A. Lewko and B. Waters, "New techniques for dual system encryption and fully secure hibe with short ciphertexts," in TCC 2010. Springer, 2010, pp. 455–479.
[25] S. Hohenberger and B. Waters, "Online/offline attribute-based encryption," in PKC 2014. Springer, 2014, pp. 293–310.
[26] E. Shimon, O. Goldreich, and S. Micali, "On-line/off-line digital signatures," Cryptology, vol. 9, no. 1, pp. 35–67, 1996.
[27] Y. F. Wang, G. Yang, and Z. C. Z, "Identity-based online/offline encryption," Computer Technology and Development, vol. 51, no. 43, pp. 247–261, 2012.

XINGGUANG ZHOU is a Ph.D. candidate in the Department of Electronic and Information Engineering, Beihang University. Her research interests include information security and communication network security.

JIANWEI LIU is currently a full professor and party secretary in the Department of Electronic and Information Engineering, Beihang University. He received his Ph.D. from the Communication and Electronic System Department, Xidian University, in 1998. His research interests include wireless communication networks, cryptography, information security, communication network security, channel coding, and modulation technology.

QIANHONG WU is currently a full professor in the Department of Electronic and Information Engineering, Beihang University. He has served as a member of the ACISP committee and more than 10 international conference procedure committees. He received his Ph.D. degree in cryptography from Xidian University in 2005. His research interests include information security, security in big data and cloud computing, and blockchains.

ZONGYANG ZHANG is an assistant professor in the Department of Electronic and Information Engineering, Beihang University. He received his Ph.D. in computer software and theory from Shanghai Jiao Tong University in 2012. His research interests include public-key cryptography and blockchains.