



Received December 30, 2020, accepted January 11, 2021, date of publication January 18, 2021, date of current version January 26, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3052247

An Efficient Access Control Scheme With Outsourcing and Attribute Revocation for Fog-Enabled E-Health

JING ZHAO¹, PENG ZENG¹ , (Member, IEEE), AND
KIM-KWANG RAYMOND CHOO^{2,3} , (Senior Member, IEEE)

¹Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai 200062, China

²Department of Information Systems and Cyber Security, The University of Texas at San Antonio, San Antonio, TX 78249, USA

³Department of Electrical and Computer Engineering, The University of Texas at San Antonio, San Antonio, TX 78249, USA

Corresponding author: Peng Zeng (pzeng@sei.ecnu.edu.cn)


This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62072184 and Grant 61601129, in part by the National Key Research and Development Program of China under Grant 2017YFB0802302, in part by the NSFC-Zhejiang Joint Fund for the Integration of Industrialization and Informatization under Grant U1509219, in part by the Key Laboratory of Information Network Security of Ministry of Public Security (The Third Research Institute of Ministry of Public Security) under Grant C18603, and in part by the Shanghai Natural Science Foundation under Grant 17ZR1408400. The work of Kim-Kwang Raymond Choo was supported by the Cloud Technology Endowed Professorship.

ABSTRACT Fog computing is increasingly popular partly due to its capability to minimize data transfer and latency requirements, for example by moving some of the computational operations away from the cloud servers and closer to the users. To achieve fine-grained access control in fog-enabled application scenarios to guarantee data security and user privacy, one could use ciphertext-policy attribute-based encryption (CP-ABE). However, the lack of an effective mechanism to carry out access right revocation in conventional CP-ABE schemes limits the deployment of such schemes in practice. Thus, we propose an efficient CP-ABE scheme with attribute revocation capability, designed to construct a fine-grained access control system in fog-enabled E-health (referred to as AC-FEH). In our AC-FEH system, fog nodes undertake data encryption and decryption operations; thus, computational costs for data owners and users are minimized. In comparison to several other competing access control schemes based on CP-ABE, our AC-FEH system reduces the computational costs associated with encryption and decryption. We also prove the selective security of the underlying CP-ABE scheme under the intractability assumption of the q -parallel BDHE problem.

INDEX TERMS Fog computing, access control, CP-ABE, outsourcing, attribute revocation, large universe.

I. INTRODUCTION

Fog computing is a distributed computing infrastructure, which may comprise Internet of Things (IoT) devices and other systems [1] in order to carry out some of the computational operations (e.g., data analysis and aggregation) at the edge of the network. Benefits associated with fog computing include low latency (in comparison to cloud-based deployments), location awareness, wide geographic distribution, and data edge processing. An important fog-based application is the so-called fog-enabled E-health (FEH). In an FEH system, information can be collected from the patients' wearable, embedded and/or implantable devices

The associate editor coordinating the review of this manuscript and approving it for publication was Chien-Ming Chen .

(e.g., physiological data) and preprocessed at the fog nodes, prior to storing relevant information (e.g., aggregated medical information) in the patients' electronic health records (EHRs) at the cloud. Consequently, such a deployment model achieves savings in time, bandwidth and computational costs.

There is, however, a corresponding need to ensure the security and privacy of patients' EHRs, particularly given the sensitivity of such information. Attribute-based encryption (ABE) is one promising cryptographic primitive to support fine-grained access control in the ciphertext environment, and can potentially be used to ensure data security and user privacy in the FEH scenario. Specifically, ABE allows one-to-many encryption mode, which differs from conventional public-key encryption [2]. ABE can be broadly categorized into ciphertext-policy ABE (CP-ABE) and key-policy ABE

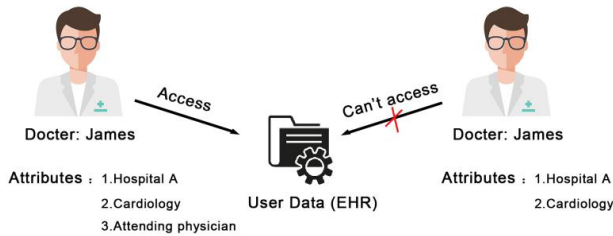


FIGURE 1. Attribute revocation.

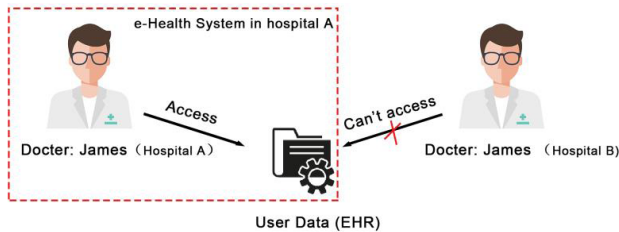


FIGURE 2. User revocation.

(KP-ABE). We posit that CP-ABE is a better choice for FEH deployment, since it supports a high-expression strategy and fine-grained access control on EHRs.

While CP-ABE is a relatively mature research area, and a large number of CP-ABE schemes with different properties (e.g., efficiency, expressibility, and security) have been proposed in the literature [3]–[8], these schemes may not be suitable for deployment in the FEH context unless they support access right revocation (e.g., attribute revocation and user revocation).

Fig. 1 describes the attribute revocation, where certain attributes owned by the user are deleted from the current collection so that the user loses access to the attribute [9]. Let Cindy denotes a patient suffering from a heart condition and James is a medical doctor and has attributes (Organization = Hospital A) AND (Department = Cardiology) AND (Occupation = Attending physician). Since the attributes of James satisfy the access policy, James can access Cindy’s EHR. If James is no longer the attending physician of Cindy, then the access right of James should be revoked (i.e., James loses the attribute Occupation = Attending physician). Fig. 2 describes the user revocation, where the user’s access right can be directly withdrawn from the cloud storage server in the event that the user is no longer part of the group. The revoked user cannot access any shared data in the system [10]. For example using the same example, if James resigns from hospital A to join hospital B, James’ access right to Cindy’s EHR should also be revoked.

It is generally considered that attribute revocation is more fine-grained than user revocation in the FEH scenario. However, the former topic is relatively under-studied in comparison to user revocation. This reinforces the importance of designing a secure, efficient and robust attribute revocation solution for E-health systems. If the system only supports user revocation, whenever a revocation occurs, the complexity in

redesigning the access control mechanism can be complex. In addition, since an attribute in CP-ABE may be shared by multiple users, a user’s single-attribute revocation may affect other users with the same attribute in the system. Hence, when user attributes are revoked, it can be to challenging to carry this out in practice, without affecting other users’ access and use of these revoked attributes. These challenges motivate our research in this article.

Specifically, we design a CP-ABE scheme with attribute revocation (AR-CPABE) and use it to construct an efficient access control scheme for FEH (AC-FEH). In the system architecture of our AC-FEH scheme, both data owners (e.g., patients) and data users (e.g., medical doctors) communicate via the fog nodes. Consequently, response time is reduced which leads to lower latency and higher quality of service. Our AC-FEH scheme employs proxy re-encryption (PRE) technology to realize attribute revocation. In particular, if an attribute of some user needs to be revoked, the access privileges of other users with the same attribute will not be affected. Our AC-FEH scheme is also capable of resisting collusion attacks from unauthorized users. The underlying AR-CPABE scheme also supports the large universe property, which enables the size of the common parameters to be independent of the number of attributes. At the same time, it supports the access structure of the linear secret sharing scheme (LSSS) to implement fine-grained access control.

The rest of this article is organized as follows. We will briefly review the related literature in Section II, prior to introducing the relevant preliminaries in Section III. The system architecture and security model are shown in Section IV. We present the concrete construction of our AC-FEH scheme in Section V. The security and efficiency analysis of our AC-FEH scheme are presented in Sections VI and VII, respectively. We conclude this article in Section VIII.

II. RELATED WORK

In 2006, Goyal *et al.* [11] elaborated on the concept of ABE and divided it into two categories: CP-ABE and KP-ABE. Bethencourt *et al.* [12] proposed a concrete CP-ABE scheme with a strong expression, which can provide fine-grained access control over ciphertext. Waters [13] proposed several CP-ABE schemes that support the access structure of LSSS. For the application aspect of ABE, Yu *et al.* [14] considered how to apply ABE technology in cloud computing. Guo *et al.* [15] designed a new architecture to control and search encrypted EHRs. In 2013, Li *et al.* [16] proposed a secure sharing scheme of EHRs in a cloud computing environment based on multiauthority ABE. Subsequently, there were several ABE schemes [17]–[24] designed for the cloud computing environment. Unfortunately, all these schemes either failed to provide revocability or needed a high computation cost for implementing this property. So user revocation and attribute revocation have become more and more concerned issues in one system.

User revocation is a common mechanism in ABE schemes. Attrapadung and Imai [25] and Junod and Karlov [26]

proposed ABE schemes with user revocation combined with broadcast encryption and ABE scheme. However, the schemes require the data owner to maintain a list of members, and data cannot be directly controlled once stored in a cloud storage system. Therefore, they are not suitable for cloud storage systems. Xu and Martin [27] proposed a dynamic revocation scheme for cloud storage systems. However, the cloud server is in charge of re-encrypting ciphertext by utilizing a delegation key, which makes it only achieve user revocation. So once a user's attribute is revoked, other users with the same attribute will lose access right. This is the same case with the schemes [28], [29], where user revocation is only performed on system-level. Although the above schemes enable user revocation, they are not ideal in many practical fine-grained access controls.

