1-2022

Lightweight and expressive fine-grained access control for healthcare internet-of-things

Shengmin XU
Singapore Management University, smxu@smu.edu.sg

Yingjiu Li
Singapore Management University, yjli@smu.edu.sg

Robert H. DENG
Singapore Management University, robertdeng@smu.edu.sg

Yinghui ZHANG
Singapore Management University, yinghuizhang@smu.edu.sg

Xiangyang LUO
State Key Laboratory of Mathematical Engineering and Advanced Computing

See next page for additional authors

Published IEEE Transactions on Cloud Computing, 2022, 10 (1), 474-490. DOI: 10.1109/TCC.2019.2936481

Lightweight and Expressive Fine-Grained Access Control for Healthcare Internet-of-Things

Shengmin Xu, Yingjiu Li, *Member, IEEE*, Robert H. Deng, *Fellow, IEEE*, Yinghui Zhang, *Member, IEEE*, Xiangyang Luo, and Ximeng Liu, *Member, IEEE*

Abstract—Healthcare Internet-of-Things (IoT) is an emerging paradigm that enables embedded devices to monitor patients vital signals and allows these data to be aggregated and outsourced to the cloud. The cloud enables authorized users to store and share data to enjoy on-demand services. Nevertheless, it also causes many security concerns because of the untrusted network environment, dishonest cloud service providers and resource-limited devices. To preserve patients' privacy, existing solutions usually apply cryptographic tools to offer access controls. However, fine-grained access control among authorized users is still a challenge, especially for lightweight and resource-limited end-devices. In this paper, we propose a novel healthcare IoT system fusing advantages of attribute-based encryption, cloud and edge computing, which provides an efficient, flexible, secure fine-grained access control mechanism with data verification in healthcare IoT network without any secure channel and enables data users to enjoy the lightweight decryption. We also define the formal security models and present security proofs for our proposed scheme. The extensive comparison and experimental simulation demonstrate that our scheme has better performance than existing solutions.

Index Terms—Internet-of-Things, access control, cloud computing, edge computing, attribute-based encryption

1 Introduction

Internet-of-Things (IoT) is a novel paradigm for machine-to-machine communication by connecting the various physical devices through the Internet to collect, analyze and exchange data. As agreed by both the academic and industrial communities, IoT is the future of ubiquitous computing. According to the reports from International Data Corporation (IDC) [1], there will be 80 billion connected devices in 2025. The real-time data created by embedded devices, machine-to-machine communications, and IoT networks will reach about 50 Zettabyte (ZB), and it helps to incur 163 ZB of data in that year. IoT has been extensively adapted to many fields such as smart home, environment monitoring, logistics, etc., and healthcare is certainly one of the most promising scenarios of its applications. The MGC architecture (i.e., e*M*bedded devices, *G*ateways, and *C*loud architecture) of healthcare IoT network is illustrated in Fig. 1. Embedded devices collect information about patients by monitoring the corresponding vital signals. The gateway device aggregates these data from embedded devices and uploads them to the remote cloud server. Medical staffs are allowed to fetch the outsourced data in the cloud through end-devices. However, these devices are usually resource-limited, and the untrusted network environment is vulnerable to a variety of threats. Hence, it is not straightforward to directly apply MGC architecture in real-world applications due to a variety of practicality concerns.

In a healthcare IoT network, sensitive personal information, such as health records and treatment reports, is transferred in the untrusted network environment. According to the report from Thales Data Threat [2], about 49 percent of healthcare organizations use IoT to collect sensitive data in 2018. To protect data confidentiality, cryptographic tools such as public-key and identity-based cryptosystems would be potential solutions to provide access control for authorized devices. However, these methods only provide coarse-level data sharing, which requires the encryptor to know the information of the corresponding decryptor in advance to produce the ciphertext; this strategy is a one-to-one data sharing mechanism (i.e., one ciphertext for one decryptor rather than for a group of decryptors). In healthcare IoT network, the confidential data usually is shared with multiple users who have similar or related sets of attributes with the unknown group size, such as the group of attending physicians and the medical staff on duty. To provide fine-grained access control, attribute-based encryption (ABE) was introduced as a promising tool, which allows one ciphertext to be shared with multiple authorized users. However, the

- *S. Xu, Y. Li, and R.H. Deng are with the Secure Mobile Centre, School of Information Systems, Singapore Management University, Singapore 178902. E-mail: {smxu, yjli, robertdeng}@smu.edu.sg.*
- *Y. Zhang the Secure Mobile Centre, School of Information Systems, Singapore Management University, Singapore 178902, and also with the National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications, Xi'an 710121, P.R. China. E-mail: yhzhaang@163.com.*
- *X. Luo is with the State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, P.R. China. E-mail: Luoxy_ieu@sina.com.*
- *X. Liu is the College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350108, P.R. China and also with the Secure Mobile Centre, School of Information Systems, Singapore Management University, Singapore 178902. E-mail: snbnix@gmail.com.*

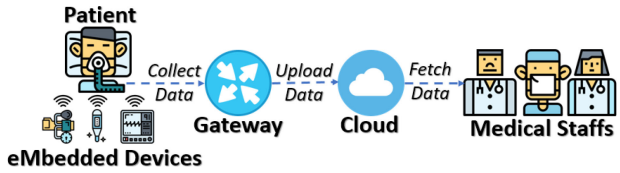


Fig. 1. MGC architecture of healthcare IoT network.

standard ABE have a number of limitations which need to be addressed before they can be applied in practice.

One disadvantage of ABE is its significant decryption overhead. ABE cannot directly apply to healthcare IoT network since lightweight devices in the IoT ecosystem have many constraints such as computational resource, battery lifetime and bandwidth cost. Specifically, medical staffs fetch data collected by the embedded devices from the remote cloud server via lightweight end-devices such as smartphone and tablet. Directly operating ABE decryption on these devices is challenging because of their limited computational resource and battery power. To mitigate this issue, it is critical to require another party to provide computational power to unite the lightweight devices and ABE mechanism. One possible solution is the cloud computing which pre-processes ciphertexts and produces short ciphertexts to end-devices. However, the spending on data transmission and the amount of data transport is becoming the bottleneck, the feedback from the remote cloud server is lag and even lost because of the unreliable data transmission in the network. The other practical solution is edge computing [3], [4], [5], which allows data to process at the edge of the network to address the concerns of lightweight devices (e.g., response time requirement, battery life constraint, and bandwidth cost saving).

Another drawback of ABE is not sufficiently flexible for data sharing, especially with large amount of lightweight devices in healthcare IoT network. There are two main flavors of standard ABE. In Key-Policy ABE (KP-ABE), secret keys based on access trees, and ciphertexts are encrypted over a set of attributes. Alternatively, Ciphertext-Policy ABE (CP-ABE) inverts the relationship between ciphertexts and keys. These two types of ABE cannot provide content-based and role-based access control simultaneously [6]. KP-ABE only offers content-based access control. Data users are explicitly authorized to access the collection of records matching certain requirement, which supports individuals whose roles are not precisely defined such as contractors or medical researchers. For example, some data owners might be given access only to records within certain periods and certain ranges of diseases. CP-ABE only provides role-based access control. The access control policy is based on roles associated with authorized accessors. The medical staff has attributes including title, patient list, and specialty. Each protected data embeds a complex access control policy specifying the type of authorized group. To aggregate advantages of role-based and content-based access control, dual-policy ABE (DP-ABE) was introduced [7]. In DP-ABE, a ciphertext is generated according to a subjective access tree and a set of an objective attributes simultaneously, and a secret key is generated according to a set of subjective attributes and objective access tree. Hence, DP-ABE offers a flexible data sharing mechanism for healthcare IoT system, data users (e.g., Affiliation*: Hospital*; Occupation*: Doctor*; Department*:

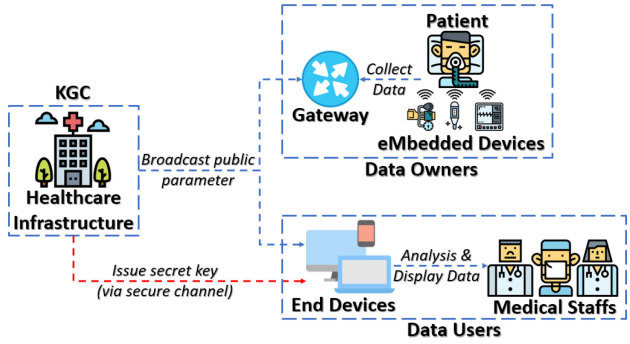


Fig. 2. Distributing system parameter.

Cardiology for role-based access control) have authority to reveal some particular information (e.g., EmergencyInfo*: BloodType* and EmergencyInfo*: AllergyMedicine* for content-based access control) for emergency cases. Either KP-ABE or CP-ABE cannot work in this scenario individually. Therefore, DP-ABE is the promising cryptographic tool to offer content-based and role-based access controls within one cryptographic system.

The standard ABE also suffers key distribution problem. Many cryptographic tools including DP-ABE remain the problem of key distribution. Specifically, the key generation center (KGC) is required to establish secure channels with each data user as shown in Fig. 2 to distribute secret keys. The secure channel is very expensive since both parties such as healthcare infrastructure (as KGC) and data users must be passed the mutual authentication to negotiate a session key to build a secure channel to distribute secret keys against the untrusted network environment. The agreed session key will be discarded after data users receive secret decryption valid keys. Hence, the scalability and usability still remain problems.

Besides drawbacks of standard ABE, untrusted cloud service providers also threat outsourced data integrity. The sensitive data is encrypted and uploaded to remote could service providers to enjoy cost savings and productivity enhancements. However, the untrusted cloud service providers may modify the data to confuse users. The data integrity checking remains one of the challenge issues in many cloud-based applications.

1.1 Our Contributions

In this paper, we introduce an efficient, secure and expressive fine-grained access control mechanism for lightweight devices called server-aided dual-policy attribute-based encryption (*SA-DP-ABE*) to build healthcare IoT network to solve the above problems simultaneously.

Practical and Expressive Fine-Grained Access Control. Many healthcare IoT cryptosystems [8], [9], [10], [11], [12], [13], [14] only consider either content-based access control or role-based access control. To support more efficient access control (i.e., content-based and role-based access control simultaneously) for the lightweight devices, we introduce a novel scheme called *SA-DP-ABE* by refining and fusing many technologies, i.e., DP-ABE [7], edge computing [3] and outsourced ABE [15]. This combination is a challenge since these techniques have individual purposes and constructions, which leads to the combination to be non-trivial. Specifically,

the DP-ABE provides content-based, and role-based access controls simultaneously. Edge computing offers the immediate response, computational resource and data storage to lightweight end-devices. The outsourced ABE eliminates the security channel to diminish costs during key distribution. Besides, our security model not only considers data users are untrusted but also permits collusion attacks among the remote cloud server, the edge server and data users to reveal unauthorized messages. Furthermore, we introduce the modified q assumption derived from the assumptions in [16] and prove the security of the proposed assumption in the generic group model. After that, we give the formal proof for our *SA-DP-ABE* scheme based on the modified q assumption. We then prove our proposed healthcare IoT system is secure based on the *SA-DP-ABE* scheme.

Lightweight Decryption Cost for End-Devices. Our scheme only requires the constant-size public parameters, as well as all operations and elements are in prime-order groups. In comparison, the previous DP-ABE schemes are less efficient, which require either linear-size public parameter [7] or composite-order group [17]. Furthermore, the decryption cost in our scheme is constant since expensive decryption tasks are processed by the edge server rather than data users themselves. The experimental analysis as shown in Section 6 demonstrates our scheme has significantly better performance regarding computational cost and experimental simulation than previous works [7], [17].

Scalable Key Distribution for Healthcare Infrastructure. Many cryptosystems require secure channels between KGC and each user to distribute secret keys against untrusted network environment. However, the secure channel is very expensive that affects the scalability and usability of the underlying systems. To overcome this issue, we require that each data user to generate a secret and public key pair, and that KGC generates attribute-based keys depending on the public keys of corresponding data users to eliminate the secure channel, where attribute-based keys refer to transformation keys since they are published publicly. Furthermore, with the help of the edge server, data users need not store the corresponding transformation keys to reduce the storage cost of data users, and some heavy computation tasks are also performed by the edge server to mitigate the workload of data users.

Data Integrity and Hybrid Data Encryption. Data integrity checking prevents compromised data from malicious cloud servers, which is performed by data users based on randomness extractors and collision-resistant hash functions, such that the malicious cloud server cannot forge a valid verification key for any compromised data. Besides, data integrity checking also prevents the malicious edge server, which may not transform the ciphertext honestly. Hybrid encryption is used to reduce the cost of data sharing. Specifically, we use *SA-DP-ABE* to encrypt a random key K, and then encrypt data files using a symmetric-key encryption scheme (e.g., AES) with K. The valid data users will retrieve K and then reveal corresponding files. By aggregating above techniques with hybrid encryption and randomness extractors, our healthcare IoT cryptosystem not only provides flexible fine-grained data sharing with lightweight cost, but also allows data owners to verify the integrity of outsourced data. It is worth to notice that this hybrid encryption technology is also called key encapsulated mechanism (KEM), which requires to choose a session key (encrypted in ABE scheme) and hash this key to be a symmetric key (to encrypt files). However, the simple hash function incurs many security issues including entropy loss and non-uniform distribution. To address this problem, the randomness extractor is used for producing a uniform symmetric key from non-uniform session key [18].

1.2 Related Work

Healthcare System. Wan et al. [8] proposed a fine-grained access control data sharing system based on hierarchical ABE, and it cannot apply to the IoT ecosystem since the huge computational overhead in data owners. Abbas and Khan [9] provided a survey about access control in the e-Health cloud, but they did not consider the how to apply IoT devices to e-Health cloud. Yang et al. applied ABE to provide fine-grained access control and keyword search in the healthcare system. However, the massive computational workloads of decryption and keyword search are allocated to data users. Yang et al. [11], [12], [13], [14] provided a serial of fine-grained healthcare data sharing systems via the cloud. However, the workload of end-users still heavy and cannot offer content-based and role-based access control simultaneously. Therefore, it is desirable to have a solution in lightweight devices with flexible access control.