Recently, some ABE schemes with attribute revocation property have been proposed in order to achieve fine-grained access controls. Pirretti *et al.* [30] first introduced the timed rekeying mechanism, in which each attribute is associated with an expiration time. Bethencourt *et al.* [12] proposed a CP-ABE scheme in which the users are allowed to update their secret key frequently. Based on the tree structure, the three ABE schemes [31]–[33] achieved the attribute revocation function. However, the data manager in these schemes needs to re-encrypt all ciphertexts when an attribute is revoked, which incurs a high computation cost. Yang and Jia [34] proposed a CP-ABE scheme with attribute revocation based on the access structure of LSSS. This scheme has also a relatively high computation cost.

There are also some works proposed to reduce infrastructure cost and enhance dynamic resource tuning for resource-constrained mobile devices [35]–[37]. Zhang *et al.* [38] proposed a fully outsourced ABE scheme which reduces the communication cost by outsourcing the operations of key generation, encryption, and decryption. Fu *et al.* [35] constructed an access control scheme based on a large universe CP-ABE scheme, which outsources the decryption operation to the cloud server. Nonetheless, it suffers from the revocable problem. Zhang *et al.* [39] proposed a cloud-based access control scheme with user revocation and attribute update, but not suitable for the outsourcing computation framework. Further, Zhang *et al.* [40] proposed an efficient access control scheme with outsourcing and attribute update capacities based on the architecture of fog computing, but failed to realize efficiently the attribute revocation. Zu *et al.* [41] provided a large universe CP-ABE with the user and attribute revocation. Nevertheless, it is not suitable for mobile devices with limited resources.

Based on the fog computing framework and our newly proposed AR-CPABE scheme, an effective access control scheme—AC-FEH—is constructed for E-health systems. AC-FEH employs PRE technology to achieve the attribute revocation function. The main encryption and decryption operations are outsourced to fog nodes and thus the computation load of data owners and users can be greatly reduced. AC-FEH also supports the large universe property and the

TABLE 1. Some notations used in this article.

Notation	Description
$[\ell]$	The set $\{1, 2, \dots, \ell\}$
$\{s_i\}_{i \in [\ell]}$	Another representation of set $\{s_i \mid i \in [\ell]\}$
$ S $	The cardinality of set S
$a \xleftarrow{R} S$	Picking element a from S uniformly at random
$\vec{v} = (v_1, v_2, \dots, v_n)$	A row vector
\vec{v}^T	Transpose of \vec{v}
A_i	The i -th row of matrix A of order $\ell \times n$

access structure of LSSS to realize the fine-grained access control in ciphertext data.

III. PRELIMINARIES

Table. 1 lists some notations to be used in this article.

A. BASIC CONCEPTS

Definition 1 (Access Structure): Let $\mathbb{P} = \{P_1, P_2, \dots, P_n\}$ be a set of parties. A collection $\mathbb{A} \subseteq 2^{\mathbb{P}}$ is called monotone if $B \in \mathbb{A}$ and $B \subseteq C$ implies $C \in \mathbb{A}$. A (monotone) access structure \mathbb{A} is a (monotone) collection of non-empty subsets of \mathbb{P} , i.e. $\mathbb{A} \subseteq 2^{\mathbb{P}} \setminus \{\emptyset\}$. The sets in \mathbb{A} are called authorized sets, otherwise called unauthorized sets.

Definition 2 (Linear secret sharing scheme (LSSS) [42]): LSSS is often used to represent the access control strategy in the design of ABE schemes, which consists of the following two algorithms in general:

- 1) **Share** ($s, (A, \rho)$) $\rightarrow (\lambda_1, \lambda_2, \dots, \lambda_\ell)$: given as input a secret s to be shared and an access structure $\mathbb{A} = (A, \rho)$ with an ℓ by n share-generating matrix A and a mapping ρ from $[\ell] = \{1, 2, \dots, \ell\}$ to the attribute space, it generates n shares λ_i of s , $i \in [\ell]$. To this end, it first chooses $n - 1$ integers $r_i \xleftarrow{R} \mathbb{Z}_p$, $i = 2, 3, \dots, n$, and constructs a vector $\vec{v} = (s, r_2, \dots, r_n)$ of dimension n . Then for each $i \in [\ell]$, it computes $\lambda_i = A_i \cdot \vec{v}^T$ as the share belonging to the attribute $\rho(i)$, where A_i is the i -th row of A .
- 2) **Reconstruction** ($\lambda_1, \lambda_2, \dots, \lambda_\ell, (A, \rho)$) $\rightarrow s$: It retrieves the secret s based on the ℓ shares λ_i , $i \in [\ell]$, and the access policy (A, ρ) . Let S be an authorized set on (A, ρ) and $I_S = \{i \in [\ell] \mid \rho(i) \in S\}$. Then constants $\{\omega_i \in \mathbb{Z}_p\}_{i \in I_S}$ could be found in polynomial time s.t. $\sum_{i \in I_S} \omega_i \lambda_i = s$.

Definition 3 (Bilinear Groups): Let \mathbb{G} and \mathbb{G}_T be two multiplicative cyclic groups of prime order p and g a generator of \mathbb{G} . The five-tuple $(p, g, \mathbb{G}, \mathbb{G}_T, \hat{e})$ is called a bilinear group if the mapping $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ satisfies the following properties.

- 1) **Bilinearity:** $\hat{e}(u^a, v^b) = \hat{e}(u, v)^{ab}$, $\forall u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p$.
- 2) **Non-degeneracy:** $\hat{e}(g, g) \neq 1$.
- 3) **Computability:** $\hat{e}(u, v)$ is computable for any $u, v \in \mathbb{G}$.

In the following, BGen is adopted to represent the bilinear group generator.

B. COMPLEXITY ASSUMPTION

The security of our AR-CPABE scheme is based on the decisional q -parallel bilinear Diffie-Hellman exponent (BDHE) assumption which can be described as follows.

Definition 4 (q -parallel BDHE assumption [13]): Let λ be a security parameter and let $\text{BGGen}(\lambda) \rightarrow (p, g, \mathbb{G}, \mathbb{G}_T, e)$ be a bilinear group. Assume that $a, s, b_1, b_2, \dots, b_q$ are $q + 2$ elements chosen from \mathbb{Z}_p uniformly at random and $\vec{y} =$

$$\begin{aligned} &g, g^s, g^a, g^{a^2}, \dots, g^{a^q}, g^{a^{q+2}}, \dots, g^{a^{2q}}, \\ &g^{s \cdot b_j}, g^{a/b_j}, g^{a^2/b_j}, \dots, g^{a^q/b_j}, g^{a^{q+2}/b_j}, \dots, g^{a^{2q}/b_j}, \\ &1 \leq j \leq q, \\ &g^{a \cdot s \cdot b_i/b_j}, g^{a^2 \cdot s \cdot b_i/b_j}, \dots, g^{a^q \cdot s \cdot b_i/b_j}, \quad 1 \leq i \neq j \leq q. \end{aligned}$$

The decisional q -parallel BDHE assumption holds that if for any probabilistic polynomial-time (PPT) algorithm \mathcal{A} , the advantage $\text{Adv}_{\mathcal{A}}^q(\lambda)$ of \mathcal{A} distinguishing the element $\hat{e}(g, g)^{sa^{q+1}}$ from a random element $R \in \mathbb{G}_T$

$$\left| \Pr \left[\mathcal{A}(\vec{y}, \hat{e}(g, g)^{sa^{q+1}}) = 0 \right] - \Pr \left[\mathcal{A}(\vec{y}, R \xleftarrow{R} \mathbb{G}_T) = 0 \right] \right|$$

is negligible with respect to λ .

IV. SYSTEM ARCHITECTURE AND SECURITY MODEL

A. SYSTEM ARCHITECTURE

Our AC-FEH scheme employs a key/data encapsulation mechanism to encrypt the sensitive EHRs. In particular, the underlying AR-CPABE scheme of AC-FEH is applied to encrypt a symmetric session key \hat{K} and then the key \hat{K} is further used to encrypt the outsourced EHRs with an efficient symmetric encryption algorithm (e.g. AES). Fig. IV-A illustrates the system architecture of AC-FEH, which includes five entities: data owner (DO), data user (DU), cloud service provider (CSP), fog node (FN), and attribute authority (AA).

- **DO** refers to a patient who is the owner of EHRs obtained from different channels (e.g. digital-physical examination, digital laboratory examination, and digital pharmacy prescription, etc). DO can define an access structure \mathbb{A} and generate a template ciphertext with \mathbb{A} .
- **DU** may be a doctor or medical researcher who needs to access DO's EHR stored in CSP. DU has a set S of attributes and can decrypt the ciphertext of DO only if S satisfies the access policy \mathbb{A} embedded in the ciphertext.
- **CSP** is a powerful distributed computing facility and provides the service of data storage and management in ciphertext forms. When there are attributes needed to be revoked, CSP is responsible to re-encrypt the corresponding ciphertexts.
- **FN** denotes a semi-trusted third party located between DO/DU and CSP. FN undertakes the main tasks of encryption/decryption operations and is responsible for the upload/download of ciphertexts.
- **AA** is a fully trusted third party responsible for the generation of the system public parameters, the master key, and the private keys for system users.

AC-FEH consists of five phases: system initialization, private key generation, file encryption, re-encryption, and file decryption, which can be described as follows.