Standard ABE. There are two main flavors of ABE. In KP-ABE, users' secret keys are based on access trees, and ciphertexts are encrypted over a set of attributes. The encryptor has no control over who has access to the data except by choosing descriptive attributes for the data. Fuzzy IBE (FIBE) as the initial work of KP-ABE with k-out-of-n policy was introduced by Sahai and Waters [19]. To enrich the expression, Goyal et al. [20] provided KP-ABE with monotonic span programs and Ostrovsky et al. [21] proposed KP-ABE supporting non-monotonic access structures. Attrapadung et al. [22] and Rouselakis and Waters [16] then proposed KP-ABE with constant-size ciphertexts. Alternatively, in CP-ABE, access trees are used to encrypt data and users' secret keys associate a set of attributes. The encryptor has to manage the access tree to specify the users' access policy. The seminal work was introduced by Bethencourt et al. [23] with two-level random masking methodology. To improve the performance, Waters [24] introduced the first selectively secure CP-ABE under the non-standard assumption, and Rouselakis and Waters [16] provided CP-ABE with the large universe. Unfortunately, the security of above CP-ABE schemes is based on either selectively secure or the security depending on the non-standard assumptions. Lewko et al. [25] provided the first fully secure CP-ABE based on the dual encryption system [26]. To imporve the efficiency, Zhang et al. [27] provided the constant-size CP-ABE. Recently, Koppula and Waters [28] provided a transformation to improve the security of ABE in chosen plaintext secure to chosen ciphertext secure. Both KP-ABE and CP-ABE can prevent any users to access unauthorized data, even if the outsourced data is in the untrusted server and collusion attacks among multiple unauthorized users. Therefore, ABE schemes are attractive in the cloud-based cryptosystems. However, directly using them to build cryptosystem suffers many problems explained below.

Dual-Policy ABE. Standard ABE scheme fails to provide content-based and role-based access control simultaneously. To overcome this problem, DP-ABE was introduced in [7],

[17], which is a conjunctively combined between two types of ABE. Ciphertexts are specified access trees and a set of attributes simultaneously, and secret keys are also required to specified a set of attributes and access trees. There are two types of DP-ABE: sequential DP-ABE and parallel DP-ABE. In sequential DP-ABE, receivers will be able to decrypt ciphertexts if who pass both restrictions. In parallel DP-ABE (sometimes called one-policy DP-ABE), receivers only required to satisfy one of two limitations to reveal messages. In this paper, we consider the sequential DP-ABE. In the rest of this paper, we only consider sequential DP-ABE since the transformation from sequential DP-ABE to paralleled DP-ABE is not complicated [7].

Outsourced ABE. One drawback of ABE is large overhead. To address this issue, the outsourced ABE was proposed in [15] to enable a third party to transform ciphertexts to partially decrypted ciphertexts to reduce the workload of data users for applying in lightweight end-devices. This concept has been widely adapted to various cryptosystems such as revocable identity-based encryption [29] and revocable attribute-based encryption [30], [31]. In cybersecurity, this concept is also called edge computing to optimize the IoT ecosystem [3]. An edge server such as router closed to data users can be configured to provide a variety of services and reduce the transmission cost by accessing the remote cloud server. In cryptosystems, the primary responsibility of the edge server is to reduce the workload of data users such as moving the majority of the cost of data decryption into the edge server.

1.3 Outline

In Section 2, we introduce the preliminaries of proposed schemes, including the access structure to achieve fine-grained access control, hard problems of our proposed schemes, definition and security model of *SA-DP-ABE*, and randomness extractor to improve usability. In Section 3, we present the system model and formal definitions, including the system architecture of the proposed healthcare IoT system, the corresponding threat model and the formal definition of the security model. In Section 4, we propose the workflow and generic construction of our system based on the hybrid encryption and its formal proof. In Section 5, we introduce a novel *SA-DP-ABE* scheme including the concrete construction and formal proofs, which is an important building block to realize healthcare IoT system. In Section 6, we give performance analysis, including functionality, computational complexity, and experimental simulation. We then summary this paper in Section 7.

2 Preliminaries

2.1 Notation

Let $\mathbb{N}$ denote the set of all natural numbers. For $n \in \mathbb{N}$, let $[n]$ be a set of numbers from 1 to n, denoted $[n] = \{1, 2, \ldots, n\}$. If $\mathcal{S}$ is a finite set, let $\mathcal{S}_i$ denote the ith value in the set $\mathcal{S}$. To simplicity and prevent any understanding to readers, we give the Table 1 to describe some frequently used notations in the rest of the paper.

TABLE 1
Notation Table

Notation	Description
$\mathcal{SA}$	*Abbreviation of SA-DP-ABE scheme*
$\mathcal{SE}$	*Symmetric encryption scheme*
$\mathcal{HS}$	*Healthcare IoT system*
Ω_o	*Universe of objective attribute*
Ω_s	*Universe of subjective attribute*
$\mathcal{P}_o$	*Objective policies*
$\mathcal{P}_s$	*Subjective policies*
ω	*A set of objective attribute s.t.* $\omega \in \Omega_o$
ψ	*A set of subjective attribute s.t.* $\psi \in \Omega_s$
$\mathbb{O}$	*An objective access structure s.t.* $\mathbb{O} \in \mathcal{P}_o$
$\mathbb{S}$	*A subjective access structure s.t.* $\mathbb{S} \in \mathcal{P}_o$
m	*Message in plaintext*
c	*Ciphertext message*
c'	*Partially decrypted ciphertext (by edge server)*
KGC	*Entity: key generate center*
DO	*Entity: data owner*
DU	*Entity: data user*
ES	*Entity: edge server*
CS	*Entity: cloud server*

2.2 Bilinear Map

Let $\mathbb{G}$ and $\mathbb{G}_T$ be two cyclic multiplicative groups of prime order p and g be a generator of $\mathbb{G}$. The map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is said to be an admissible bilinear pairing if the following properties hold.

- *Bilinearity*: for all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p$, $e(u^a, v^b) = e(u, v)^{ab}$.
- *Non-degeneration*: $e(g, g) \neq 1$.
- *Computability*: it is efficient to compute $e(u, v)$ for any $u, v \in \mathbb{G}$.

2.3 Linear Secret Sharing Scheme

We recall the definition of linear secret sharing scheme (LSSS) [32]. A LSSS policy is of the type (M, ρ), where M is a $\ell \times n$ matrix over the base field $\mathbb{F}$ and ρ is a mapping function from the set $[\ell]$ to the attribute universe. A policy (M, ρ) satisfies an attribute set ψ if $(1, 0, \ldots, 0) \in \mathbb{F}^n$ is contained in $\mathsf{Span}_{\mathbb{F}}(M_i : \rho(i) \in \psi)$, where M_i is the ith row of M. For simplicity, we use $\mathbb{F} = \mathbb{Z}_p$ for the rest of the paper.

2.4 Assumptions

We propose the modified q assumption, which is derived from q-1 and q-2 assumptions [16], [33]. The purpose of this assumption is to allow the underlying KP-ABE and CP-ABE in our proposed *SA-DP-ABE* share the same master secret key in our formal security proof.

Recall the notations $[i : x]$ and $[i : x, i' : y]$ in [33]. Let $[i : x]$ and $[i : x, i' : y]$ denote the row vectors in $\mathbb{Z}_p^{1\times q}$ in which all components equal to 0 except the ith component for the first vector and the i, i'th component for the second. The non zero elements are x for the first vector and x, y for the i, i'th possible, respectively, of the second vector. Below we present the formal definition of modified q assumption.

Definition 1 (Modified q assumption). *Suppose a challenge runs a group generator algorithm $\mathcal{G}(1^\lambda) \to (p, \mathbb{G}, \mathbb{G}_T, e)$ and chooses generator $g \in \mathbb{G}$, and $2q + 2$ random exponents $a, s, b_1, b_2, \ldots, b_q, c_1, c_2, \ldots, c_q \in \mathbb{Z}_p$ as well as following terms:*

$g, g^s, g^{(sa)^2}$	
$g^{a^i}, g^{b_j}, g^{sb_j}, g^{a^i b_j}, g^{a^i/b_j^2}$	$\forall (i,j) \in [q,q]$
$g^{a^i b_j / b_{j'}^2}$	$\forall (i,j,j') \in [2q,q,q]$ with $j \neq j'$
g^{a^i/b_j}	$\forall (i,j) \in [2q,q]$ with $i \neq q+1$
$g^{sa^i b_j/b_{j'}}, g^{sa^i b_j/b_{j'}^2}$	$\forall (i,j,j') \in [q,q,q]$ with $j \neq j'$
$g^{c_k}, g^{sac_k}, g^{sa/c_k}$	$\forall k \in [q]$
$g^{sa^2 c_k}, g^{a^q/c_i^2}, g^{a^{q^2}/c_k^2}$	$\forall k \in [q]$
$g^{sac_k/c_{k'}}, g^{a^q c_k/c_{k'}^2}$	$\forall (k,k') \in [q,q]$ with $k \neq k'$
$g^{sa^{q+1}c_k/c_{k'}^2}, g^{(sa)^2 c_k/c_{k'}}$	$\forall (k,k') \in [q,q]$ with $k \neq k'$

By given the above terms, the modified q assumption is that no polynomial-time adversary is able to distinguish the term $e(g,g)^{sa^{q+1}}$ from a random term $R \in \mathbb{G}_T$ with more than a negligible advantage.

Remark. The modified q assumption is derived from the q-1 and q-2 assumptions [16]. The absence of g^{a^{q+1}/b_j} in the forth line is to prevent trivially attack by pairing it with the corresponding g^{sb_j} term. On the other hand, the terms $g^{a^{q+1}b_j/b_{j'}^2}$ in the third line and $g^{sa^{q+1}c_k/c_{k'}^2}$ in the last line are given, and this poses no problems in generic group model since $j \neq j$ and $k \neq k'$ and by possible pairing the adversary cannot get rid of the b_j's and c_k's. See Appendix D for further details.

2.5 Definition of SA-DP-ABE

We propose the *SA-DP-ABE* based on definitions of DP-ABE [7], [17] and outsourced ABE [15]. Our scheme not only provides content-based and role-based access simultaneously, but also offers the lightweight decryption mechanism for lightweight end-devices.

Definition 2 (SA-DP-ABE). *A* SA-DP-ABE *with lightweight decryption cost (or simply $\mathcal{SA}$) scheme $\mathcal{SA}$ with the subjective attribute universe Ω_s and the objective universe Ω_o that supports the subjective policies $\mathcal{P}_s$ and the objective policies $\mathcal{P}_o$ with the identity space $\mathcal{I}$ and the message space $\mathcal{M}$ involves four types of entities: a KGC, data users, data owners and an untrusted server, and consists of the following six algorithms:*

$\mathcal{SA}$.Setup $(1^\lambda) \rightarrow (pp, msk)$: The probabilistic setup algorithm is run by KGC. It takes a security parameter $\lambda \in \mathbb{N}$ as input, and outputs the public parameter pp and the master secret key msk.

$\mathcal{SA}$.KeyGen $(pp, id) \rightarrow (sk_{id}, pk_{id})$: The probabilistic key generation algorithm is run by data users. It takes the public parameter pp and an identity $id \in \mathcal{I}$ as input, and outputs the secret key sk_{id} and the corresponding public key pk_{id}.

$\mathcal{SA}$.TKGen $(pp, msk, id, pk_{id}, (\psi, \mathbb{O})) \rightarrow tk_{id}$: The probabilistic transformation key generation algorithm is run by KGC. It takes the public parameter pp, the master secret key msk, the identity $id \in \mathcal{I}$, the corresponding public key pk_{id}, a set of subjective attributes $\psi \in \Omega_s$ and an objective access structure $\mathbb{O} \in \mathcal{P}_o$ as input, and outputs the transformation key tk_{id}.

$\mathcal{SA}$.Enc $(pp, m, (\mathbb{S}, \omega)) \rightarrow c$: The probabilistic encryption algorithm is run by data owners. It takes the public parameter pp, a message $m \in \mathcal{M}$, a subjective access structure $\mathbb{S} \in \Omega_s$ and a set of objective attributes $\omega \in \mathcal{P}_o$ as input, and outputs the ciphertext c.

$\mathcal{SA}$.Transform $(pp, c, (\mathbb{S}, \omega), id, tk_{id}, (\psi, \mathbb{O})) \rightarrow c'/\perp$: The deterministic transformation algorithm is run by the untrusted server. It takes the public key pk, the ciphertext c and the associated pair of the subjective access structure $\mathbb{S} \in \mathcal{P}_s$ and the set of objective attributes $\omega \in \Omega_o$, an identity $id \in \mathcal{I}$ and the corresponding transformation key tk_{id} associated pair of the set of subjective attribute $\psi \in \omega_s$ and the objective access structure $\mathbb{O} \in \mathcal{P}_o$ as input, and outputs the partially decrypted ciphertext c' or a failure symbol $\perp$.

$\mathcal{SA}$.Dec $(pp, id, sk_{id}, c') \rightarrow m$: The deterministic decryption algorithm is run by data users. It takes the public parameter pp, an identity $id \in \mathcal{I}$, the corresponding secret key sk_{id}, the partially decrypted ciphertext c' as input, and outputs the message m.

The consistency condition is for all security parameter $\lambda \in \mathbb{N}$, all public parameter pp and master secret key msk output by the setup algorithm, and all secret and public key pairs (sk_{id}, pk_{id}) output by the key generation algorithm, all messages $m \in \mathcal{M}$, and the ciphertext c and the partially decrypted ciphertext as

$$\mathcal{SA}.\mathsf{Enc}(pp, m, (\mathbb{S}, \omega)) \rightarrow c,$$
$$\mathcal{SA}.\mathsf{Transform}(pp, c, (\mathbb{S}, \omega), id, tk_{id}, (\psi, \mathbb{O})) \rightarrow c',$$

we have $\mathcal{SA}.\mathsf{Dec}(pp, id, sk_{id}, c') = m$ with probability 1.

Green et al. [15] introduced the security model for outsourced ABE, we refine this model and describe the security model called selectively indistinguishable against chosen plaintext attack (sIND-CPA) the following:

Definition 3 (sIND-CPA in $\mathcal{SA}$). *A* SA-DP-ABE *scheme consists of six algorithms given above. For an adversary $\mathcal{A}$, we define the following experiment:*

$$\begin{aligned}
&\textbf{Experiment } \mathbf{Exp}^{\mathsf{sIND\text{-}CPA}}_{\mathcal{SA},\mathcal{A}}(1^\lambda)\\
&\quad (\mathbb{S}^*, \omega^*) \leftarrow \mathcal{A}(1^\lambda); (pp, msk) \leftarrow \mathcal{SA}.\mathsf{Setup}(1^\lambda);\\
&\quad (m_0, m_1) \leftarrow \mathcal{A}^{\mathcal{O}}(pp); b \leftarrow \{0,1\};\\
&\quad c^* \leftarrow \mathcal{SA}.\mathsf{Enc}(pp, m_b, (\mathbb{S}^*, \omega^*)); b' \leftarrow \mathcal{A}^{\mathcal{O}}(c^*);\\
&\quad \text{If } b = b' \text{ return 1 else return 0.}
\end{aligned}$$

$\mathcal{O}$ denotes a set of oracles $\{\mathcal{O}^{\mathcal{SA}}_{\mathsf{create}}(\cdot,\cdot), \mathcal{O}^{\mathcal{SA}}_{\mathsf{corrupt}}(\cdot)\}$:

$\mathcal{O}^{\mathcal{SA}}_{\mathsf{create}}(\cdot,\cdot)$ is a creating oracle that allows $\mathcal{A}$ to query on any set of subjective attribute $\psi \in \Omega_s$ and any objective access structure $\mathbb{O} \in \mathcal{P}_o$. It picks a random identity $id \in \mathcal{I}$, then runs $\mathcal{SA}.\mathsf{KeyGen}(pp, id)$ to obtain the corresponding secret and public key pair (sk_{id}, pk_{id}) and runs $\mathcal{SA}.\mathsf{TKGen}(pp, msk, id, pk_{id}, (\psi, \mathbb{O}))$ to obtain the corresponding transformation key tk_{id}. Finally, it returns the transformation key tk_{id} to $\mathcal{A}$.

$\mathcal{O}^{\mathcal{SA}}_{\mathsf{corrupt}}(\cdot)$ is a corrupt oracle that allows $\mathcal{A}$ to query on any identity $id \in \mathcal{I}$, and it returns the corresponding secret key sk_{id} if it has been derived in the creating oracle. If no such key exists, then it returns the error symbol $\perp$.

$\mathcal{A}$ is allowed to issue above oracles with the restriction such that for each corrupted users, the corresponding set of subjective attribute $\psi \in \Omega_s$ and objective access structure $\mathbb{O} \in \mathcal{P}_o$ cannot satisfy the condition of the challenge subjective access structure $\mathbb{S}^*$ and set of objective attribute ω^* simultaneously.

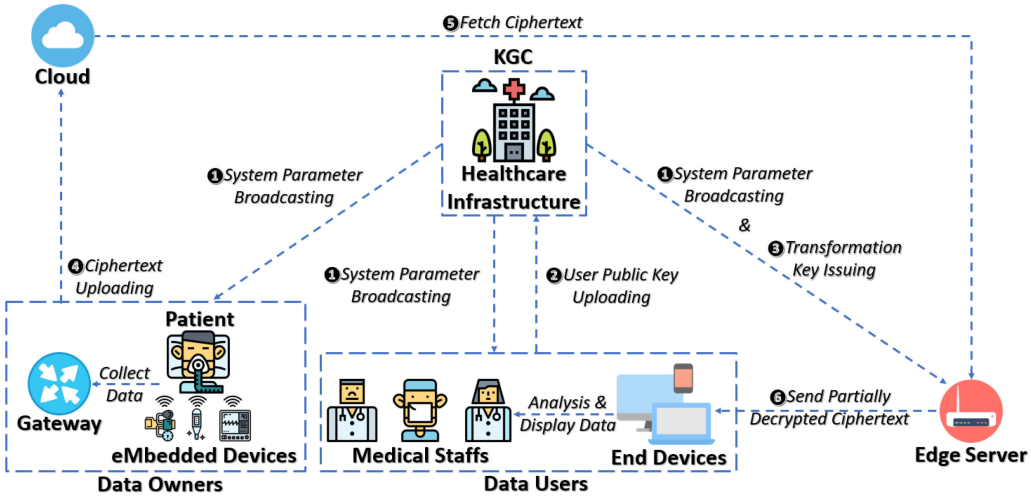


Fig. 3. System model of healthcare IoT network.

A SA-DP-ABE *scheme is said to be* sIND-CPA *secure if for any probabilistic polynomial-time adversary* $\mathcal{A}$*, the following advantage is negligible:*

$$\mathbf{Adv}^{\mathsf{sIND\text{-}CPA}}_{\mathcal{SA},\mathcal{A}}(1^\lambda) = \left|\Pr[\mathbf{Exp}^{\mathsf{sIND\text{-}CPA}}_{\mathcal{SA},\mathcal{A}}(1^\lambda) = 1] - 1/2\right|.$$

2.6 Definition of Symmetric Encryption

Definition 4 (Symmetric Encryption). *A symmetric encryption scheme* $\mathcal{SE}$ *with the key space* $\mathcal{K}$ *and the message space* $\mathcal{M}$ *involves two types of entities: data owners and data users, and consists of following two algorithms:*

$\mathcal{SE}.\mathsf{Enc}(K, m) \rightarrow c$*: The deterministic encryption algorithm is run by data owners. It takes a key* $K \in \mathcal{K}$ *and the message* $m \in \mathcal{M}$ *as input, and outputs the ciphertext* c*.*

$\mathcal{SE}.\mathsf{Dec}(K, c) \rightarrow m$*: The deterministic decryption algorithm is run by data users. It takes a key* $K \in \mathcal{K}$ *and the ciphertext* c *as input, and outputs the message* m*.*

For the security model of symmetric encryption, we only consider the one-time semantically secure (SS) since our proposed healthcare IoT system is based on the key encapsulated mechanism where the key $K \in \mathcal{K}$ is a session key and would not appear in the other sessions. Below we present the details:

Definition 5 (SS in $\mathcal{SE}$**).** *A symmetric key encryption scheme consists of two algorithms given above. For an adversary* $\mathcal{A}$*, we define the following experiment:*

$$\begin{aligned}&\text{Experiment } \mathbf{Exp}^{\mathsf{SS}}_{\mathcal{SE},\mathcal{A}}(1^\lambda)\\ &(m_0, m_1) \leftarrow \mathcal{A}(1^\lambda); b \leftarrow \{0,1\}; K \leftarrow \mathcal{K};\\ &c \leftarrow \mathcal{SE}.\mathsf{Enc}(K, m_b); b' \leftarrow \mathcal{A}(c);\\ &\text{If } b = b' \text{ return } 1 \text{ else return } 0.\end{aligned}$$

A symmetric key is said to be one-time semantically secure if for any probabilistic polynomial-time adversary $\mathcal{A}$*, the following advantage is negligible:*

$$\mathbf{Adv}^{\mathsf{SS}}_{\mathcal{SE},\mathcal{A}}(1^\lambda) = \left|\Pr[\mathbf{Exp}^{\mathsf{SS}}_{\mathcal{SE},\mathcal{A}}(1^\lambda) = 1] - 1/2\right|.$$

2.7 Randomness Extractor

We recall the definition of randomness extractor [18]. For a discrete distribution X over $\mathcal{X}$, $H_\infty(X)$ is the min-entropy $-\log\left(max_{\omega\in\mathcal{X}} \Pr[X = \omega]\right)$, and $\tilde{H}_\infty(X|Y)$ is the average min-entropy of X conditioned on Y as $-\log\left(E_{y\leftarrow Y}[2^{-H_\infty(X|Y=y)}]\right)$.

We also recall the lemma that will be used in our proof.

Lemma 1 ([29]). *Let* X, Y *and* Z *be random variables. If* Y *has at most* 2^r *possible values, then* $\tilde{H}_\infty(X|(Y,Z)) \geq \tilde{H}_\infty(X|Z) - r$*.*

Definition 6 (Randomness Extractor). *The function* $\mathsf{Ext} : \mathcal{X} \times \{0,1\}^{\bar{t}} \rightarrow \mathcal{Y}$ *is an average-case* (k, ϵ)*-strong extractor if for all random variables* (X, Z) *such that* $X \in \mathcal{X}$ *and* $\tilde{H}_\infty(X|Z) \geq k$*, we have the statistical distance* SD *between two distribution within* ϵ*, such that* $\mathsf{SD}((Z, s, \mathsf{Ext}(\mathsf{X}, \mathsf{s})), (Z, s, U_\mathcal{Y})) \leq \epsilon_\mathcal{H}$*, where* $s \in \{0,1\}^{\bar{t}}, U_\mathcal{Y} \in \mathcal{Y}$*,* $(Z, s, \mathsf{Ext}(\mathsf{X}, \mathsf{s}))$ *and* $(Z, s, U_\mathcal{Y})$ *represent two distribution.*

3 System Model and Formal Definitions

In this section, we introduce our system construction including the functionality of each entity and the system architecture with the formal definition, and the security about the system threat models.

3.1 Functionalities of Entities in Healthcare IoT System

Our healthcare IoT system includes five types of entities: a key generate center (*KGC*), data owners (*DOs*), data users (*DUs*), an edge server (*ES*) and a cloud server (*CS*), as shown in Fig. 3. We assume *CS* is a remote server and *ES* is a server closer to *DUs*, and *CS* and *ES* can be merged into a single entity if we do not consider the geographical issues. Moreover, our system model does not require any secure channel. The characteristics and functions of each entity are described as follows:

KGC *KGC* has responsibilities to initialize the whole system, manage the credentials of *DUs* and issue transformation keys to *ES*.

DOs *DOs* represent a set of users who have confidential messages uploaded to *CS* for distributing them to *DUs* securely.

DUs *DUs* are a group of users who have valid credentials from *KGC* and obtained data from *ES*. Furthermore, each *DU* only has limited resources to manage data storage and computation in our model.

ES *ES* manages transformation keys of *DUs* and helps them to decrypt ciphertexts partially.

CS *CS* is a remote server who has a vast amount of storage to accommodate the data from *DOs* and distributes ciphertexts to *ES*.

3.2 Syntax of Healthcare IoT System

Definition 7 (Healthcare IoT System). *Healthcare IoT system with lightweight decryption and data verification (or simply $\mathcal{HS}$) with the subjective attribute universe Ω_s and the objective attribute universe Ω_o that supports the subjective policies $\mathcal{P}_s$ and the objective policies $\mathcal{P}_o$ with the identity space $\mathcal{I}$ and the message space $\mathcal{M}_{\mathcal{SE}}$ involves five types of entities as the above and consists of following six algorithms:*

*$\mathcal{HS}$.*Setup*$(1^\lambda) \rightarrow (pp, msk)$: The probabilistic setup algorithm is run by* KGC*. It takes a security parameter $\lambda \in \mathcal{N}$ as input, and outputs the public parameter pp and the master secret key msk.*

*$\mathcal{HS}$.*KeyGen*$(pp, id) \rightarrow (pk_{id}, sk_{id})$: The probabilistic key generation algorithm is run by* DUs*. It takes the public parameter pp and an identity $id \in \mathcal{I}$ as input, and outputs the secret key sk_{id} and the corresponding public key pk_{id}.*

*$\mathcal{HS}$.*TKGen*$(pp, msk, id, pk_{id}, (\psi, \mathbb{O})) \rightarrow tk_{id}$: The probabilistic transformation key generation algorithm is run by* KGC*. It takes the public parameter pp, the master secret key msk, the identity $id \in \mathcal{I}$, the corresponding public key pk_{id}, a set of subjective attributes $\psi \in \Omega_s$ and an objective access structure $\mathbb{O} \in \mathcal{P}_o$ as input, and outputs the transformation key tk_{id}.*

*$\mathcal{HS}$.*Enc*$(pp, m, (\mathbb{S}, \omega)) \rightarrow (c, vk_m)$: The probabilistic encryption algorithm is run by* DOs*. It takes the public parameter pp, a message $m \in \mathcal{M}$, a subjective access structure $\mathbb{S} \in \Omega_s$ and a set of objective attributes $\omega \in \mathcal{P}_o$ as input, and outputs the ciphertext c and the corresponding verification key vk_m.*

*$\mathcal{HS}$.*Transform*$(pp, c, (\mathbb{S}, \omega), id, tk_{id}, (\psi, \mathbb{O})) \rightarrow c'/\perp$: The deterministic transformation algorithm is run by* ES*. It takes the public parameter pp, the ciphertext c associated pair of the subjective access structure $\mathbb{S} \in \mathcal{P}_s$ and the set of objective attributes $\omega \in \Omega_o$, an identity $id \in \mathcal{I}$ and the corresponding transformation key tk_{id} associated pair of the set of subjective attribute $\psi \in \Omega_s$ and the objective access structure $\mathbb{O} \in \mathcal{P}_o$ as input, and outputs the partially decrypted ciphertext c' or a failure symbol $\perp$.*

*$\mathcal{HS}$.*Dec*$(pp, id, sk_{id}, c', vk_m) \rightarrow m/\perp$: The deterministic decryption algorithm is run by* DUs*. It takes the public parameter pp, an identity $id \in \mathcal{I}$, the corresponding secret key sk_{id}, the partially decrypted ciphertext c' as input, and outputs the message m.*

The consistency condition requires that for all security parameter $\lambda \in \mathbb{N}$, all public parameter pp and master secret key msk output by the setup algorithm, and all secret and public key pairs (sk_{id}, pk_{id}) output by the key generation algorithm, all messages $m \in \mathcal{M}$, and the ciphertext c and the partially decrypted ciphertext as

$$\mathcal{HS}.\mathsf{Enc}(pp, m, (\mathbb{S}, \omega)) \rightarrow (c, vk_m),$$
$$\mathcal{HS}.\mathsf{Transform}(pp, c, (\mathbb{S}, \omega), id, tk_{id}, (\psi, \mathbb{O})) \rightarrow c',$$

we have $\mathcal{HS}.\mathsf{Dec}(pp, id, sk_{id}, c', vk_m) = m$ with probability 1 for data sharing and $\mathcal{SA}.\mathsf{Dec}(pp, id, sk_{id}, \tilde{c}', vk_m) = \perp$ for any $c' \neq \tilde{c}'$ with probability 1 for data verification.

Remark. Healthcare IoT system as shown in *Definition 7* is same as the proposed *SA-DP-ABE* as *Definition 2* except we consider the verification key to check the data integrity from the edge server and key encapsulation mechanism for real-world applications in healthcare IoT system.

3.3 Threat Model

We assume the *KGC* and *DOs* are fully trusted entities. *KGC* generates the system parameter and distributes the valid transformation keys to the *ES* for all *DUs*. *DOs* encrypt the messages based on pre-defined algorithms and then upload corresponding ciphertexts to the *CS*.