- 1) **System Initialization.** AA generates the public system parameters PP , the master key MSK and a delegation key SK_d with an initialization algorithm Setup . All system participants (DO, DU, FN, and CSP) are able to own PP .
- 2) **Private key Generation.** With an attribute set S submitted by some system user U , AA generates a secret key SK_u for U based on a key generation algorithm KeyGen .
- 3) **File Encryption.** When DO generates his/her EHR, he/she first selects a symmetric key \hat{K} and uses it to encrypt EHR with an efficient symmetric encryption algorithm ($\langle \text{EHR} \rangle_{\hat{K}}$ denotes the symmetric ciphertext). Then the key \hat{K} is encrypted by the algorithm Encrypt to generate an asymmetric ciphertext CT_0 . In our AR-CPABE scheme, Encrypt covers two sub-algorithms DO.Encrypt and FN.Encrypt , which are executed in turn by DO and FN. Finally, the ciphertexts $\langle \text{EHR} \rangle_{\hat{K}}$ and CT_0 are uploaded to CSP by FN.
- 4) **Re-encryption.** CSP re-encrypts the ciphertext CT_0 into the ciphertext CT_1 with the delegation key SK_d and a re-encryption algorithm ReEncrypt . This stage provides forward and backward security for attribute revocation.
- 5) **File Decryption.** If DU intends to access the health data EHR of DO, he/she informs some fog node FN to download the ciphertexts $\langle \text{EHR} \rangle_{\hat{K}}$ and CT_1 from CSP and execute the partial decryption on CT_1 with the decryption sub-algorithm FN.Decrypt . Then DU executes the final decryption on CT_1 with another decryption sub-algorithm DU.Decrypt . Further, the two sub-algorithms FN.Decrypt and DU.Decrypt form the decryption algorithm Decrypt in our AR-CPABE scheme. After getting the key \hat{K} from CT_1 , DU adopts it to decrypt the ciphertext $\langle \text{EHR} \rangle_{\hat{K}}$ to obtain EHR.

B. SECURITY MODEL

AC-FEH is mainly based on our new scheme AR-CPABE. Let $\mathcal{E} = (\text{Setup}, \text{KeyGen}, \text{Encrypt}, \text{ReEncrypt}, \text{Decrypt})$ be a CP-ABE scheme with attribute revocation. To define the selective security for \mathcal{E} , the following game is designed and $\text{Game}_{\mathcal{E}, \mathcal{A}}^{\text{SS}}$ is denoted, involving a PPT attacker \mathcal{A} and a PPT challenger \mathcal{C} (refer to Tu et al. [43]).

- 1) **Initialization.** For a given security parameter λ , the challenger \mathcal{C} runs the algorithm $\text{BGGen}(\lambda)$ to generate a bilinear group $(p, g, \mathbb{G}, \mathbb{G}_T, \hat{e})$. The adversary \mathcal{A} chooses a targeted access structure $\mathbb{A}^* = (A^*, \rho^*)$ and a revocation list R_y , where R_y is the set of all the users owning the revoked attribute y .
- 2) **Setup.** \mathcal{C} first runs the algorithm $\text{Setup}(\lambda)$ to generate the public parameter PP , the master key MSK and a

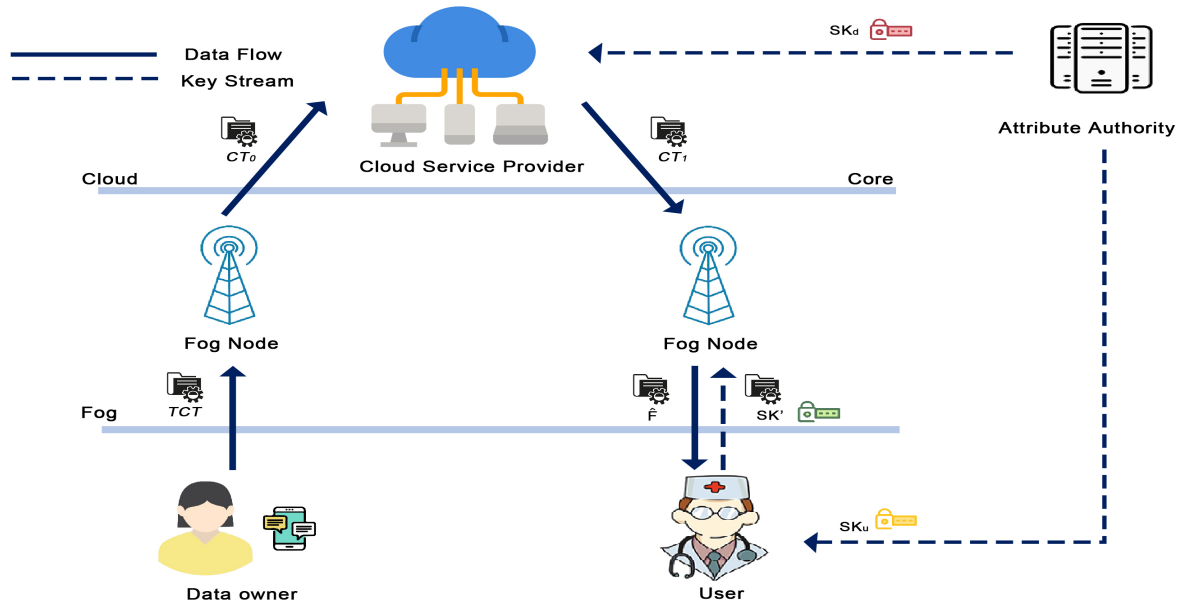


FIGURE 3. System architecture of AC-FEH.

delegation key SK_d . Then C sends PP and SK_d to A and the cloud service provider CSP, via secure channels, respectively.

- 3) **Phase 1.** For the private key query of A on each pair (ID_i, S_i) , $1 \leq i \leq q_1$, of identities and attribute sets, C runs the algorithm $\text{KeyGen}(PP, MSK, S_i)$ to generate the key SK_{u_i} for A . The only restriction is that there is no attribute set S_i (in the q_1 queries) satisfying the access policy \mathbb{A}^* . Here, we set $S'_i = S_i$ for $ID_i \notin R_y$, otherwise $S'_i = S_i \setminus \{y\}$ for $ID_i \in R_y$.
- 4) **Challenge.** A submits two messages \hat{M}_0 and \hat{M}_1 of equal length to C . Then C chooses a random coin $b \xleftarrow{R} \{0, 1\}$ and computes $\text{Encrypt}(PP, \mathbb{A}^*, \hat{M}_b) \rightarrow CT_0^{(b)}$ and $\text{ReEncrypt}(PP, CT_0^{(b)}, y^*, SK_d) \rightarrow CT_1^{(b)}$. Finally, C returns the challenge ciphertext $CT_1^{(b)}$ to A .
- 5) **Phase 2.** A continues to make the private key queries on the pairs (ID_i, S_i) , $q_1 + 1 \leq i \leq q$, of identities and attribute set as in Phase 1. Also, each attribute set S_i in this phase is not allowed to satisfy \mathbb{A}^* .
- 6) **Guess.** A outputs a guess bit b' of b . If $b' = b$, then A wins the game $\text{Game}_{\mathcal{E}, \mathcal{A}}^{\text{SS}}$ and we denote by $\text{Succ}_{\mathcal{E}, \mathcal{A}}^{\text{SS}}$ the event.

Definition 5 (Selective Security): We say that \mathcal{E} is of selective security if for any PPT adversary \mathcal{A} , the advantage

$$\text{Adv}_{\mathcal{E}, \mathcal{A}}^{\text{SS}}(\lambda) = \left| \Pr[\text{Succ}_{\mathcal{E}, \mathcal{A}}^{\text{SS}}] - 1/2 \right|$$

is always negligible with respect to λ .

V. OUR CONSTRUCTION

In this section, we present the concrete construction of our AC-FEH scheme. We mention that the algorithms Setup , KeyGen , Encrypt , ReEncrypt , and Decrypt involved in the AC-FEH scheme form our new CP-ABE scheme AR-CPABE.

A. SYSTEM INITIALIZATION

For a given security parameter λ , AA executes the algorithm $\text{BGGen}(\lambda)$ to generate a bilinear group $(p, g, \mathbb{G}, \mathbb{G}_T, \hat{e})$. Then AA sets the attribute space $\mathcal{U} = \mathbb{Z}_p$ and executes the following Setup algorithm to generate the public parameter PP and the mask secret key MSK and a delegation key SK_d .

- $\text{Setup}(\lambda) \rightarrow (PP, MSK, SK_d)$: It selects three integers $\alpha_0, \alpha_1, a \xleftarrow{R} \mathbb{Z}_p$. The public parameter and the master key are $PP = (g, g^a, \hat{e}(g, g)^{\alpha_0 + \alpha_1})$ and $MSK = (\alpha_0, \alpha_1, g^{\alpha_0 + \alpha_1})$, respectively. Denote $\alpha = \alpha_0 + \alpha_1$ and $\hat{Y} = \hat{e}(g, g)$, then $PP = (g, g^a, \hat{Y}^\alpha)$ and $MSK = (\alpha_0, \alpha_1, g^\alpha)$. Finally, AA sets $SK_d = g^{\alpha_1}$ and sends SK_d to the cloud service provider CSP, via secure channels.

B. PRIVATE KEY GENERATION

Assume that U is a system user (DO or DU) with an attribute set $S \subset \mathbb{Z}_p$. AA generates a secret key SK associated with U by the following algorithm.

- $\text{KeyGen}(PP, MSK, S) \rightarrow SK$: Taken as input the public parameter $PP = (g, g^a, \hat{Y}^\alpha)$, the master key $MSK = (\alpha_0, \alpha_1, g^\alpha)$, and the attribute set S , AA selects an element $c \xleftarrow{R} \mathbb{Z}_p$ and generates the decryption key as

$$SK_u = (S, K, L, \bar{L}, \{K_x\}_{x \in S}), \quad (1)$$

where $K = g^{\alpha_0} \cdot (g^a)^c = g^{\alpha_0 + ac}$, $L = g^c$, $\bar{L} = (g^a)^c = g^{ac}$, $K_x = \mathcal{H}(x)^c$, and $\mathcal{H} : \{0, 1\}^* \rightarrow \mathbb{G}$ is a hash function. Finally, AA sends SK_u to the user U via secure channels.

C. FILE ENCRYPTION

As is mentioned in our system architecture (refer to Sec. IV-A), our AC-FEH scheme employs hybrid encryption for the outsourced EHRs. Especially, DO adopts an efficient symmetric encryption algorithm (e.g. AES) and a symmetric

key $\hat{K} \xleftarrow{R} \mathbb{G}_T$ to encrypt his/her EHR. Then the symmetric key \hat{K} is encrypted with the following algorithm **Encrypt** under an access policy $\mathbb{A} = (A, \rho)$ specified by DO. It is pointed out that **Encrypt** includes two sub-algorithms **DO.Encrypt** and **FN.Encrypt**, which are executed by DO and FN in turn. We denote by $\langle \text{EHR} \rangle_{\hat{K}}$ and CT_0 the two resulting ciphertexts of EHR and \hat{K} , respectively. Finally, FN uploads both the ciphertexts $\langle \text{EHR} \rangle_{\hat{K}}$ and CT_0 to CSP.

• **Encrypt**($PP, (A, \rho), \hat{K}$) $\rightarrow CT_0$: Taken as input the public parameter $PP = (g, g^a, \hat{Y}^\alpha)$, the access policy (A, ρ) with an ℓ by n matrix and a mapping $\rho : [\ell] \rightarrow \mathcal{U} \subset \mathbb{Z}_p$, and a message $\hat{K} \in \mathbb{G}_T$, the algorithm calls the following two sub-algorithms **DO.Encrypt** and **FN.Encrypt** to generate the ciphertext CT_0 .

- 1) **DO.Encrypt**(PP, \hat{K}) $\rightarrow TCT$: Taken as input the public parameter $PP = (g, g^a, \hat{Y}^\alpha)$ and a message $\hat{K} \in \mathbb{G}_T$, DO selects an integer $s \xleftarrow{R} \mathbb{Z}_p$ and computes $\hat{C} = \hat{K} \cdot (\hat{Y}^\alpha)^s = \hat{K} \cdot \hat{Y}^{\alpha s}$, $C_1 = g^{s_0}$. Then DO outputs the temple ciphertext of \hat{K} as

$$TCT = ((A, \rho), \hat{C}, C_1). \quad (2)$$

and sends TCT and $\langle \text{EHR} \rangle_{\hat{K}}$ to some fog node FN.

- 2) **FN.Encrypt**(PP, TCT) $\rightarrow CT_0$: Taken as input the public parameter $PP = (g, g^a, \hat{Y}^\alpha)$ and the temple ciphertext $TCT = ((A, \rho), \hat{C}, C_1)$, the fog node FN randomly selects a vector $\vec{v} = (s_0, v_2, \dots, v_n) \in \mathbb{Z}_p^n$ and computes the ℓ shares $\lambda_i = A_i \cdot \vec{v}^T, i \in [\ell]$. Then it adopts ℓ integers $\tau_i \xleftarrow{R} \mathbb{Z}_p, i \in [\ell]$, and computes $C_2 = C_1 \cdot g^{s_0} = g^{s+s_0}$, $C_{i,1} = (g^a)^{\lambda_i} \mathcal{H}(\rho(i))^{\tau_i} = g^{a\lambda_i} \mathcal{H}(\rho(i))^{\tau_i}, i \in [\ell]$, $C_{i,2} = g^{\tau_i}, i \in [\ell]$. Finally, the ciphertext is output as follows:

$$CT_0 = ((A, \rho), \hat{C}, C_1, C_2, \{C_{i,1}, C_{i,2}\}_{i \in [\ell]}). \quad (3)$$

and CT_0 and $\langle \text{EHR} \rangle_{\hat{K}}$ is uploaded to CSP. In the following, CT_0 is called as the original ciphertext to distinguish it from the following re-encrypted ciphertext CT_1 .

D. RE-ENCRYPTION

After receiving the ciphertexts CT_0 and $\langle \text{EHR} \rangle_{\hat{K}}$, CSP uses the delegation key SK_d to update the ciphertext CT_0 with the following **ReEncrypt** algorithm.

CSP skips this phase if there are no attributes needed to be revoked. Otherwise, CSP needs to executes the following **ReEncrypt** algorithm to re-encrypt the ciphertext CT_0 .

• **ReEncrypt**(PP, CT_0, y, SK_d) $\rightarrow CT_1$: Taken as input the public parameter $PP = (g, g^a, \hat{Y}^\alpha)$, a ciphertext $CT_0 = ((A, \rho), \hat{C}, C_1, C_2, \{C_{i,1}, C_{i,2}\}_{i \in [\ell]})$, an attribute y to be revoked, and the delegation key $SK_d = g^{\alpha_1}$, CSP chooses a random integer $v \xleftarrow{R} \mathbb{Z}_p$ and computes $D_1 = C_1^{1/v} = g^{s/v}, D_2 = (SK_d)^v = g^{\alpha_1 v}$. Then CSP generates the key components $\bar{C}_{i,1}$ and $\bar{C}_{i,2}, i \in [\ell]$, based on the following two different cases:

Case 1: y is a true attribute of some user U_r to be revoked.

In this case, CSP chooses a random key $\delta_y \xleftarrow{R} \mathbb{Z}_p$ and encrypts it to generate a ciphertext \tilde{C} with an access policy such that all authorized users (except U_r) with attribute y have the ability to decrypt \tilde{C} (please refer to the encryption and decryption details in [44]). Then CSP chooses an integer $u \xleftarrow{R} \mathbb{Z}_p$ and computes

$$\begin{cases} \bar{C}_{i,1} = C_{i,1} \cdot \mathcal{H}(\rho(i))^u = g^{a\lambda_i} \mathcal{H}(\rho(i))^{\tau_i+u}, i \in [\ell], \\ \bar{C}_{i,2} = C_{i,2} \cdot g^u = g^{\tau_i+u}, i \in [\ell] \text{ and } \rho(i) \neq y, \\ \bar{C}_{i,2} = (C_{i,2} \cdot g^u)^{1/\delta_y} = (g^{\tau_i+u})^{1/\delta_y}, i \in [\ell] \text{ and } \rho(i) = y. \end{cases}$$

Case 2: y is a fictional attribute. In other words, there is no attribute needed to be revoked. In this case, CSP selects an integer $u \xleftarrow{R} \mathbb{Z}_p$ and sets $\bar{C}_{i,1} = C_{i,1} \cdot \mathcal{H}(\rho(i))^u = g^{a\lambda_i} \mathcal{H}(\rho(i))^{\tau_i+u}$ and $\bar{C}_{i,2} = C_{i,2} \cdot g^u = g^{\tau_i+u}, \forall i \in [\ell]$.

Finally, CSP outputs the re-encrypted ciphertext:

$$CT_1 = ((A, \rho), \hat{C}, C_1, C_2, D_1, D_2, \{\bar{C}_{i,1}, \bar{C}_{i,2}\}_{i \in [\ell]}). \quad (4)$$

E. FILE DECRYPTION

Assuming that DU is a data user who has the decryption key $SK_u = (S, K, L, \bar{L}, \{K_x\}_{x \in S})$ and wishes to access the health data EHR of some patients. If DU owns an attribute $y \in S$ which has been revoked before, he/she first decrypts the corresponding \tilde{C} to obtain the random value δ_y and updates the key component K_y by computing $K_y \leftarrow (K_y)^{\delta_y} = \mathcal{H}(y)^{\delta_y}$. Then DU informs FN to download the ciphertexts $CT_1, \langle \text{EHR} \rangle_{\hat{K}}$ and runs the following algorithm **Decrypt** together with FN to get the key \hat{K} from CT_1 . Moreover, **Decrypt** consists of two sub-algorithms **FN.Decrypt** and **DU.Decrypt** executed by FN and DU, respectively. In the end, DU decrypts the ciphertext $\langle \text{EHR} \rangle_{\hat{K}}$ with \hat{K} to get the health data EHR.

• **Decrypt**(CT_1, SK_u) $\rightarrow \hat{K}$: Taken as input the ciphertext $CT_1 = ((A, \rho), \hat{C}, C_1, C_2, D_1, D_2, \{\bar{C}_{i,1}, \bar{C}_{i,2}\}_{i \in [\ell]})$ and the decryption key $SK_u = (S, K, L, \bar{L}, \{K_x\}_{x \in S})$, it calls the following two sub-algorithms **FN.Decrypt** and **DU.Decrypt** to recover \hat{K} .