CS and *ES* are both honest but curious. They follow our protocols and algorithms defined in healthcare IoT system but try to learn sensitive information without authority. The collusion attacks of them are also allowed. Besides, *CS* may modify the original outsourced data from *DOs* to reduce costs or fool *DUs*.

DUs are untrusted who try to decrypt unauthorized ciphertexts, and even collusion attacks with other entities (i.e., other unauthorized *DUs*, *ES* and *CS*) to reveal unauthorized data.

To ensure the security of our system against the above attacks, we present two security models. One is sIND-CPA game to the unauthorized information. This model includes all possible attacks in our threat model, and it prevents any probabilistic polynomial-time adversary to learn any sensitive data without permission. The other one is verifiable one-wayness secure game against chosen plaintext attack, or simply vOW-CPA, for data integrity, which detects the dishonest *CS* discarding or replacing the original data from *DOs*.

3.4 Security Models

Definition 8 (sIND-CPA in $\mathcal{HS}$). *An $\mathcal{HS}$ consists of six algorithms given above. For an adversary $\mathcal{A}$, we give experiment:*

Experiment $\mathbf{Exp}^{\mathsf{sIND\text{-}CPA}}_{\mathcal{HS},\mathcal{A}}(1^\lambda)$
$(\mathbb{S}^*, \omega^*) \leftarrow \mathcal{A}(1^\lambda)$; $(pp, msk) \leftarrow \mathcal{HS}.\mathsf{Setup}(1^\lambda)$;
$(m_0, m_1) \leftarrow \mathcal{A}^{\mathcal{O}}(pp)$; $b \leftarrow \{0, 1\}$;
$(c^*, vk_{m_b}) \leftarrow \mathcal{HS}.\mathsf{Enc}(pp, m_b, (\mathbb{S}^*, \omega^*))$;
$b' \leftarrow \mathcal{A}^{\mathcal{O}}(c^*, vk_{m_b})$;
If $b = b'$ *return* 1 *else return* 0.

$\mathcal{O}$ denotes a set of oracles $\{\mathcal{O}^{\mathcal{HS}}_{\mathsf{create}}(\cdot,\cdot), \mathcal{O}^{\mathcal{HS}}_{\mathsf{corrupt}}(\cdot)\}$, and details are given below:

- *$\mathcal{O}^{\mathcal{HS}}_{\mathsf{create}}(\cdot,\cdot)$ is a creating oracle that allows $\mathcal{A}$ to query on any set of subjective attribute $\psi \in \Omega_s$ and any objective access structure $\mathbb{O} \in \mathcal{P}_o$. It picks a random identity $id \in \mathcal{I}$, then runs $\mathcal{HS}.\mathsf{KeyGen}(pp, id)$ to obtain the corresponding secret and public key pair (sk_{id}, pk_{id}) and runs $\mathcal{HS}.\mathsf{TKGen}(pp, msk, id, pk_{id}, (\psi, \mathbb{O}))$ to obtain the transformation key tk_{id}. Finally, it returns the transformation key tk_{id} to $\mathcal{A}$.*
- *$\mathcal{O}^{\mathcal{HS}}_{\mathsf{corrupt}}(\cdot)$ is a corrupt oracle that allows $\mathcal{A}$ to query on any identity $id \in \mathcal{I}$, and it returns the corresponding*

Let $\mathcal{SA}$ be a *SA-DP-ABE* scheme, $\mathcal{SE}$ denote a symmetric encryption scheme and $\mathcal{H}$ represent the universe of extractors, respectively. The generic construction of healthcare IoT system with lightweight decryption and data verification $\mathcal{HS}$ are described as follows:

- $\mathcal{HS}.\mathsf{Setup}(1^\lambda) \rightarrow (pp, msk)$: The setup algorithm runs $\mathcal{SA}.\mathsf{Setup}(1^\lambda) \rightarrow (pp_{\mathcal{SA}}, msk_{\mathcal{SA}})$ to obtain the pubic parameters $pp_{\mathcal{SA}}$ and the master secret key $msk_{\mathcal{SA}}$ of the outsourced *SA-DP-ABE*. The algorithm then picks an extractor h from the universe of extractors $\mathcal{H}$ *s.t.* $h : \mathcal{M} \rightarrow \mathcal{K}$, and two collision-resistant hash functions $\mathsf{H} : \mathcal{M}_{\mathcal{SA}} \rightarrow \mathcal{T}$ and $\mathsf{H}' : \mathcal{T} \times \mathcal{C}_{\mathcal{SE}} \rightarrow \mathcal{T}$, where $\mathcal{T}$ is the universe of tags and $\mathcal{C}_{SE}$ is the ciphertext space in symmetric encryption. Finally, it outputs public parameters $pp = (pp_{\mathcal{SA}}, \mathsf{H}, \mathsf{H}', h)$ and the master secret key $msk = msk_{\mathcal{SA}}$.
- $\mathcal{HS}.\mathsf{KeyGen}(pp, id) \rightarrow (pk_{id}, sk_{id})$: The key generation algorithm runs $\mathcal{SA}.\mathsf{KeyGen}(pp_{\mathcal{SA}}, id) \rightarrow (sk_{id}, pk_{id})$ to return the key pairs (sk_{id}, pk_{id}).
- $\mathcal{HS}.\mathsf{TKGen}(pp, msk, id, pk_{id}, (\psi, \mathbb{O})) \rightarrow tk_{id}$: The transformation key generation algorithm returns the transformation key tk_{id} by running $\mathcal{SA}.\mathsf{TKGen}(pp_{\mathcal{SA}}, msk_{\mathcal{SA}}, id, pk_{id}, (\psi, \mathbb{O})) \rightarrow tk_{id}$.
- $\mathcal{HS}.\mathsf{Enc}(pp, m, (\mathbb{S}, \omega)) \rightarrow (c, vk_m)$: The encryption algorithm chooses a random key $K \in \mathcal{M}_{SA}$ and encrypts the key K by running $\mathcal{SA}.\mathsf{Enc}(pp_{\mathcal{SA}}, K, (\mathbb{S}, \omega)) \rightarrow c_{\mathcal{SA}}$. Next, the algorithm extracts the symmetric key $K_{\mathcal{AE}}$ via the extractor h *s.t.* $h(K) \rightarrow K_{\mathcal{SE}}$ and encrypts files to obtain the ciphertext $c_{\mathcal{SE}}$ by running $\mathcal{SE}.\mathsf{Enc}(K_{\mathcal{SE}}, m) \rightarrow c_{\mathcal{SE}}$. To protect the integrity of the outsourced data, the algorithm generates verification key vk_m based on the key K and ciphertext $c_{\mathcal{SE}}$ as $\mathsf{H}'(\mathsf{H}(K), c_{\mathcal{SE}}) \rightarrow \mathsf{Tag}$. Finally, it outputs the ciphertext $c = (c_{\mathcal{SA}}, c_{\mathcal{SE}})$ and the verification key $vk_m = \mathsf{Tag}$.
- $\mathcal{HS}.\mathsf{Transform}(pp, c, (\mathbb{S}, \omega), id, tk_{id}, (\psi, \mathbb{O})) \rightarrow c'/\perp$: The transformation algorithm generates the partially decrypted ciphertext $c'_{\mathcal{SA}}$ by running $\mathcal{SA}.\mathsf{Transform}(pp_{\mathcal{SA}}, c_{\mathcal{SA}}, (\mathbb{S}, \omega), id, tk_{id}, (\psi, \mathbb{O})) \rightarrow c'_{\mathcal{SA}}/\perp$. The algorithm returns ciphertexts $c' = (c_{\mathcal{SE}}, c'_{\mathcal{AE}})$ if the underlying transformation algorithm returns the valid ciphertexts $c'_{\mathcal{SA}}$, otherwise, returns the failure symbol $\perp$.
- $\mathcal{HS}.\mathsf{Dec}(pp, id, sk_{id}, c', vk_m) \rightarrow m/\perp$: The decryption algorithm runs $\mathcal{SA}.\mathsf{Dec}(pp_{\mathcal{SA}}, id, sk_{id}, c'_{\mathcal{SA}}) \rightarrow K$ to recovers the random key $K \in \mathcal{M}_{\mathcal{SA}}$. Then, it computes $\mathsf{H}(K) \rightarrow \mathsf{Tag}$. If $\mathsf{H}'(\mathsf{Tag}, c_{\mathcal{SE}}) \neq vk_m$, the algorithm returns $\perp$, otherwise, computes the key $K_{\mathcal{SK}} = h(K)$ and reveals files $m \in \mathcal{M}_{\mathcal{SE}}$ by running $\mathcal{SE}.\mathsf{Dec}(K_{\mathcal{SE}}, c_{\mathcal{SE}}) \rightarrow m$.

Fig. 4. Generic construction of healthcare IoT system.

secret key sk_{id} if it has been derived in the creating oracle. If no such key exists, then it returns the error symbol $\perp$.

$\mathcal{A}$ is allowed to issue above oracles with the restriction such that for each corrupted users, the corresponding set of subjective attribute $\psi \in \Omega_s$ and objective access structure $\mathbb{O} \in \mathcal{P}_o$ cannot satisfy the condition of the challenge subjective access structure $\mathbb{S}^*$ and set of objective attribute ω^* simultaneously.

The $\mathcal{HS}$ is said to be sIND-CPA *secure if for any probabilistic polynomial-time adversary $\mathcal{A}$, the advantage is negligible:*

$$\mathbf{Adv}^{\mathsf{sIND\text{-}CPA}}_{\mathcal{HS},\mathcal{A}}(1^\lambda) = \left| \Pr[\mathbf{Exp}^{\mathsf{sIND\text{-}CPA}}_{\mathcal{HS},\mathcal{A}}(1^\lambda) = 1] - 1/2 \right|.$$

Definition 9 (vOW-CPA in $\mathcal{HS}$). *An $\mathcal{HS}$ consists of six algorithms given above. For an adversary $\mathcal{A}$, we define the following experiment:*

Experiment $\mathbf{Exp}^{\mathsf{vOW\text{-}CPA}}_{\mathcal{HS},\mathcal{A}}(1^\lambda)$

$(pp, msk) \leftarrow \mathcal{HS}.\mathsf{Setup}(1^\lambda); (m^*, \mathbb{S}^*, \omega^*) \leftarrow \mathcal{A}^{\mathcal{O}}(pp);$

$(c^*, vk_{m^*}) \leftarrow \mathcal{HS}.\mathsf{Enc}(pp, m_b, (\mathbb{S}^*, \omega^*));$

$(c', id) \leftarrow \mathcal{A}^{\mathcal{O}}(c^*, vk_{m^*});$

If $\mathcal{HS}.\mathsf{Dec}(pp, id, sk_{id}, c', vk'_{m^*}) \notin \{m^*, \perp\} return 1$

else return 0.

$\mathcal{O}$ denotes a set of oracles $\{\mathcal{O}^{\mathcal{HS}}_{\mathsf{create}}(\cdot,\cdot), \mathcal{O}^{\mathcal{HS}}_{\mathsf{corrupt}}(\cdot)\}$, and restrictions of these oracle are identical to the settings in the sIND-CPA *model.*

The $\mathcal{HS}$ is said to be sIND-CPA *secure if for any probabilistic polynomial-time adversary $\mathcal{A}$, the following advantage is negligible:*

$$\mathbf{Adv}^{\mathsf{vOW\text{-}CPA}}_{\mathcal{HS},\mathcal{A}}(1^\lambda) = \left| \Pr[\mathbf{Exp}^{\mathsf{vOW\text{-}CPA}}_{\mathcal{HS},\mathcal{A}}(1^\lambda) = 1] \right|.$$

4 Proposed Healthcare IoT System

In this section, we introduce the workflow of healthcare IoT system architecture as shown in Fig. 3 based on the generic construction of the underlying scheme as shown in Fig. 4, and the formal security proofs of this generic construction.

4.1 Workflow of Healthcare IoT System

The workflow can be categorized into two phases: *key distribution* and *data sharing*. Let $\mathcal{HS}$ be the generic construction of healthcare IoT system as shown in Fig. 4, the details of the *key distribution* phase are given below:

1) *KGC* runs the setup algorithm $\mathcal{HS}.\mathsf{Setup}(1^\lambda)$ to obtain public parameters pp and the master secret key msk, and broadcast public parameter pp to *DOs*, *DUs* and *ES* via the public channel or publishes these parameters in the public platform allowing other entities to access (See ❶).
2) After receiving the public parameter from *KGC* (From ❶), each *DU* obtains key pairs (sk_{id}, pk_{id}) by running the key generation algorithm $\mathcal{HS}.\mathsf{KeyGen}(pp, id)$ and sends the public component pk_{id} to *KGC* via the public channel (See ❷).
3) After receiving the public key pk_{id} (From ❷), *KGC* defines a set of subjective attributes $\psi \in \Omega_s$ and an objective access structure $\mathbb{O} \in \mathcal{P}_o$ based on the role of the corresponding *DU* and runs $\mathcal{HS}.\mathsf{TKGen}(pp_{\mathcal{SA}}, msk_{\mathcal{SA}}, id, pk_{id}, (\psi, \mathbb{O}))$ to derive the transformation key tk_{id}. Next, *KGC* sends this transformation key tk_{id} to *ES* through the public channel (See ❸). Notice that transformation keys are distributed publicly since partially decrypted ciphertexts remain secure without the knowledge of secret keys about corresponding *DUs*.

The *data sharing* phase likes the cloud-based cryptosystem except for the majority workload of decryption are moved from *DUs* to *ES*, and the details are described as follows:

4) To encrypt files $m \in \mathcal{M}_{\mathcal{SE}}$, *DOs* define a subjective access structure $\mathbb{S} \in \mathcal{P}_s$ and a set of objective attributes $\omega \in \Omega_o$ for authorized *DUs*, and runs $\mathcal{HS}.\mathsf{Enc}(pp, m, (\mathbb{S}, \omega))$ to derive and send the ciphertext c and the corresponding verification key vk_m to *CS*. Then, it uploads the ciphertext c and the verification key vk_m to *CS* to accommodate ciphertexts (See ❹).
5) When *DUs* intend to reveal messages, *ES* first fetches ciphertexts $c = (c_{\mathcal{SE}}, c_{\mathcal{AE}})$ and the corresponding verification key vk_m from *CS* (See ❺) and produces partially decrypted ciphertexts $c' = (c_{\mathcal{SE}}, c'_{\mathcal{AE}})$ by running $\mathcal{HS}.\mathsf{Transform}(pp, c, (\mathbb{S}, \omega), id, tk_{id}, (\psi, \mathbb{O}))$. Then, it returns the partially decrypted ciphertext c' and the corresponding verification key vk_m to *DUs* (See ❻). If the algorithm returns the failure symbol $\perp$, *ES* rejects current ciphertexts to corresponding *DUs*.
6) After receiving ciphertexts c' and corresponding verification keys vk_m (From ❻), *DUs* reveal files $m \in \mathcal{M}_{\mathcal{SE}}$ by running $\mathcal{HS}.\mathsf{Dec}(pp, id, sk_{id}, c', vk_m)$ or return the failure symbol $\perp$ if the outsourced data is compromised.