- 1) **FN.Decrypt**(CT_1, SK') $\rightarrow \hat{F}$: Taken as input the ciphertext $CT_1 = ((A, \rho), \hat{C}, C_1, C_2, D_1, D_2, \{\bar{C}_{i,1}, \bar{C}_{i,2}\}_{i \in [\ell]})$ and the partial key $SK' = (S, L, \bar{L}, \{K_x\}_{x \in S})$ of SK_u , FN first checks whether the attribute set S in SK' satisfies the access policy (A, ρ) in CT_1 . If not, it outputs an error symbol \perp . Otherwise, it can find an index set $I = \{i \in [\ell] \mid \rho(i) \in S\}$ and $|I|$ constants $\omega_i, i \in I$, which satisfy the equation $\sum_{i \in I} \omega_i A_i = (1, 0, \dots, 0)$. Then the following formula is calculated:

$$\hat{F} = \frac{\hat{C} \cdot \prod_{i \in I} \hat{e}(\bar{C}_{i,2}, K_{\rho(i)})^{\omega_i} \cdot \hat{e}(C_2, \bar{L})}{\hat{e}\left(\prod_{i \in I} \bar{C}_{i,1}^{\omega_i}, L\right) \cdot \hat{e}(D_1, D_2)}. \quad (5)$$

Finally, FN sends \hat{F} and C_1 to DU.

After receiving \hat{F} and C_1 from FN, DU runs the following DU.Decrypt algorithm to obtain the symmetric key \hat{K} .

- 2) DU.Decrypt(\hat{F}, C_1, K) $\rightarrow \hat{K}$: Taken as input the elements $\hat{F} \in \mathbb{G}_T$ and $C_1, K \in \mathbb{G}$, the algorithm computes:

$$\hat{K} = \frac{\hat{F}}{\hat{e}(K, C_1)}. \quad (6)$$

F. CORRECTNESS

Only the correctness of decryption in the case of attribute revocation is verified. (i.e. the **Case 1** in the re-encryption phase). Similarly, another case can be verified. For simplicity, the thesis assumes the index $j \in I$ satisfies $\rho(j) = y$ and denotes $h_i = \mathcal{H}(\rho(i))$, $i \in I$. Then we have

$$\begin{aligned} \hat{F} &= \frac{\hat{C} \cdot \prod_{i \in I} \hat{e}(\bar{C}_{i,2}, K_{\rho(i)})^{\omega_i} \cdot \hat{e}(C_2, \bar{L})}{\hat{e}\left(\prod_{i \in I} \bar{C}_{i,1}^{\omega_i}, L\right) \cdot \hat{e}(D_1, D_2)} \\ &= \frac{\hat{C} \cdot \hat{e}(\bar{C}_{j,2}, K_{\rho(j)})^{\omega_j} \cdot \prod_{j \neq i \in I} \hat{e}(\bar{C}_{i,2}, K_{\rho(i)})^{\omega_i} \cdot \hat{e}(C_2, \bar{L})}{\hat{e}\left(\prod_{i \in I} \bar{C}_{i,1}^{\omega_i}, L\right) \cdot \hat{e}(D_1, D_2)} \\ &= \frac{\hat{C} \cdot \hat{e}\left(g^{\frac{\tau_j+u}{\delta_y}}, h_j^{c\delta_y}\right)^{\omega_j} \cdot \prod_{j \neq i \in I} \hat{e}(g^{\tau_i+u}, h_i^c)^{\omega_i} \cdot \hat{e}(C_2, \bar{L})}{\hat{e}\left(\prod_{i \in I} \bar{C}_{i,1}^{\omega_i}, L\right) \cdot \hat{e}(D_1, D_2)} \\ &= \frac{\hat{C} \cdot \hat{e}(g^{\tau_j+u}, h_j^c)^{\omega_j} \cdot \prod_{j \neq i \in I} \hat{e}(g^{\tau_i+u}, h_i^c)^{\omega_i} \cdot \hat{e}(C_2, \bar{L})}{\hat{e}\left(\prod_{i \in I} \bar{C}_{i,1}^{\omega_i}, L\right) \cdot \hat{e}(D_1, D_2)} \\ &= \frac{\hat{K} \cdot \hat{Y}^{\alpha s} \cdot \prod_{i \in I} \hat{e}(g^{\tau_i+u}, h_i^c)^{\omega_i} \cdot \hat{e}(g^{s+s_0}, g^{ac})}{\hat{e}\left(\prod_{i \in I} (g^{a\lambda_i} h_i^{\tau_i+u})^{\omega_i}, g^c\right) \cdot \hat{e}(g^{s/v}, g^{\alpha_1 v})} \\ &= \frac{\hat{K} \cdot \hat{Y}^{\alpha s} \cdot \prod_{i \in I} \hat{e}(g^{\tau_i+u}, h_i^c)^{\omega_i} \cdot \hat{Y}^{ac(s+s_0)}}{\hat{e}\left(\prod_{i \in I} g^{a\lambda_i \omega_i}, g^c\right) \cdot \hat{e}\left(\prod_{i \in I} h_i^{(\tau_i+u)\omega_i}, g^c\right) \cdot \hat{Y}^{\alpha_1 s}} \\ &= \frac{\hat{K} \cdot \hat{Y}^{\alpha_0 s} \cdot \hat{Y}^{ac(s+s_0)}}{\hat{e}\left(g^{a \sum_{i \in I} \lambda_i \omega_i}, g^c\right)} \\ &= \frac{\hat{K} \cdot \hat{Y}^{\alpha_0 s} \cdot \hat{Y}^{ac(s+s_0)}}{\hat{e}\left(g^{a \sum_{i \in I} \omega_i A_i \cdot \bar{v}^T}, g^c\right)} \\ &= \hat{K} \cdot \hat{Y}^{\alpha_0 s + acs}. \end{aligned}$$

Thus, we can get $\frac{\hat{F}}{\hat{e}(K, C_1)} = \frac{\hat{K} \cdot \hat{Y}^{\alpha_0 s + acs}}{\hat{e}(g^{\alpha_0 + ac}, g^s)} = \hat{K}$.

VI. SECURITY ANALYSIS

The following theorem shows the selective security of our AR-CPABE scheme.

Theorem 1: Under the q -parallel BDHE assumption, our AR-CPABE is selectively secure in the standard model.

Proof: Let $\mathcal{E} = (\text{Setup}, \text{KeyGen}, \text{Encrypt}, \text{ReEncrypt}, \text{Decrypt})$ be our AR-CPABE scheme and \mathcal{A} a PPT adversary on \mathcal{E} . We show that if there exists a PPT adversary \mathcal{A} who has a non-negligible advantage to win the game $\text{Game}_{\mathcal{E}, \mathcal{A}}^{\text{SS}}$, then the involved challenger \mathcal{C} can break the underlying q -parallel BDHE assumption.

- 1) **Initialization.** For a given security parameter λ , the challenger \mathcal{C} runs $\text{BGGGen}(\lambda)$ to generate a bilinear group $(p, g, \mathbb{G}, \mathbb{G}_T, \hat{e})$. The adversary \mathcal{A} selects an LSSS access structure (A^*, ρ^*) with a matrix A^* of size $\ell^* \times n^*$ ($\ell^*, n^* \leq q$) and an attribute revocation list RL_y of attribute y . \mathcal{A} sends them both to \mathcal{C} .

- 2) **Setup.** \mathcal{C} chooses $\alpha', \alpha'' \xleftarrow{R} \mathbb{Z}_p$ randomly, and sets $\alpha_0 = \alpha' + a^{q+1}$, $\alpha_1 = \alpha''$, $\alpha = \alpha' + a^{q+1} + \alpha''$. It is obvious that we have $\hat{e}(g, g)^\alpha = \hat{e}(g^a, g^{a^q}) \cdot \hat{e}(g, g)^{\alpha'}$.

While considering the call to $\mathcal{H}(x)$, if $\mathcal{H}(x)$ is already defined in the table, the returned answer will be exactly the same as before.

Otherwise let $X = \{i : \rho^*(i) = x\}$ for each $x \in \mathcal{U}$, where i is the index of the row in A^* and $\varsigma = \mathcal{H}(x)$. Then a random value d_x is selected and \mathcal{C} programs the oracle as $\varsigma = g^{d_x} \prod_{i \in X} g^{a^{A_{i,1}^*}/b_i} \cdot g^{a^2 A_{i,2}^*/b_i} \dots g^{a^n A_{i,n}^*/b_i}$. If $X = \emptyset$, then $\varsigma = g^{d_x}$, the responses from the oracle are distributed randomly due to the value of g^{d_x} .

- 3) **Phase 1.** At this stage, \mathcal{C} responds to each key query. It is supposed that the adversary \mathcal{A} has a key query associated with the attribute set S for (ID_i, S_i) that cannot satisfy the access structure (A^*, ρ^*) , and let the set $I_S = \{i | \rho^*(i) \in S\}$. If $ID_i \in RL_y$, $S'_i = S_i \setminus \{y\}$. Otherwise, $S'_i = S_i$.

Then, \mathcal{C} chooses a random value $\hat{r} \xleftarrow{R} \mathbb{Z}_p$ and finds a vector $\vec{w} = (w_1, \dots, w_{n^*}) \in \mathbb{Z}_p^{n^*}$. In this way, $w_1 = -1$ and for all $i \in I_S$ we have $\vec{w} \cdot A_i^* = 0$. According to property of LSSS in Section 3, it could be obtained that there must be such a vector w that $w \cdot A_i = 0$. Next, \mathcal{C} sets $c = \hat{r} + w_1 a^q + w_2 a^{q-1} + \dots + w_{q^*} a$ and has $L = g^{\hat{r}} \prod_{i=1}^{n^*} (g^{a^{q+1-i}})^{w_i} = g^c$.

Then, we can get $\bar{L} = L^a = g^{ac}$. It should be observed that when $w_1 = -1$, g^{ac} contains a term of $g^{-a^{q+1}}$, which can be eliminated by the unknown term in g^{α_0} . Set $K = g^{\alpha'} g^{ac} \prod_{i=2}^{n^*} (g^{a^{q+2-i}})^{w_i}$. Then a key K_x is calculated, which is related to attribute value x . If there is no i such that $\rho(i)^* = x$ for all attributes $x \in S$, let $K_x = L^{d_x}$. In order to create the key components K_x for the attribute $x \in S$, where x is used in the access structure. \mathcal{C} is unable to simulate all the terms of the form g^{a^{q+1}/b_i} . Nevertheless, we have that $\vec{w} \cdot A_i^* = 0$; therefore, all these terms are eliminated. \mathcal{C} sets $K_x = L^{d_x} \prod_{i \in X} \prod_{j=1}^{n^*} \left(g^{\hat{r}(d^j/b_i)}\right)^{A_{i,j}^*}$.

$$\prod_{i \in X} \prod_{j=1}^{n^*} \prod_{\gamma=1, \gamma \neq j}^{n^*} \left(g^{a^{q+1+j-\gamma}/b_i}\right)^{w_{\gamma} A_{i,j}^*}.$$

TABLE 2. Function comparison with other related works.

Schemes	[31]	[34]	[38]	[39]	[40]	[45]	Ours
Large universe	√	√	×	×	×	×	√
LSSS	×	√	√	×	×	√	√
Outsourcing capability	×	×	√	×	√	×	√
Attribute revocation	√	√	×	×	×	×	√
Scalability	√	√	√	×	√	×	√

4) **Challenge.** \mathcal{A} sends \hat{M}_0 and \hat{M}_1 to \mathcal{C} . The latter flips a coin $b \xleftarrow{R} \{0, 1\}$. Then \mathcal{C} gets a challenge re-encryption ciphertext $CT_1 = (\hat{C}, C_1, C_2, D_1, D_2, \{\tilde{C}_{i,1}, \tilde{C}_{i,2}\}_{i \in [\ell]})$ and sends CT_1 to \mathcal{A} .

The tricky part is to simulate the values of $\tilde{C}_i, i \in [\ell]$. D_2 is the ciphertext of the delegated key. $\tilde{C}_{i,1}$ contains the terms that \mathcal{C} is unable to change. So, to cancel out these, \mathcal{C} needs to choose values y'_2, y'_3, \dots, y'_n and $\tau'_1, \dots, \tau'_\ell \in \mathbb{Z}_p$ randomly and build a secret sharing vector $\tilde{v} = (s, sa+y'_2, sa^2+y'_3, \dots, sa^{n-1}+y'_n) \in \mathbb{Z}_p^n$. It is assumed that the other rows of A^* have the same attributes as row i . We get the set $I_S^* = \{\gamma : \rho^*(\gamma) = \rho^*(i) \text{ and } \gamma \neq i\}$ for each row A_i^* of A^* . Set $\tau_i + u = -\tau_i - sb_i$ for the non-revoked attributes, which means one of the challenge ciphertexts will be generated as $\tilde{C}_{i,2} = g^{-\tau_i} \cdot g^{-sb_i}$. Nevertheless, if the attributes are revoked: $\rho^*(j) = y$ and $j \neq i$, another challenge ciphertext is computed as $\tilde{C}_{i,2} = (g^{-\tau_i} \cdot g^{-sb_i})^{\delta_y}$ by selecting a random key δ_y . It is noted that \tilde{C} is defined as the ciphertext of δ_y under the access structure (A^*, ρ^*) . For all attributes, we have $\tilde{C}_{i,1} = \mathcal{H}(\rho^*(i))^{-\tau'_i} \cdot (g^{b_i})^{-d_{\rho^*(i)}} \cdot \left(\prod_{j=2}^{n^*} (g^{a^{i,j} \cdot y'_j})\right) \cdot \left(\prod_{\gamma \in I_S^*} \prod_{j=1}^{n^*} (g^{d_{sb_i/b_\gamma}})\right)^{-A_{\gamma,j}^*}$.

5) **Phase 2.** Phase 2 is the same as phase 1.
 6) **Guess.** \mathcal{A} outputs a guess b' of b . If $b = b', \mathcal{C}$ outputs 0 to guess the value $\hat{e}(g, g)^{a^{q+1}s}$. Otherwise, \mathcal{C} outputs 1. We can get $\Pr[\mathcal{A}(\tilde{y}, \hat{e}(g, g)^{sa^{q+1}}) = 0] = \text{Adv}_{\mathcal{A}}^q(\lambda) + \frac{1}{2}$. M_b is hidden from \mathcal{A} when a random group element R is given. Then we have $\Pr[\mathcal{A}(\tilde{y}, R \xleftarrow{R} \mathbb{G}_T) = 0] = \frac{1}{2}$. Therefore, the q -parallel BDHE assumption can be broken by \mathcal{C} with a non-negligible advantage $\text{Adv}_{\mathcal{A}}^q(\lambda) = \Pr[\mathcal{A}(\tilde{y}, \hat{e}(g, g)^{sa^{q+1}}) = 0] - \Pr[\mathcal{A}(\tilde{y}, R \xleftarrow{R} \mathbb{G}_T) = 0]$.
 The proof is finished.

Forward and backward securities are the crucial properties for a practical CP-ABE scheme. The former means that any user whose access right has been revoked will no longer be able to access the subsequent data, while the latter ensures that any user who has an access right cannot access the data exchanged before he owns the right. As for the two security properties, we have the following theorem.

Theorem 2: Our AP-CPABE scheme satisfies the forward and backward securities.

Proof: Recall that in our AR-CPABE scheme, any user is unable to extract a random key from ciphertext \tilde{C} for

the update of the key SK_u when some of his attributes have been revoked. Therefore, the user can not decrypt any components related to the revoked attributes in ciphertext and loses the access right to the plaintext data any more. The means that our AR-CPABE scheme has the forward security.

On the other hand, let us assume that there is a user U who comes to hold an attribute set satisfying the access policy in the ciphertexts generated after a certain period. The user U is unable to decrypt a previous ciphertext because the component $C_{i,2} = (g^{\tau_i+u})^{1/\delta_y}$ in the previous ciphertext is re-encrypted with previous random numbers δ_y and u and U doesn't know these two numbers. This shows the backward security of our AR-CPABE scheme.

VII. PERFORMANCE ANALYSIS

We give a detailed analysis of our AR-CPABE scheme in view of the functions, storage costs, and computation costs.

A. FUNCTIONS

TABLE 2 presents the comparison of the functions between our work and other related works. It is clear that only our AR-CPABE scheme has simultaneously the attribute revocation and outsourcing functions. The schemes [38], [45] solved the problem of user revocation, but not the attribute revocation. Nonetheless, compared with the ones [34], [38], [45], their schemes are based on the access tree structure rather than LSSS.

For the aspect of attribute universes, the schemes [38]–[40], [45] are based on the small universe, which means that the attributes in the setup phase are fixed and no more attributes are allowed to be added afterwards. This feature is not suitable for fog computing environment since it likely needs to change attribute set and access structure operation of the system. Our AR-CPABE scheme is constructed on large universe and more efficient than [38] and [40]. The scalability means that the increase of revoked users should not result in the increase of storage burden of cloud servers. In the scheme [45], the user's unique identity is associated with the private key and is embedded in the ciphertext for revocation. Thus the size of the ciphertext increases linearly with the number of revoked users, which increases the storage burden. The scheme [39] has the same case as the scheme [45]. Compared to other schemes, our AR-CPABE scheme is more practical because the size of the private key and ciphertext is independent of the number of revoked users.

TABLE 3. Storage cost comparison with other related works.

Schemes	PP		MSK			SK _u		CT ₀		
	G	G _T	G	G _T	Z _p	G	Z _p	G	G _T	Z _p
[31]	1	1	1	0	1	2 S + 1	0	3ℓ + 1	1	0
[34]	2	1	0	0	4	S + 4	1	3ℓ + 1	1	0
[38]	5	1	5	1	1	4 S + 4	4 S + 4	6ℓ + 4	0	2ℓ
[39]	U + 2	1	1	0	U + 1	2 S + 1	0	3ℓ + 2r + 1	1	0
[40]	U + 3	1	U + 1	0	1	S + 3	0	ℓ + 3	1	0
[45]	n U + 4	1	1	0	0	n + S + 3	0	nℓ + 2r + 1	1	0
Our scheme	2	1	1	0	2	S + 3	0	2ℓ + 2	1	0

TABLE 4. The comparison of encryption cost (EncC) and decryption cost (DecC) with other related works.

Schemes		[39]			[40]			Our scheme		
		G	G _T	ê	G	G _T	ê	G	G _T	ê
EncC	FN.Encrypt	–	–	–	ℓ + 2	0	0	2ℓ + 1	0	0
	Do.Encrypt	3ℓ + 3r + 1	2	0	4	2	0	1	1	0
	All Cost	3ℓ + 3r + 1	2	0	ℓ + 6	2	0	2ℓ + 2	1	0
DecC	FN.Decrypt	–	–	–	0	2ω + 2	I + 2	0	1	I + 3
	DU.Decrypt	0	4 I + 3r + 1	3 I + 2r + 1	0	2	1	0	1	1
	AllCost	0	4 I + 3r + 1	3 I + 2r + 1	0	2ω + 4	I + 3	0	2	I + 4

† I refers to the set involved in the decryption phase; ‡ ω denotes the number of the nodes satisfying an access structure.