4.2 Security Proofs

Theorem 1. *Suppose the underlying* SA-DP-ABE *scheme $\mathcal{SA}$ is* sIND-CPA *secure, $\mathcal{H}$ is a family pairwise independent hash function and symmetric encryption scheme $\mathcal{SE}$ is* SS *secure, then the generic construction $\mathcal{HS}$ in Fig. 4 is* sIND-CPA *secure with the following advantage*

$$\mathbf{A}dv^{\mathsf{sIND\text{-}CPA}}_{\mathcal{HS},\mathcal{A}}(1^\lambda) \leq \mathbf{A}dv^{\mathsf{sIND\text{-}CPA}}_{\mathcal{SA},\mathcal{A}}(1^\lambda) + \epsilon_{\mathcal{H}} + \mathbf{A}dv^{\mathsf{SS}}_{\mathcal{SE},\mathcal{A}}(1^\lambda).$$

The following proof based on sequence of games. Particularly, our proof includes on a serial games from $Game_0$ to $Game_2$. $Game_0$ is the original game, $Game_1$ and $Game_2$ will do some modifications based on the previous game (modifications are indistinguishable from the view of adversary), and the advantage of adversary in $Game_2$ is negligible. It is worth to notice that messages $m_0, m_1 \in \mathcal{M}_{\mathcal{SE}}$ in $\mathcal{HS}$ sIND-CPA game rather $m_0, m_1 \in \mathcal{M}_{\mathcal{AS}}$ in $\mathcal{SA}$ sIND-CPA game, the verification key and the message domain transformation are based on $\mathcal{H}$ and $\mathcal{SE}$. Hence, our $\mathcal{HS}$ system relies on the security of underlying $\mathcal{SA}$, $\mathcal{H}$ and $\mathcal{SE}$. See Appendix A for the details of proofs. Below we describe the details of each game.

- $Game_0$ is the original sIND-CPA game. Specifically, the challenger ciphertext c^* and verification key vk^* are in the forms of $c^* = (c^*_{\mathcal{SA}}, c^*_{\mathcal{SE}})$ and $vk^*_m = \mathsf{Tag}$, and the key $K^*_{\mathcal{SE}} = h(K^*)$ is derived from the randomness extractor $h \in \mathcal{H}$, where $K^* \in \mathcal{M}_{\mathcal{SA}}$.
- $Game_1$ is the same as $Game_0$ except that we use a random message $R^* \in \mathcal{M}_{\mathcal{SA}}$ for symmetric encryption to derive the challenger ciphertext c^* instead of the $K^* \in \mathcal{M}_{\mathcal{SA}}$ in $Game_0$.
- $Game_2$ is the same as $Game_1$ except that we use a random string $R^*_{\mathcal{SE}} \in \mathcal{M}_{\mathcal{AS}}$ instead of $K^*_{\mathcal{SE}}$ from the randomness extractor.

Theorem 2. *Suppose* H *and* H′ *are two collision-resistant hash functions, then the generic construction $\mathcal{HS}$ in Fig. 4 is* vOW-CPA *secure.*

The proof based on security reduction. If an adversary can break our vOW-CPA secure, then we can build an algorithm $\mathcal{B}$ to break collision-resistance of underlying hash functions. It is worth to notice that $\mathcal{B}$ knows everything including the master secret key msk and each user's secret key sk_{id} except for the constructions of two collision-resistance hash functions and the security model vOW-CPA is an adaptive model. The details are given in Appendix B.

5 Proposed SA-DP-ABE Scheme

5.1 Concrete Scheme

The concrete scheme as shown in Fig. 5. which is derived from Rouselakis and Waters' ABE schemes [16] and Green et al.'s outsourced ABE [15].

The high-level idea of our proposed scheme is that *DUs* individually generate the secret and public key pairs based on the ElGamal type key pair such as (β, g^β), where β is the secret key only known by themselves, and g^β is the public key. The *KGC* keeps a master secret key α to issue transformation keys based on users' public keys g^β. Specifically, for each *DU*, the *KGC* randomly picks $\alpha_{id} \in \mathbb{Z}_p$ and derives the transformation key which includes one KP-ABE secret key to embed the secret information $\alpha_{id}\beta$ and one CP-ABE secret key to maintain the secret information $(\alpha - \alpha_{id})\beta$. Anyone except the corresponding *DU* cannot eliminate β in secret keys.

DOs encrypt data based on a subjective access structure, a set of objective attribute and the public parameter from the *KGC*. Specifically, messages are encrypted in the form of $m \cdot e(g,g)^{\alpha s}$. The *ES* can retrieve $e(g,g)^{\alpha_{id} s \cdot \beta}$ for KP-ABE component and $e(g,g)^{(\alpha - \alpha_{id}) s \cdot \beta}$ for CP-ABE component, then combine them to return $e(g,g)^{\alpha s \cdot \beta}$ for authorized data users. The edge device cannot fully decrypt these ciphertexts since β is unknown to the *ES*. By eliminating β, *DUs* retrieve the message hiding component $e(g,g)^{\alpha s}$ and reveal message m by removing this component.

5.2 Security Analysis

Theorem 3. *If the modified q assumption holds, then all probabilistic polynomial-time adversaries have a negligible advantage in selectively breaking our scheme.*

Our proof is based on the Rouselakis and Waters' ABE schemes [16]. One of the most challenge is to answer the transformation key for data users who have $(\psi, \mathbb{O})$ s.t. $\psi \in \mathbb{S}^*$ and/or $\omega^* \in \mathbb{O}$. For data users who have either $\psi \in \mathbb{S}^*$ or $\omega^* \in \mathbb{O}$, $\mathcal{B}$ randomly chooses $\alpha_{id} \in \mathbb{Z}_p$ to generate the one part of transformation key satisfying the challenge, and simulates the other component with the secret information $\alpha - \alpha_{id}$ normally. For data users who have both $(\psi, \mathbb{O})$ s.t. $\psi \in \mathbb{S}^*$ and/or $\omega^* \in \mathbb{O}$, we chooses a random element $\beta \in \mathbb{Z}_p$ to return tk_{id} by running $\mathsf{TKGen}(pp, \beta, id, pk_{id}, (\psi, \mathbb{O}))$, where tk_{id} has the right distribution of the transformation key and the corresponding secret key is unknown to $\mathcal{B}$. Notice that $\mathcal{A}$ cannot query the secret key for the users who have $(\psi, \mathbb{O})$ s.t. $\psi \in \mathbb{S}^*$ and $\omega^* \in \mathbb{O}$ as the restriction is our model. See appendix E for further details.

Let Ω_s and Ω_o be the subjective attribute universe and the objective universe, $\mathcal{P}_s$ and $\mathcal{P}_o$ denote subjective policies and objective policies, $\mathcal{M}$ represent the message space. The concrete scheme of *SA-DP-ABE* with lightweight decryption $\mathcal{SA}$ are described as follows:

- $\mathcal{SA}.\mathsf{Setup}(1^\lambda)$: The setup algorithm runs the group generator $\mathcal{G}(1^\lambda)$ to generate the description of bilinear group $(\mathbb{G}, \mathbb{G}_T, g, p)$. It then randomly pick terms $w, v, u, h, \tilde{u}, \tilde{h} \in \mathbb{G}$ and $\alpha \in \mathbb{Z}_p$. The algorithm returns the public parameter $pp = (g, w, v, u, h, \tilde{u}, \tilde{h}, e(g,g)^\alpha)$ and the master secret key $msk = \alpha$.
- $\mathcal{SA}.\mathsf{KeyGen}(pp, id)$: The key generation algorithm chooses a random exponent $\beta \in \mathbb{Z}_p$, and returns the secret key $sk_{id} = \beta$ and the corresponding public key $pk_{id} = g^\beta$.
- $\mathcal{SA}.\mathsf{TKGen}(pp, msk, id, pk_{id}, (\psi, \mathbb{O}))$: Parse a set of subjective attributes is $\psi = (\psi_1, \psi_2, ..., \psi_{k_s}) \subseteq \mathcal{U}_s$ and $\mathbb{O} = (M, \rho) \in \mathcal{P}_o$ is an objective access structure, where M is a $\ell_o \times n_o$ matrix and $\rho : [\ell_o] \to \mathbb{Z}_p$. The transformation key generation algorithm picks $\vec{x} = (\alpha_{id}, x_2, ..., x_{n_o})^\top \in \mathbb{Z}_p^{n_o \times 1}$ and computes $\vec{\lambda} = (\lambda_1, \lambda_2, ..., \lambda_{\ell_o}) = M\vec{x}$. It then chooses exponents $r, \{r_i\}_{i\in[\ell_o]}, \{r_\tau\}_{\tau\in[k_s]} \in \mathbb{Z}_p$, then returns the transformation key tk_{id} as:

$$\begin{aligned} tk_{id} &= (\{tk_{1,i}, tk_{2,i}, tk_{3,i}\}_{i\in[\ell_o]}, tk_4, \{tk_{5,\tau}, tk_{6,\tau}\}_{\tau\in[k_s]}), \\ &= (\{pk_{id}^{\alpha-\alpha_{id}+\lambda_i} w^{r_i} w^r, (\tilde{u}^{\rho(i)}\tilde{h})^{-r_i}, g^{r_i}\}_{i\in[\ell_o]}, g^r, \{g^{r_\tau}, (u^{\psi_\tau} h)^{r_\tau} v^{-r}\}_{\tau\in[k_s]}). \end{aligned}$$

- $\mathcal{SA}.\mathsf{Enc}(pp, m, (\mathbb{S}, \omega))$: Parse a subjective access structure $\mathbb{S} = (N, \pi)$ and a set of objective attributes $\omega = (\omega_1, \omega_2, ..., \omega_{k_o})$, where N is a $\ell_s \times n_s$ matrix and $\pi : [\ell_s] \to \mathbb{Z}_p$. The encryption algorithm picks $\vec{y} = (s, y_2, ..., y_{n_s})^\top \in \mathbb{Z}_p^{n_s \times 1}$ and computes $\vec{\Pi} = (\Pi_1, \Pi_2, ..., \Pi_{\ell_s}) = N\vec{y}$. It then chooses $\{s_i\}_{i\in[k_o]}, \{s_\tau\}_{\tau\in[\ell_s]} \in \mathbb{Z}_p$ and returns the ciphertext c as:

$$\begin{aligned} c &= (c_0, c_1, \{c_{2,i}, c_{3,i}\}_{i\in[k_o]}, \{c_{4,\tau}, c_{5,\tau}, c_{6,\tau}\}_{\tau\in[\ell_s]}), \\ &= (m \cdot e(g,g)^{\alpha s}, g^s, \{g^{s_i}, (\tilde{u}^{\omega_i}\tilde{h})^{s_i} w^{-s}\}_{i\in[k_o]}, \{w^{\Pi_\tau} v^{s_\tau}, (u^{\pi(\tau)} h)^{-s_\tau}, g^{s_\tau}\}_{\tau\in[\ell_s]}). \end{aligned}$$

- $\mathcal{SA}.\mathsf{Transform}(pp, c, (\mathbb{S}, \omega), id, tk_{id}, (\psi, \mathbb{O}))$: For the pair $(\psi, \mathbb{O})$, the transformation algorithm calculates the set of rows in M that provide a share to attributes in ψ, i.e. $I = \{i : \rho(i) \in \psi\}$ and computes the constants $\vec{u} = \{u_i \in \mathbb{Z}_p\}$ $s.t.$ $\sum_{i\in I} M_i u_i = (1, 0, ..., 0)$. For the pair $(\mathbb{S}, \omega)$, the algorithm calculates the set of rows in N that provide a share to attributes in ω, i.e. $J = \{j \in \pi(j) \in \omega\}$ and computes the constants $\vec{v} = \{v_i \in \mathbb{Z}_p\}$ $s.t.$ $\sum_{j\in J} N_j v_j = (1, 0, ..., 0)$. The algorithm computes B_1 as:

$$\begin{aligned} B_1 &= \prod_{j\in J} (e(c_{4,j}, tk_4) \cdot e(c_{5,j}, tk_{5,j}) \cdot e(c_{6,j}, tk_{6,j}))^{v_j} \\ &= \prod_{j\in J} (e(w^{\Pi_j} v^{s_j}, g^r) \cdot e((u^{\pi(j)} h)^{-s_j}, g^{r_j}) \cdot e(g^{s_j}, (u^{\omega_j} h)^{r_j} v^{-r}))^{v_j} \\ &= e(g,w)^{rs}. \end{aligned}$$

  For all $i \in I$, the algorithm computes $B_{2,i}$ as:

$$B_{2,i} = e(c_1, tk_{1,i})/B_1 = e(g^s, pk_{id}^{\alpha-\alpha_{id}+M_i x_i} w^{r_i} w^r)/e(g,w)^{rs} = e(g^s, pk_{id}^{\alpha-\alpha_{id}+M_i x_i} w^{r_i}).$$

  The algorithm then calculates c_0' as:

$$\begin{aligned} c_0' &= \prod_{i\in I} \left(B_{2,i} \cdot e(c_{2,i}, tk_{2,i}) \cdot e(c_{3,i}, tk_{3,i})\right)^{u_i} \\ &= \prod_{i\in I} (e(g^s, pk_{id}^{\alpha-\alpha_{id}+\lambda_i} w^{r_i}) \cdot e((\tilde{u}^{\rho(i)}\tilde{h})^{-r_i}, g^{s_i}) \cdot e(g^{r_i}, (\tilde{u}^{\omega_i}\tilde{h})^{s_i} w^{-s}))^{u_i} \\ &= e(g, pk_{id})^{\alpha s}. \end{aligned}$$

- $\mathcal{SA}.\mathsf{Dec}(pp, id, sk_{id}, c')$: The decryption algorithm returns the message m as:

$$c_0/(c_0')^{1/\beta} = m \cdot e(g,g)^{\alpha s}/e(g,g)^{\alpha\beta s/\beta} = m.$$

Fig. 5. Concrete construction of *SA-DP-ABE* scheme.

6 Efficiency Analysis

To our knowledge, in addition to our work in this paper, [7], [17] are also DP-ABE schemes. The comparison of functionalities for above DP-ABE schemes is described in Table 2. Our scheme is based prime-order group since the purpose of this paper is to derive a lightweight scheme to IoT devices, and composite-order group will incur heavy workload to data encryption and decryption, even transmission bandwidth. Our scheme inherits the advantages of edge computing. Data users keep own secret keys, and their secret keys are unknown to the edge server. Hence, the edge server could be untrusted, it cannot learn any sensitive data without knowing the corresponding users' secret keys. Besides, the users' secret keys are unknown to KGC, and the transformation keys generated by KGC are published publicly. Hence, our scheme need not initialize any secure channel.

To demonstrate the high performance of our proposed scheme, in the rest of this section, we focus on efficiency analysis for theoretical complexity and experimental simulation.

6.1 Theoretical Complexity

Table 3 focus on the cost of data users and data owners, then compares the theoretical complexity among [7], [17] and

TABLE 2
Functionality Summary of DP-ABE Schemes

	Functionality					
	Type of ABE	Edge Server	Order of Group	Security	Secure Channel	Data Verification
AI09 [7]	*DP-ABE*	*No*	*Prime*	*Selective*	*Yes*	*No*
AY15 [17]	*DP-ABE*	*No*	*Composite*	*Adaptive*	*Yes*	*No*
Ours	*DP-ABE*	*Untrusted*	*Prime*	*Selective*	*No*	*Yes*

ours, including space complexity and computational complexity. The theoretical analysis shows that our scheme is comparable to DP-ABE schemes [7], [17].

For space complexity, our scheme requires KGC to generate constant-size public parameter which inherits from the Rouselakis and Waters' ABE [16], and constant-size secret key (with one $\mathbb{Z}_p$ element). It also requires cloud server to store permanently the encrypted ciphertext, which size is based on the ABE cryptosystem and the corresponding plaintext, and requires edge server to store temporarily the partially decrypted ciphertext (with its size equal to two $\mathbb{G}_T$ elements) by applying outsourced ABE [15].

For computational complexity, all schemes have the same cost of encryption since the data owners encrypt ciphertexts without any support from either the remote cloud server and the edge server. In our scheme, data users only demand one multiplication in $\mathbb{G}_T$ group to reveal messages from the partially decrypted ciphertexts. The other two schemes require the data owners to decrypt ciphertext based on the corresponding access policy and attributes. We do not consider the cost of the edge server in Table 3 since the purpose of this paper is to design an efficient, flexible and secure mechanism for IoT devices and the edge service helps IoT devices to reduce the cost. We will give the time and storage consumption in the next section.

6.2 Experimental Simulation

For experimental simulation, we focus on evaluating AI09 [7], and our scheme because AY15 [17] is in the composite-order group, and the composite-order is inefficient [34]. In general, the composite-order group has a much bigger size than the prime-order group. Specifically, the composite-order group needs 1024 bits if the prime-order group requires 160 bits (discrete log versus factoring). Our experiential simulation was performed on a PC running 64-bit Windows 10 with 3.60 GHz Intel(R) Core(TM) i7-4790 CPU and 24 GB memory. In particular, we have implemented AI09 [7] and our scheme in Java using the JPBE library with Type A elliptic curve and the symmetric pairing setting from "a.properties" provided by JPBE library. Hence, in our scheme, p is a 160-bit prime number, and elements in $\mathbb{G}$ and $\mathbb{G}_T$ have 512 bits and 1024 bits, respectively. The experimental performances are presented in Fig. 6.

Figs. 6a and 6b present the experimental performances of the initializing system by increasing the maximum number of attribute set allowed to be assigned to a key and a ciphertext. In Fig. 6a, our scheme only remains the constant-size system parameters rather than the linear to the maximum number of attribute set in AI09. In Fig. 6a, our system only requires about 20 ms to initialize the system and keeps stable even the maximum number of attribute set increasing. The results are similar to what we expected performances in Table 3.

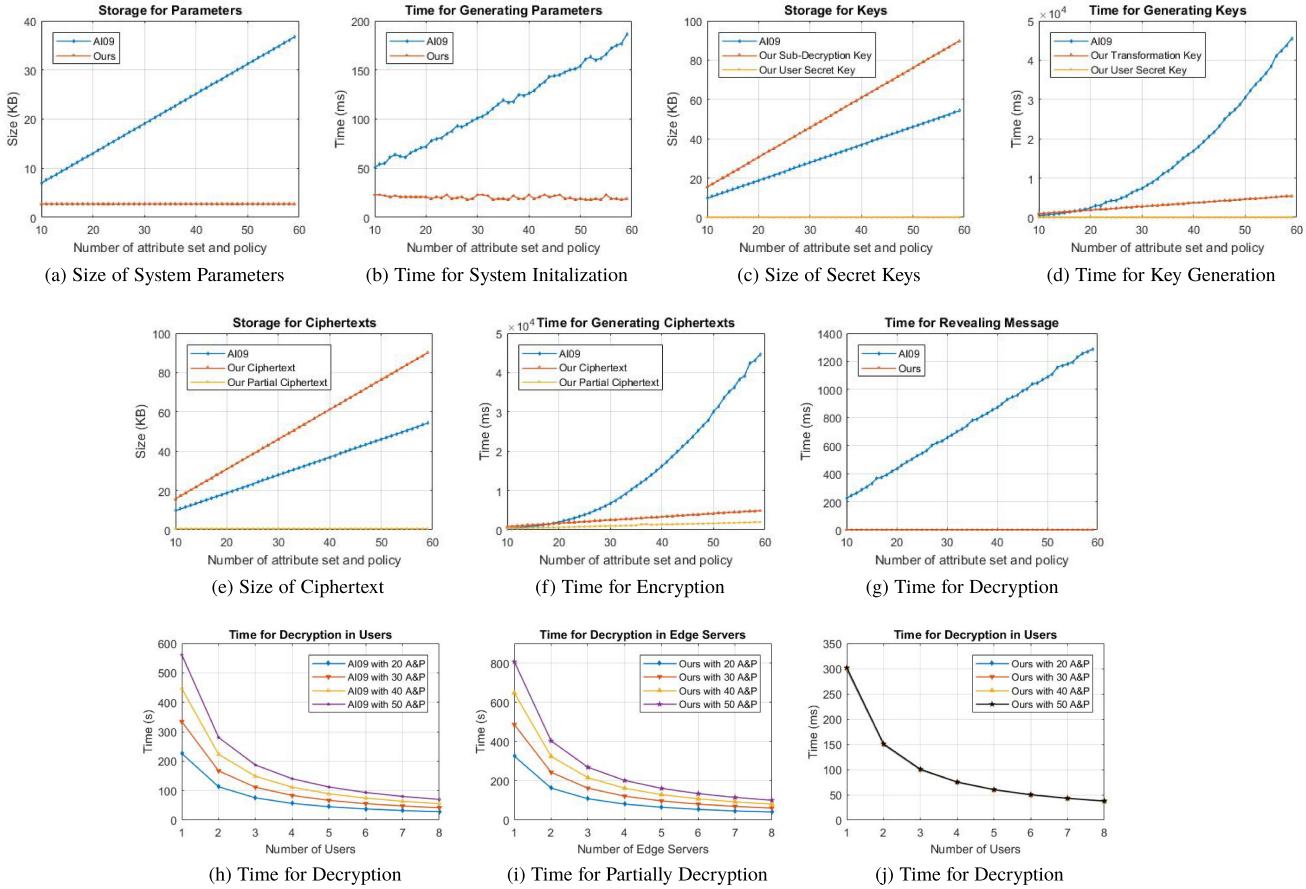
Figs. 6c and 6d give the performances of key generation based on the number of attribute set and policies assigned to a key. In AI09, we focus on KGC generating secret keys to data. In our proposed scheme, we consider two cases. One is the secret user key, each data user generates the own secret and public key pair and broadcasts the corresponding public key to KGC. The other one is the transformation key, KGC generates the corresponding transformation key based on the data user's public key and sends the transformation key to the edge server. From the point of data users, they need to keep the secret key (as the blue line) in AI09 and the user secret key (as the yellow line) in ours. Hence, from the data user side, our scheme has much less cost of storage and computation about user secret key than AI09.

Figs. 6e and 6f present the performances of encryption based on the different number of attribute set and policy assigned to a ciphertext (refers to key attributes). AI09 only has ciphertext which is generated by the data owner. In our scheme, we consider the ciphertext which is also generated by the data owner and the partially decrypted ciphertext which is derived from the edge server. Although our scheme has larger ciphertexts than AI09, these ciphertexts are stored in the remote cloud server, and we assume the cloud server has a vast amount of storage to accommodate ciphertexts.

TABLE 3
Theoretical Analysis of DP-ABE Scheme

	Space Complexity			Computational Complexity	
	Public Parameter	Secret Key	Ciphertext	Encryption	Decryption
AI09 [7]	$\mathcal{O}(max_s + max_o)$	$\mathcal{O}(\psi + \mathbb{O})$	$\mathcal{O}(\mathbb{S} + \omega)$	$\mathcal{O}(\mathbb{S} + \omega)$	$\mathcal{O}(\psi + \mathbb{O})$
AY15 [17]	$\mathcal{O}(1)$	$\mathcal{O}(\psi + \mathbb{O})$	$\mathcal{O}(\mathbb{S} + \omega)$	$\mathcal{O}(\mathbb{S} + \omega)$	$\mathcal{O}(\psi + \mathbb{O})$
Ours	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(\mathbb{S} + \omega)$	$\mathcal{O}(1)$

max_s denotes the maximum size of subjective attribute set allowed to be assigned to a key;
max_o is the maximum size of objective attribute set allowed to be associated with a ciphertext;
ψ and $\mathbb{O}$ represent the size of subjective attributes and objective access structure assigned to a key;
$\mathbb{S}$ and ω are the size of subjective access structure and objective attributes assigned to a ciphertext.



(a) Size of System Parameters (b) Time for System Initalization (c) Size of Secret Keys (d) Time for Key Generation

(e) Size of Ciphertext (f) Time for Encryption (g) Time for Decryption

(h) Time for Decryption (i) Time for Partially Decryption (j) Time for Decryption

Fig. 6. Experimental performances.

From the point of data owners, our scheme takes less computational cost than AI09. From the point of data users, the storage cost is stable ($2\mathbb{G}_T$ elements) in our scheme rather than linear to key attributes in AI09. Hence, our scheme has much less cost of storage and computation to handle of ciphertexts.

In Fig. 6g, we consider data users to decrypt ciphertexts based on the number of attribute set and policies assigned to a ciphertext. Our scheme only takes the constant-size computational time since the ciphertext has been pre-processed by the edge server rather than data users individually in AI09.

Figs. 6h, 6i, and 6j demonstrate the scalability based on the number of users or edge servers. Specifically, we apply multithreading to decrypt 500 ciphertexts with different number of A&P (is short for attribute set and policy). Although the partial decryption in the edge server takes more time than AI09 decryption, the decryption cost in user side of our scheme is much faster (about 1900× to 1000×) than AI09 since no matter the number of A&P our scheme only requires one pairing calculation. Besides, we consider edge servers have much more powerful CPU than our PC, and we can combine multiple devices to build edge servers for the real-world application.

The above figures show that the experimental outcomes are similar to what we expected in Table 3. In general, our scheme has the constant cost in terms of system initialization, key generation, data encryption and decryption in end-devices. Compared to the existing solution, our system requires only 1/10 storage space and very little computing time in the large-scale system (number of attribute set and policy is 60). Hence, the proposed construction has compared performance, and the data user has much less workload than the existing DP-ABE schemes. Therefore, our scheme is suitable for the healthcare IoT system, especially for lightweight end-devices.

Note that the above experimental simulations from Figs. 6a, 6b, 6c, 6d, 6e, 6f, and 6g are processed in the CPU limited in a single core to simulate the environment of healthcare IoT system. In healthcare IoT system, we consider the router and PC (usually with the comparable computational resource to our experimental devices) as encryptors, and lightweight devices such that smartphone and tablet (with comparable computational power e.g., Apple A12 with six-core CPU and 2.49 GHz) as decryptors. Since the computational consumption in our experimental simulation is in milliseconds, we consider our proposed scheme suitable for the IoT ecosystem in the present and foreseeable future.

The scalable key distribution as shown in Table 4, we compare key distribution overhead among AI09, AY15 and ours. In AI09 and AY15, the secret key must be distributed via the secret channel. To simulate this environment, we apply Diffie-Hellman key exchange with 2048-bit large prime number to form the secure channel, which has been widely used in secure sockets layer and transport layer security and Diffie-Hellman key exchange with 1024-bit has been proved unreliable. Diffie-Hellman key exchange requires at least 3-round (2 rounds for key exchange between the server and the data user and 1 round for key distribution from the server to the data user), the server and the data user must keep the

TABLE 4
Performances of Scalable Key Distribution

	Server				User			
	Time Overhead	Bandwidth	Storage Overhead	Round	Time Overhead	Bandwidth	Storage Overhead	Round
AI09 [7]	2.076s	$\mathcal{O}(\psi+\mathbb{O})$	≥ 4096 bits	2	0.684 s	$\mathcal{O}(\psi+\mathbb{O})$	≥ 4096 bits	1
AY15 [17]	2.076s	$\mathcal{O}(\psi+\mathbb{O})$	≥ 4096 bits	2	0.684 s	$\mathcal{O}(\psi+\mathbb{O})$	≥ 4096 bits	1
Ours	-	$\mathcal{O}(\psi+\mathbb{O})$	-	1	-	$\mathbb{O}(1)$	-	1

ephemeral keys they have chosen. Our proposed scheme does not require the server and the data user to record any ephemeral information. Therefore, our scheme has no time overhead and storage overhead in the key distribution phase. Note that the server may distribute keys to multiple data users simultaneously, overhead costs affect the performance of key distribution.

The proposed scheme is a hybrid cryptosystem, such that *SA-DP-ABE* is used for encrypting a symmetric key $K \in \mathcal{K}$, and symmetric-key encryption is used for encrypting a data file in binary format with the key K. Hence, our healthcare IoT system enables the encryptor to encode any health document such as X-Ray pictures and MRI scan files into a binary string $m \in \{0,1\}^*$, encrypt m under the key K using symmetric-key encryption and the key K under the *SA-DP-ABE* scheme. The encryptor outsources the ciphertexts to the cloud for fine-grained data sharing. Therefore, our proposed scheme is suitable for protecting the healthcare system.