TABLE 5. The comparison of revocation cost (RevC) with other related works.

Schemes		[31]			[34]			Our scheme		
		G	G _T	Z _p	G	G _T	Z _p	G	G _T	Z _p
RevC	ReEncryption	5ℓ _y + μ + 3	1	0	–	–	–	5ℓ _y + 3	0	0
	CiphertextUpdate	–	–	–	6ℓ _y + 3	0	0	–	–	–
	KeyUpdate	1	0	0	6	0	4	1	0	0
	All Cost	5ℓ _y + μ + 4	1	0	6ℓ _y + 9	0	4	5ℓ _y + 4	0	0

† μ refers to the number of users; ‡ ℓ_y denotes the number of the revoked attributes.

B. STORAGE COSTS

Table 3 gives the storage costs by comparing the system public parameters PP, the master key MSK, the decryption key SK_u, and the ciphertext CT₀ with other related works [31], [34], [38]–[40], [45]. In Table 3, G and G_T are two cyclic groups of prime order, n is the column size in the access structure, r refers to the number of revoked users, and ℓ, U, S denote the number of the attributes appeared in ciphertext, the attribute universe, and the attribute set, respectively. From Table 3, we find that the size of PP in our system and the schemes [31], [34], [38] is a constant, while the size in the three schemes [39], [40], [45] increases linearly with the size of the attribute universe. Compared with schemes [39], [40], the size of MSK in our scheme is also constant. Further, we can find from Table 3 that SK_u in our scheme has a smaller size compared with the schemes [31], [34], [38], [39], [45]. As for the size of CT₀, our scheme has a smaller value than other schemes based on LSSS [34], [38], [45].

C. COMPUTATION COSTS

Table 4 presents the comparative summary of the computation costs between our scheme and those of [39], [40]. The metrics are the runtime associated with the exponential operations on groups G, G_T and the bilinear pairing operation ê in the encryption and decryption algorithms. Fig. 4 shows the average execution time of our scheme and the schemes of

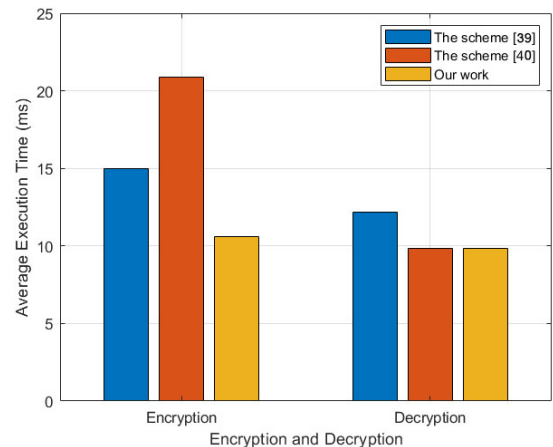


FIGURE 4. Encryption and Decryption time.

[39], [40]. It is based on a linux machine using an Intel Core i7-10750 cpu and an BN curve of 128 bits security level. Although the computation overhead of FN.Encryption in our scheme is higher than that of the scheme [40], the computation overhead of DO.Encryption in our scheme is smaller. This feature is more effective for the ABE systems with the outsourcing function. From the perspective of decryption, the overhead of FN.Decrypt in [40] is greater than our scheme because the decryption cost of recovering secrets is related

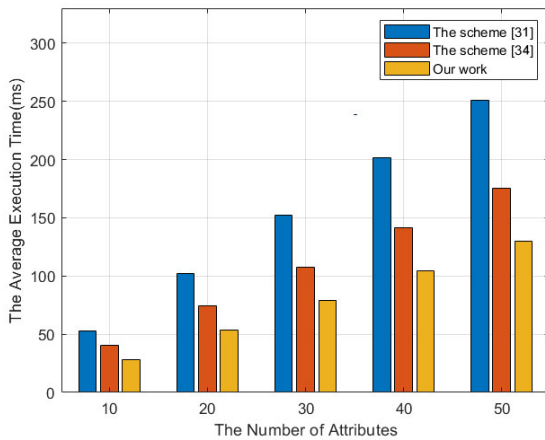


FIGURE 5. Revocation time.

to the node number of the access tree. Finally, we mention that our scheme is more efficient than the schemes [39], [40] since the total computation overheads of our scheme are much lower.

Finally we evaluate the revocation efficiency of our scheme by comparing it with the two schemes [31], [34] since all the three schemes support the attribute revocation function (please refer to Table 2). Table 5 gives a detailed comparison from the aspects of ReEncryption, CiphertextUpdate, KeyUpdate, and all cost. It shows that our scheme has the lowest computation overheads from the perspective of all cost. Fig. 5 also shows the advantage of our scheme over the schemes [31], [34] in terms of the average execution time.

VIII. CONCLUSION

Motivated by the lack of expressible and practical CP-ABE schemes that are capable of supporting key functionalities (e.g., outsourcing calculations, traceability and revocation), we proposed a new CP-ABE scheme. The latter leverages the LSSS access structure and has strong expressiveness. Furthermore, our new scheme supports the large universe and attribute revocation properties. In order to improve efficiency, our scheme is deployed in the fog computing environment to reduce costs associated with encryption and decryption. The result is an efficient access control scheme with outsourcing and attribute revocation for fog-enabled E-health, which can be deployed on resource constrained devices.

No cryptographic scheme is perfect. The limitation of our proposed access control scheme is that it does not support traceability. Thus, this is one potential future research extension – “How do we incorporate traceability in our scheme?”. Also, can we combine this scheme with another scheme with black-box mandatory traceability to simultaneously realize traceable, revocable, and outsourced calculations?

REFERENCES

- [1] T.-Y. Wu, Z. Lee, M. S. Obaidat, S. Kumari, S. Kumar, and C.-M. Chen, “An authenticated key exchange protocol for multi-server architecture in 5G networks,” *IEEE Access*, vol. 8, pp. 28096–28108, 2020.
- [2] S. Kamara and K. Lauter, “Cryptographic cloud storage,” in *Financial Cryptography and Data Security*, vol. 6054. Berlin, Germany: Springer, 2010, pp. 136–149.
- [3] H. Cui, R. H. Deng, B. Qin, and J. Weng, “Key regeneration-free ciphertext-policy attribute-based encryption and its application,” *Inf. Sci.*, vol. 517, pp. 217–229, May 2020.
- [4] Y. Cui, Q. Huang, J. Huang, H. Li, and G. Yang, “Ciphertext-policy attribute-based encrypted data equality test and classification,” *Comput. J.*, vol. 62, no. 8, pp. 1166–1177, Aug. 2019.
- [5] S. Gao, G. Piao, J. Zhu, X. Ma, and J. Ma, “TrustAccess: A trustworthy secure ciphertext-policy and attribute hiding access control scheme based on blockchain,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 5784–5798, Jun. 2020.
- [6] J. Hao, C. Huang, J. Ni, H. Rong, M. Xian, and X. Shen, “Fine-grained data access control with attribute-hiding policy for cloud-based IoT,” *Comput. Netw.*, vol. 153, pp. 1–10, Apr. 2019.
- [7] J. Li, S. Wang, Y. Li, H. Wang, H. Wang, H. Wang, J. Chen, and Z. You, “An efficient attribute-based encryption scheme with policy update and file update in cloud computing,” *IEEE Trans. Ind. Informat.*, vol. 15, no. 12, pp. 6500–6509, Dec. 2019.
- [8] S. Xu, J. Yuan, G. Xu, Y. Li, X. Liu, Y. Zhang, and Z. Ying, “An efficient attribute-based hierarchical data access control scheme in cloud computing,” *Inf. Sci.*, vol. 538, pp. 19–38, Oct. 2020.
- [9] Al-Dahhan, Shi, Lee, and Kifayat, “Survey on revocation in ciphertext-policy attribute-based encryption,” *Sensors*, vol. 19, no. 7, p. 1695, Apr. 2019.
- [10] C.-W. Liu, W.-F. Hsien, C.-C. Yang, and M.-S. Hwang, “A survey of attribute-based access control with user revocation in cloud data storage,” *Int. J. Netw. Secur.*, vol. 18, no. 5, pp. 900–916, Sep. 2016.
- [11] V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-based encryption for fine-grained access control of encrypted data,” in *Proc. Comput. Commun. Secur.*, 2006, pp. 89–98.
- [12] J. Bethencourt, A. Sahai, and B. Waters, “Ciphertext-policy attribute-based encryption,” *Proc. IEEE Symp. Secur. Privacy*, Dec. 2007, pp. 321–334.
- [13] B. Waters, “Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization,” in *Proc. Public Key Cryptogr.*, Dec. 2011, pp. 53–70.
- [14] S. Yu, C. Wang, K. Ren, and W. Lou, “Achieving secure, scalable, and fine-grained data access control in cloud computing,” in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 534–542.
- [15] C. Guo, R. Zhuang, Y. Jie, Y. Ren, T. Wu, and K.-K.-R. Choo, “Fine-grained database field search using attribute-based encryption for E-healthcare clouds,” *J. Med. Syst.*, vol. 40, no. 11, pp. 1–8, Nov. 2016.
- [16] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, “Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 1, pp. 131–143, Jan. 2013.
- [17] L. Cheung and C. Newport, “Provably secure ciphertext policy ABE,” in *Proc. Conf. Comput. Commun. Secur.*, 2007, pp. 456–465.
- [18] J. Li, Q. Huang, X. Chen, S. S. M. Chow, D. S. Wong, and D. Xie, “Multi-authority ciphertext-policy attribute-based encryption with accountability,” in *Proc. Conf. Comput. Commun. Secur.*, 2011, pp. 386–390.
- [19] F. Guo, Y. Mu, W. Susilo, D. S. Wong, and V. Varadarajan, “CP-ABE with constant-size keys for lightweight devices,” *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 5, pp. 763–771, May 2014.
- [20] Q. Li, J. Ma, R. Li, J. Xiong, and X. Liu, “Large universe decentralized key-policy attribute-based encryption,” *Secur. Commun. Netw.*, vol. 8, no. 3, pp. 501–509, Feb. 2015.
- [21] Q. Li, J. Ma, R. Li, J. Xiong, and X. Liu, “Provably secure unbounded multi-authority ciphertext-policy attribute-based encryption,” *Secur. Commun. Netw.*, vol. 8, no. 18, pp. 4098–4109, Dec. 2015.
- [22] Q. Malluhi, “A ciphertext-policy attribute-based encryption scheme with optimized ciphertext size and fast decryption,” in *Proc. Conf. Comput. Commun. Secur.*, 2017, pp. 230–240.
- [23] J. Ning, Z. Cao, X. Dong, K. Liang, L. Wei, and K.-K.-R. Choo, “CryptCloud+: Secure and expressive data access control for cloud storage,” *IEEE Trans. Services Comput.*, early access, Jan. 9, 2019, doi: 10.1109/TSC.2018.2791538.
- [24] C.-H. Liu, F.-Q. Lin, D.-L. Chiang, T.-L. Chen, C.-S. Chen, H.-Y. Lin, Y.-F. Chung, and T.-S. Chen, “Secure PHR access control scheme for healthcare application clouds,” in *Proc. 42nd Int. Conf. Parallel Process.*, Oct. 2013, pp. 1067–1076.
- [25] N. Attrapadung and H. Imai, “Conjunctive broadcast and attribute-based encryption,” in *Proc. Int. Conf. Pairing-Based Cryptogr.*, 2009, pp. 248–265.