7 Conclusion and Future Work

In this work, we investigated the problems of the IoT network in the healthcare application scenario and produced a system to realize secure IoT communication with source-limited devices and fine-grained access control. The system architecture, threat model and security definition were presented for the proposed system. We also provided the proof of our proposed scheme and presented experimental analysis to demonstrate our proposed construction has significantly better performance than previous solutions. The core technique to build secure healthcare IoT system is SA-DP-ABE, which is a promising tool to provide lightweight and expressive fine-grained access control. In the future, investigating more scenarios based on SA-DP-ABE and discover more applications could be interesting future works.

Appendix A
Security Proof for Confidentiality

Claim 1. *Suppose the underlying $\mathcal{SA}$ is* sIND-CPA *secure, then the adversary's view in $Game_0$ and $Game_1$ are computationally indistinguishable.*

Proof. We can build an algorithm $\mathcal{B}$ to break the sIND-CPA of the underlying $\mathcal{SA}$ scheme $\mathcal{C}_{\mathcal{SA}}$ under the help of $\mathcal{A}$. $\mathcal{B}$ simulates $Game_0$ or $Game_1$ based on the underlying $\mathcal{SA}$ scheme.

Init. $\mathcal{B}$ receives challenge information $(\mathbb{S}^*, \omega^*)$ from $\mathcal{A}$ and forwards them to $\mathcal{C}_{\mathcal{SA}}$.

Setup. $\mathcal{C}_{\mathcal{SA}}$ returns the public parameter $pp_{\mathcal{SA}}$ to $\mathcal{B}$. $\mathcal{B}$ chooses two collision-resistant hash function H and H', a random extractor $h \in \mathcal{H}$ and a SS secure one-time encryption scheme $\mathcal{SE}$. $\mathcal{B}$ then sends public parameters $pp = (pp_{\mathcal{SA}}, \mathsf{H}, \mathsf{H}', h)$ to $\mathcal{A}$.

Query Phase 1 and 2. $\mathcal{A}$ adaptively queries the creating oracle $\mathcal{O}^{\mathcal{HS}}_{\mathsf{create}}(\psi, \mathbb{O})$ and the corrupt oracle $\mathcal{O}^{\mathcal{HS}}_{\mathsf{corrupt}}(id)$. $\mathcal{B}$ responses them directly based on oracles in $\mathcal{C}_{\mathcal{SA}}$ such as $\mathcal{O}^{\mathcal{SA}}_{\mathsf{create}}(\psi, \mathbb{O})$ and $\mathcal{O}^{\mathcal{SA}}_{\mathsf{corrupt}}(id)$.

Challenge. $\mathcal{A}$ submits messages m_0 and m_1 with the same length to $\mathcal{B}$. $\mathcal{B}$ chooses two independent random keys $K^*, R^* \in \mathcal{M}_{\mathcal{SA}}$ and sends them to $\mathcal{C}_{\mathcal{SA}}$. $\mathcal{C}_{\mathcal{SA}}$ returns the challenge ciphertext $c^*_{\mathcal{SA}}$ to $\mathcal{B}$. Next, $\mathcal{B}$ chooses a bit $b \in \{0,1\}$, sets $K^*_{\mathcal{SE}} = h(R^*)$ and computes the ciphertext for symmetric encryption $c_{\mathcal{SE}} = (K^*_{\mathcal{SE}}, m_b)$ and verification key $vk_{m_b} = \mathsf{H}'(\mathsf{H}(R^*), c_{\mathcal{SE}})$. $\mathcal{B}$ then forwards $c^* = (c^*_{\mathcal{SA}}, c_{\mathcal{SE}})$ and vk_{m_b} to $\mathcal{A}$.

Guess. $\mathcal{A}$ submit a bit b' as the guessing of b to $\mathcal{B}$ and $\mathcal{B}$ forward this bit to $\mathcal{C}_{\mathcal{SA}}$.

If $c^*_{\mathcal{SA}}$ is derived from R^* then $\mathcal{B}$ perfectly simulates $Game_0$, otherwise, simulates $Game_1$. Hence, $Game_0$ and $Gane_1$ is computationally indistinguishable depending on the underlying $\mathcal{SA}$ scheme. Therefore, we have the following result:

$$\left|\Pr[b = b' | Game_0] - \Pr[b = b' | Game_1]\right| \leq \mathbf{Adv}^{\mathsf{sIND\text{-}CPA}}_{\mathcal{SA},\mathcal{A}}(1^\lambda).$$

□

Claim 2. *Suppose the underlying $\mathcal{H}$ is a family of pairwise independent hash functions, then the adversary's view in $Game_1$ and $Game_2$ are statistically indistinguishable.*

Proof. In $Game_1$, the key $K^*_{\mathcal{SE}} = h(R^*)$ is derived from the random message R^* through randomness extractor $h \in \mathcal{H}$. In $Game_2$, the key $R^*_{\mathcal{SE}}$ is a random message. By Lemma 1, the key $K^*_{\mathcal{SE}}$ from the randomness extractor in $Game_1$ and a truly random key $R^*_{\mathcal{SE}}$ in $Game_2$ are $\epsilon_{\mathcal{H}}$-statistically indistinguishable from the adversary's point of view. Therefore, we have the following result:

$$\left|\Pr[b = b' | Game_1] - \Pr[b = b' | Game_2]\right| \leq \epsilon_{\mathcal{H}}.$$

□

Claim 3. *Suppose the underlying $\mathcal{SE}$ is* SS *secure, then the adversary in $Game_2$ has a negligible advantage.*

Proof. In $Game_2$, the key $R^*_{\mathcal{SE}}$ is a truly random message. Hence, we can directly construct the algorithm $\mathcal{B}$ from $\mathcal{A}$ to break the SS secure $\mathcal{SE}$. Therefore, we have the following result:

$$\left|\Pr[b = b' | Game_2] - 1/2\right| = \mathbf{Adv}^{\mathsf{SS}}_{\mathcal{SE},\mathcal{A}}(1^\lambda).$$

□

APPENDIX B
SECURITY PROOFS FOR VERIFICATION

Proof. We can build an algorithm $\mathcal{B}$ to break collision-resistance of H and H'.

Setup. $\mathcal{B}$ receives two challenge hash functions $(\mathsf{H}, \mathsf{H}')$ and then runs $\mathcal{HS}.\mathsf{Setup}(1^\lambda)$ to generates the public parameter pp and the master secret key msk, except for hash functions H and H' are from the challenger. $\mathcal{B}$ returns the public parameter pp to $\mathcal{A}$.

Query Phase 1 and 2. $\mathcal{A}$ adaptively queries the creating oracle $\mathcal{O}^{\mathcal{HS}}_{\mathsf{create}}(\psi, \mathbb{O})$ and the corrupt oracle $\mathcal{O}^{\mathcal{HS}}_{\mathsf{corrupt}}(id)$. $\mathcal{B}$ responses them by running the corresponding algorithms since the master secret key msk is known by $\mathcal{B}$.

Challenge. $\mathcal{B}$ receives a challenge message m^* and corresponding challenge information $(\mathbb{S}^*, \omega^*)$. $\mathcal{B}$ generates the challenge ciphertext by running $\mathcal{HS}.\mathsf{Enc}(pp, m^*, (\mathbb{S}^*, \omega^*)) \rightarrow (c^*, vk_{m^*})$, where the verification key $vk_{m^*} = \mathsf{H}'(\mathsf{H}(K^*), c_{\mathcal{SE}})$. Finally, $\mathcal{B}$ returns $c^* = (c_{\mathcal{SA}}, c_{\mathcal{SE}})$ vk_{m^*} is derived from the challenge hash functions $(\mathsf{H}, \mathsf{H}')$.

Guess. $\mathcal{A}$ outputs the partially decrypted ciphertext c' and the identity id. $\mathcal{B}$ retries the secret key β_{id} and recovers the message $m \notin \{m^*, \perp\}$ via $\mathcal{HS}.\mathsf{Dec}(pp, id, sk_{id}, c', vk_m)$. Let vk_m be the verification key of the message m and $c_{\mathcal{SE}}$ be the ciphertext about symmetric component from the returning message c', we then consider following two cases:

- If $(\mathsf{H}(K), c_{\mathcal{SE}}) \neq (\mathsf{H}(K^*), c^*_{\mathcal{SE}})$, $\mathcal{B}$ obtains the collision of the hash function H' because the valid ciphertext c' from $\mathcal{A}$. Specifically, the ciphertext c' passing the verification means we have $vk_{m^*} = vk_m \ \ s.t. \ \ \mathsf{H}'(\mathsf{H}(K), c_{\mathcal{SE}}) = \mathsf{H}'(\mathsf{H}(K^*), c^*_{\mathcal{SE}})$.
- If $(\mathsf{H}(K), c_{\mathcal{SE}}) = (\mathsf{H}(K^*), c^*_{\mathcal{SE}})$, $\mathcal{B}$ obtains the collision of the hash function H since $\mathsf{H}(K) = \mathsf{H}(K^*)$.

Therefore, if $\mathcal{A}$ can break our vOW-CPA secure, then we can break collision-resistance of underlying hash functions H, H'. □

Data Integrity. The verification of outsourced data is operated by cooperating the encryption algorithm and the decryption algorithm. Specifically, the encryption algorithm generates the ciphertext c and the corresponding verification key vk_m, and vk_m is a hash result of symmetric key $K \in \mathcal{K}$ and the corresponding ciphertext $c_{\mathcal{SE}}$. The decryption algorithm recovers the symmetric key $K \in \mathcal{K}$ to generates the hash value based on K and the ciphertext $c_{\mathcal{SE}}$, and compares this hash value to the verification key vk_m to enable the data user to verify the outsourced data integrity.

APPENDIX C
ROUSELAKIS AND WATERS' ABE [16]

Rouselakis and Waters [16] proposed practical KP-ABE and CP-ABE schemes proven secure under decisional q-1 and q-2 assumptions. Our concrete scheme based on these two ABE schemes and the security of our proposed scheme can be reduced to them. Let $\mathcal{KP}$ denote KP-ABE scheme and $\mathcal{CP}$ represent CP-ABE scheme. We omit the definition and secure model of above scheme and reader may refer to [16] for details.

APPENDIX D
SECURITY PROOF FOR MODIFIED q ASSUMPTION

Before giving the formal proof, we recall the corollary in [33] as follows:

Corollary 1. *If $\tilde{A}^1 = (0, 0, \ldots, 0, 1) \in \mathbb{Z}_p^{1\times K}$ and $\langle \tilde{A}^1, A^i \rangle$ for all $i \in [L]$, the corresponding $\mathbb{G}_T$-monomial assumption is secure in the generic group model if and only if for all $i, j \in [L]$ it is true that $\tilde{A}^0 \neq A^i + A^j$.*

Lemma 2. *The modified q assumption is secure in the generic group model.*

Proof. Recall the notations $[i : x]$ and $[i : x, i' : y]$ in [33]. Let $[i : x]$ and $[i : x, i' : y]$ denote the row vectors in $\mathbb{Z}_p^{1\times q}$ will all components equal to 0 except the ith component for the first vector and the i, i'th component for the second. The non zero elements are x for the first vector and x, y for the i, i'th possible, respectively, of the second vector. The Table 5 shows a compact from of the matrix A where rows of similar type are shown in one line.

In order to prove the lemma we have to show that by adding any two rows of matrix A we cannot get the row vector $\tilde{A}^0 = (q + 1, 1, 0, 0, \ldots, 0)$. By inspecting Table 5, we only have to check the rows of types $2, 5, 10, 11, 14, 15, 16, 19, 21$ and 22, which have non zero in the s column.

The only rows that can be added to row 2 and give all zero's in the b_i and c_i columns are row 1, rows of type 3 and row 12. Both none of them we can get the $q + 1$ component in the a column. Rows of type 5 can be added to rows of type 8 to give only zeros in the b_i and c_i columns, but the term with $q + 1$ is excluded since the item $j = j$ is not given. Rows of type 10 and 11 suffer the same problem since term $b_j/b_{j'}^2$ with $j \neq k'$ prevents zero elements in the b_i and c_i columns. By adding the other rows to rows of type 14, 15 and 16, it has either only zeros in the b_i and c_i columns or the $q + 1$ term in column a. The term $c_k/c_{k'}^2$ with $k \neq k'$ is not given to rows of type 21 and 22, it prevents zero elements in the b_i and c_i columns.

Therefore, according to corollary 1, the modified q assumption is secure in the generic group model. □

Notice that the rows from 1 to 12 are the term in q-1 assumption. By modifying a to x, a^q to y, s to z and c to b in the rows of 1 to 3 and 12 to 22, we have all the terms in q-2 assumption.