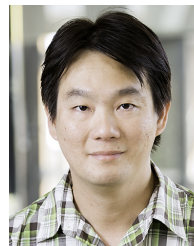
- [26] P. Junod and A. Karlov, "An efficient public-key attribute-based broadcast encryption scheme allowing arbitrary access policies," in *Proc. 10th Annu. ACM workshop Digit. Rights Manage.*, 2010, pp. 13–24.
- [27] Z. Xu and K. M. Martin, "Dynamic user revocation and key refreshing for attribute-based encryption in cloud storage," in *Proc. IEEE 11th Int. Conf. Trust, Secur. Privacy Comput. Commun.*, Jun. 2012, pp. 844–849.
- [28] X. Liang, R. Lu, X. Lin, and X. S. Shen, "Ciphertext policy attribute based encryption with efficient revocation," *Technical Report, Univ. Waterloo*, vol. 2, p. 8, Dec. 2010.
- [29] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based encryption with non-monotonic access structures," in *Proc. 14th ACM Conf. Comput. Commun. Secur. (CCS)*, 2007, pp. 195–203.
- [30] M. Pirretti, P. Traynor, P. McDaniel, and B. Waters, "Secure attribute-based systems," *J. Comput. Secur.*, vol. 18, no. 5, pp. 799–837, Aug. 2010.
- [31] J. Hur, "Improving security and efficiency in attribute-based data sharing," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 10, pp. 2271–2282, Oct. 2013.
- [32] J. Hur and D. K. Noh, "Attribute-based access control with efficient revocation in data outsourcing systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 7, pp. 1214–1221, Jul. 2011.
- [33] X. Xie, H. Ma, J. Li, and X. Chen, "New ciphertext-policy attribute-based access control with efficient revocation," in *Proc. Int. Conf. Inf. Commun. Technol.*, 2013, pp. 373–382.
- [34] K. Yang and X. Jia, *Security for Cloud Storage Systems*. Cham, Switzerland: Springer, 2014.
- [35] X. Fu, X. Nie, T. Wu, and F. Li, "Large universe attribute based access control with efficient decryption in cloud storage system," *J. Syst. Softw.*, vol. 135, pp. 157–164, Jan. 2018.
- [36] Y. Liu, Y. Zhang, J. Ling, and Z. Liu, "Secure and fine-grained access control on e-healthcare records in mobile cloud computing," *Future Gener. Comput. Syst.*, vol. 78, pp. 1020–1026, Jan. 2018.
- [37] J. Li, Y. Zhang, X. Chen, and Y. Xiang, "Secure attribute-based data sharing for resource-limited users in cloud computing," *Comput. Secur.*, vol. 72, pp. 1–12, Jan. 2018.
- [38] R. Zhang, H. Ma, and Y. Lu, "Fine-grained access control system based on fully outsourced attribute-based encryption," *J. Syst. Softw.*, vol. 125, pp. 344–353, Mar. 2017.
- [39] P. Zhang, Z. Chen, K. Liang, S. Wang, and T. Wang, "A cloud-based access control scheme with user revocation and attribute update," in *Information Security Privacy*. Cham, Switzerland: Springer, 2016, pp. 525–540.
- [40] P. Zhang, Z. Chen, J. K. Liu, K. Liang, and H. Liu, "An efficient access control scheme with outsourcing capability and attribute update for fog computing," *Future Gener. Comput. Syst.*, vol. 78, pp. 753–762, Jan. 2018.
- [41] L. Zu, Z. Liu, and J. Li, "New ciphertext-policy attribute-based encryption with efficient revocation," in *Proc. Conf. Comput. Inf. Technol.*, 2014, pp. 281–287.
- [42] Y. Shi, Q. Zheng, J. Liu, and Z. Han, "Directly revocable key-policy attribute-based encryption with verifiable ciphertext delegation," *Inf. Sci.*, vol. 295, pp. 221–231, Feb. 2015.
- [43] S.-S. Tu, S.-Z. Niu, and H. Li, "A fine-grained access control and revocation scheme on clouds," *Concurrency Comput., Pract. Exper.*, vol. 28, no. 6, pp. 1697–1714, Apr. 2016.
- [44] X. Liang, Z. Cao, H. Lin, and J. Shao, "Attribute based proxy re-encryption with delegating capabilities," in *Proc. Conf. Comput. Commun. Secur.*, 2009, pp. 276–286.
- [45] Y. Li, J. Zhu, X. Wang, Y. Chai, and S. Shao, "Optimized ciphertext-policy attribute-based encryption with efficient revocation," *Int. J. Secur. Appl.*, vol. 7, no. 6, pp. 385–394, Nov. 2013.



JING ZHAO received the bachelor's degree in computer science and technology from Henan University, in 2018. She is currently pursuing the master's degree in software engineering with East China Normal University, Shanghai, China. Her research interests include cryptography, network security, and attribute-based encryption.



PENG ZENG (Member, IEEE) received the Ph.D. degree in computer science and technology from Shanghai Jiao Tong University, Shanghai, China, in 2009. He is currently an Associate Professor with East China Normal University, Shanghai. His current research interests include applied cryptography, network information security, and coding theory.



KIM-KWANG RAYMOND CHOO (Senior Member, IEEE) received the Ph.D. degree in information security from the Queensland University of Technology, Australia, in 2006. He currently holds the Cloud Technology Endowed Professorship at The University of Texas at San Antonio (UTSA). He is an IEEE Computer Society Distinguished Visitor for the period 2021–2023 and included in the Web of Science's Highly Cited Researcher in the field of Cross-Field in 2020. He is named as the Cybersecurity Educator of the Year—APAC (Cybersecurity Excellence Awards are produced in cooperation with the Information Security Community on LinkedIn) in 2016. In 2015, he and his team won the Digital Forensics Research Challenge organized by Germany's University of Erlangen-Nuremberg. He was a recipient of the 2019 IEEE Technical Committee on Scalable Computing (TCSC) Award for Excellence in Scalable Computing (Middle Career Researcher), the 2018 UTSA College of Business Col. Jean Piccione and Lt. Col. Philip Piccione Endowed Research Award for Tenured Faculty, the Outstanding Associate Editor of 2018 for IEEE ACCESS, the British Computer Society's 2019 Wilkes Award Runner-Up, the 2014 Highly Commended Award by the Australia New Zealand Policing Advisory Agency, the Fulbright Scholarship in 2009, the 2008 Australia Day Achievement Medallion, and the British Computer Society's Wilkes Award in 2008. He has also received the Best Paper Awards from ESORICS 2015, IEEE TrustCom 2018, *EURASIP Journal on Wireless Communications and Networking* in 2019, and *IEEE Consumer Electronics Magazine* in 2020; the Korea Information Processing Society's JIPS Survey Paper Award (Gold) 2019; the IEEE Blockchain 2019 Outstanding Paper Award; and Best Student Paper Awards from Inscrypt 2019 and ACISP 2005. He serves as the Department Editor for IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT and an Associate Editor for IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING and IEEE TRANSACTIONS ON BIG DATA.

• • •