APPENDIX E
SECURITY PROOF OF PROPOSED SCHEME

Our security proof is based on the security proofs in [16]. Specifically, we merge two proofs in [16] by simulating one master secret key and less size public parameters based on our modified q assumption. The details are given below:

Proof. Suppose there exists a polynomial-time adversary $\mathcal{A}$ that can break our scheme in sIND-CPA model with a non-negligible advantage. We build a simulator $\mathcal{B}$ that can attack the modified q assumption with a non-negligible advantage. For simplicity, let $\mathcal{C}_{kp}$ and $\mathcal{C}_{cp}$ be simulators for KP-ABE and CP-ABE in [16], the following proofs will

TABLE 5
Compact from of Matrix A and Target Vector $\tilde{A}$ for the Modified q Assumption

Type	Given Terms	Conditions	a	s	b_1	b_2	...	b_q	c_1	c_2	...	c_q
1	g		0	0	0	0	...	0	0	0	...	0
2	g^s		0	1	0	0	...	0	0	0	...	0
3	g^{a^i}	$\forall i \in [q]$	i	0	0	0	...	0	0	0	...	0
4	g^{b_j}	$\forall j \in [q]$	0	0	$[j:1]$				0	0	...	0
5	g^{sb_j}	$\forall j \in [q]$	0	1	$[j:1]$				0	0	...	0
6	$g^{a^i b_j}$	$\forall (i,j) \in [q,q]$	i	0	$[j:1]$				0	0	...	0
7	g^{a^i/b_j^2}	$\forall (i,j) \in [q,q]$	i	0	$[j:(-2)]$				0	0	...	0
8	$g^{a^i b_j/b_{j'}^2}$	$\forall (i,j,j') \in [2q,q,q]$ with $j \neq j'$	i	0	$[j:1, j':(-2)]$				0	0	...	0
9	$g^{a^i b_j/b_{j'}}$	$\forall (i,j,j') \in [2q,q,q]$ with $j \neq j'$	i	0	$[j:1, j':(-2)]$				0	0	...	0
10	$g^{sa^i b_j/b_{j'}}$	$\forall (i,j,j') \in [q,q,q]$ with $j \neq j'$	i	1	$[j:1, j':(-1)]$				0	0	...	0
11	$g^{sa^i b_j/b_{j'}^2}$	$\forall (i,j,j') \in [q,q,q]$ with $j \neq j'$	i	1	$[j:1, j':(-2)]$				0	0	...	0
12	$g^{(sa)^2}$		2	2	0	0	...	0	0	0	...	0
13	g^{c_k}	$\forall k \in [q]$	0	0	0	0	...	0	$[k:1]$			
14	g^{sac_k}	$\forall k \in [q]$	1	1	0	0	...	0	$[k:1]$			
15	g^{sa/c_k}	$\forall k \in [q]$	1	1	0	0	...	0	$[k:-1]$			
16	$g^{sa^2 c_k}$	$\forall k \in [q]$	2	1	0	0	...	0	$[k:1]$			
17	g^{a^q/c_k^2}	$\forall k \in [q]$	q	0	0	0	...	0	$[k:-2]$			
18	$g^{a^{q^2}/c_k^2}$	$\forall k \in [q]$	q^2	0	0	0	...	0	$[k:-2]$			
19	$g^{sac_k/c_{k'}}$	$\forall (k,k') \in [q,q]$ with $k \neq k'$	1	1	0	0	...	0	$[k:1, k':(-1)]$			
20	$g^{a^q c_k/c_{k'}^2}$	$\forall (k,k') \in [q,q]$ with $k \neq k'$	q	0	0	0	...	0	$[k:1, k':(-2)]$			
21	$g^{sa^{q+1} c_k/c_{k'}^2}$	$\forall (k,k') \in [q,q]$ with $k \neq k'$	$q+1$	1	0	0	...	0	$[k:1, k':(-2)]$			
22	$g^{(sa)^2 c_k/c_{k'}}$	$\forall (k,k') \in [q,q]$ with $k \neq k'$	2	2	0	0	...	0	$[k:1, k':(-1)]$			
$\tilde{A}^0$	$e(g,g)^{sa^{q+1}}$		$q+1$	1	0	0	...	0	0	0	...	0

reuse some components in [16] using $\mathcal{C}_{kp}$ and $\mathcal{C}_{cp}$. Notice that the modified q assumption includes instants of the q-1 assumption and the q-2 assumption.

Init. $\mathcal{B}$ receives the given terms from the assumption and the challenge information $(\mathbb{S}^*, \omega^*)$ from $\mathcal{A}$, where $\mathbb{S}^* = (N^* \in \mathbb{Z}_p^{\ell_s \times n_s}, \pi^* : [\ell_s] \to \mathbb{Z}_p)$ and $\omega^* = (\omega_1^*, \omega_2^*, \ldots, \omega_{k_o}^*)$.

Setup. $\mathcal{B}$ generates the public parameters by following the setup in $\mathcal{C}_{kp}$ and $\mathcal{C}_{cp}$. $\mathcal{C}_{kp}$ should output $\tilde{g}$, $\tilde{u}, \tilde{h}, \tilde{w}$ and $e(g,g)^{\tilde{\alpha}}$, and $\mathcal{C}_{cp}$ should output g, u, h, w, v and $e(g,g)^{\alpha}$. Based on the terms from the modified q assumption, some of them can be merged as:

$$g = \tilde{g} = g, \quad w = \tilde{w} = g^a,$$
$$e(g,g)^{\alpha} = e(g,g)^{\tilde{\alpha}} \cdot e(g,g)^{\tilde{a}} = e(g^a, g^{a^q}) \cdot e(g,g)^{\tilde{a}},$$

where $\tilde{a} \in \mathbb{Z}_p$ is known to $\mathcal{B}$. $\mathcal{B}$ returns the public parameters $g, u, h, w, v, \tilde{u}, \tilde{h}$ to $\mathcal{A}$.

Query Phases 1 and 2. $\mathcal{A}$ queries the following oracle adaptively.

$\mathcal{O}_{\mathsf{create}}(\psi, \mathbb{O})$: Parse $\psi = (\psi_1, \psi_2, \ldots, \psi_{k_s})$ and $\mathbb{O} = (M \in \mathbb{Z}_p^{\ell_o \times n_o}, \rho : [\ell_o] \to \mathbb{Z}_p)$. $\mathcal{B}$ first chooses a random identifier id, then picks $\beta_{id} \in \mathbb{Z}_p$ and derives the transformation key tk_{id} as following cases:

Case 1: If $\omega^* \in \mathbb{O}$ and $\psi \in \mathbb{S}^*$. $\mathcal{B}$ returns the transformation key by running $\mathsf{TKGen}(pp, \beta_{id}, id, pk_{id}, (\psi, \mathbb{O}))$.

Case 2: If $\omega^* \notin \mathbb{O}$. $\mathcal{B}$ picks $\alpha_{id} \in \mathbb{Z}_p$, then simulates the KP-ABE component embedded $\alpha - \alpha_{id}$ by reusing the simulation in $\mathcal{C}_{kp}$ and the CP-ABE embedded α_{id} normally. $\mathcal{B}$ combines these two keys and modify every term with an additional exponent β_{id}. The details are given below:

$\mathcal{B}$ reuses the key generation oracle of $\mathcal{C}_{kp}$ to generates the secret key with secret information $\alpha - \tilde{a}$ as:

$$sk_{kp} = (\{sk_{1,i}, sk_{2,i}, sk_{3,i}\}_{i \in [\ell_o]}).$$

$\mathcal{B}$ picks $x_2, \ldots x_{n_o} \in \mathbb{Z}_p$ and set $\vec{x} = (\tilde{a} - \alpha_{id}, x_2, .., x_2)^\top \in \mathbb{Z}_p^{n_o \times 1}$, then computes

$$\vec{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_{\ell_o}) = M\vec{x}.$$

It modifies the secret key sk_{kp} as:

$$\begin{aligned} sk'_{kp} &= (\{sk'_{1,i}, sk'_{2,i}, sk'_{3,i}\}_{i \in [\ell_o]}) \\ &= (\{sk_{1,i} \cdot g^{\lambda_i}, sk_{2,i}, sk_{3,i}\}_{i \in [\ell_o]}). \end{aligned}$$

$\mathcal{B}$ runs $\mathcal{CP}.\mathsf{KeyGen}(pp, \alpha_{id}, \psi)$ to generates the secret key sk_{cp} as:

$$sk_{cp} = (sk_1, sk_2, \{sk_{3,i}, sk_{4,i}\}_{i \in [k_s]}).$$

$\mathcal{B}$ then returns the transformation key based on sk'_{kp}, sk_{cp} and β_{id} as:

$$\begin{aligned} tk_{id} &= (\{tk_{1,i}, tk_{2,i}, tk_{3,i}\}_{i \in [\ell_o]}, tk_4, \{tk_{5,\tau}, tk_{6,\tau}\}_{\tau \in [k_s]}), \\ &= ((sk'_{1,i} \cdot sk_1)^{\beta_{id}}, sk_{2,i}'^{\beta_{id}}, sk_{3,i}'^{\beta_{id}}\}_{i \in [\ell_o]}, sk_2^{\beta_{id}}, \\ &\quad \{sk_{3,\tau}^{\beta_{id}}, sk_{4,\tau}^{\beta_{id}}\}_{\tau \in [k_s]}). \end{aligned}$$

Case 3: If $\psi \notin \mathbb{S}^*$. $\mathcal{B}$ picks $\alpha_{id} \in \mathbb{Z}_p$, then simulates the CP-ABE component embedded $\alpha - \alpha_{id}$ by reusing the simulation in $\mathcal{C}_{cp}$ and the KP-ABE embedded α_{id} normally. $\mathcal{B}$

combines these two keys and modify every term with an additional exponent β_{id}. The details are given below:

$\mathcal{B}$ reuses the key generation oracle of $\mathcal{C}_{cp}$ to generates the secret key with secret information α as:

$$sk_{cp} = (sk_1, sk_2, \{sk_{3,i}, sk_{4,i}\}_{i\in[k_s]}).$$

$\mathcal{B}$ then modifies the secret key sk_{cp} as:

$$\begin{aligned} sk'_{cp} &= (sk'_1, sk'_2, \{sk'_{3,i}, sk'_{4,i}\}_{i\in[k_s]}) \\ &= (sk_1 \cdot g^{-\alpha_{id}}, sk_2, \{sk_{3,i}, sk_{4,i}\}_{i\in[k_s]}). \end{aligned}$$

$\mathcal{B}$ runs $\mathcal{KP}.\mathsf{KeyGen}(pp, \alpha_{id}, \mathcal{O})$ to generates the secret key sk_{kp} as:

$$sk_{kp} = (\{sk_{1,i}, sk_{2,i}, sk_{3,i}\}_{i\in[\ell_o]}).$$

$\mathcal{B}$ then returns the transformation key based on sk_{kp}, sk'_{cp} and β_{id} as:

$$\begin{aligned} tk_{id} &= (\{tk_{1,i}, tk_{2,i}, tk_{3,i}\}_{i\in[\ell_o]}, tk_4, \{tk_{5,\tau}, tk_{6,\tau}\}_{\tau\in[k_s]}), \\ &= ((sk_{1,i} \cdot sk'_1)^{\beta_{id}}, sk_{2,i}^{\beta_{id}}, sk_{3,i}^{\beta_{id}}\}_{i\in[\ell_o]}, sk_2^{\prime\beta_{id}}, \\ &\quad \{sk_{3,\tau}^{\prime\beta_{id}}, sk_{4,\tau}^{\prime\beta_{id}}\}_{\tau\in[k_s]}). \end{aligned}$$

$\mathcal{O}_{\mathsf{corrupt}}(id)$: If id has not been issued, $\mathcal{B}$ returns $\perp$. Otherwise, $\mathcal{B}$ returns β_{id}. Notice that $\mathcal{A}$ cannot query this oracle for the user id who has $(\psi, \mathbb{O})$ $s.t.$ $\omega^* \in \mathbb{O}$ and $\psi \in \mathbb{S}^*$.

Challenge. $\mathcal{A}$ submits two messages m_0 and m_1 with the same length. $\mathcal{B}$ picks a bit $b \in \{0, 1\}$ and follows the challenge phase of $\mathcal{C}_{kp}$ and $\mathcal{C}_{cp}$ to derive the challenge ciphertext and returns it to $\mathcal{A}$. Notice that both $\mathcal{C}_{kp}$ and $\mathcal{C}_{cp}$ generate the challenge ciphertext based on the randomness based on the term g^s, which is given in the modified q assumption.

Guess. $\mathcal{A}$ outputs a guess b' for the challenge bit. If $b' = b$, it outputs 0 to claim that challenge term is a valid tuple. Otherwise, it outputs 1.

If the challenge tuple is valid, the simulation in above game is proper since we have $e(g, g)^{sa^{q+1}}$ and $\mathcal{B}$ will break the modified q assumption by using $\mathcal{A}$. If the challenge tuple is invalid, a random term $R \in \mathbb{G}_T$ instead of $e(g, g)^{sa^{q+1}}$ leads to misbehave challenge ciphertext. Hence, $\mathcal{A}$ cannot learn any information from this ciphertext and the advantage of $\mathcal{A}$ is 0. Hence, if $\mathcal{A}$ can break our security game, then we can build $\mathcal{B}$ to break the modify q assumption. Therefore, the security of our proposed *SA-DP-ABE* relies on the modify q assumption, which has been proved secure in generic group model. □

ACKNOWLEDGMENTS

This research is supported in part by AXA Research Fund, the Key Research and Development Program of Shaanxi [2019KW-053], the New Star Team of Xi'an University of Posts and Telecommunications [2016-02], the National Natural Science Foundation of China (No. U1636219, U1804263 and 61702105), the National Key R&D Program of China (No. 2016QY01W0105, 2016YFB0801303), and the Science and Technology Innovation Talent Project of Henan Province (No. 184200510018).

REFERENCES

[1] D. Reinsel, J. Gantz, and J. Rydning, "Data age 2025: The evolution of data to life-critical," 2017. [Online]. Available: https://www.seagate.com/www-content/our-story/trends/files/Seagate-WP-DataAge2025-March-2017.pdf

[2] T. D. T. Report, "Thales data threat report: Trend om encryption and data security," 2018. [Online]. Available: https://dtr-healthcare.thalesesecurity.com/pdf/2018-thales-data-threat-report-healthcare-edition-executive-summary.pdf

[3] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[4] M. Chiang and T. Zhang, "Fog and iot: An overview of research opportunities," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, Dec. 2016.

[5] J. Ren, H. Guo, C. Xu, and Y. Zhang, "Serving at the edge: A scalable iot architecture based on transparent computing," *IEEE Netw.*, vol. 31, no. 5, pp. 96–105, Aug. 2017.

[6] J. A. Akinyele, M. W. Pagano, M. D. Green, C. U. Lehmann, Z. N. J. Peterson, and A. D. Rubin, "Securing electronic medical records using attribute-based encryption on mobile devices," in *Proc. 1st ACM Workshop Secur. Privacy Smartphones Mobile Devices*, 2011, pp. 75–86.

[7] N. Attrapadung and H. Imai, "Dual-policy attribute based encryption," in *Proc. Int. Conf. Appl. Cryptography Netw. Secur.*, vol. 5536, pp. 168–185, 2009.

[8] Z. Wan, J. Liu, and R. H. Deng, "HASBE: A hierarchical attribute-based solution for flexible and scalable access control in cloud computing," *IEEE Trans. Inf. Forensics Secur.*, vol. 7, no. 2, pp. 743–754, Apr. 2012.

[9] A. Abbas and S. U. Khan, "A review on the state-of-the-art privacy-preserving approaches in the e-health clouds," *IEEE J. Biomed. Health Informat.*, vol. 18, no. 4, pp. 1431–1441, Jul. 2014.

[10] J. Yang, J. Li, and Y. Niu, "A hybrid solution for privacy preserving medical data sharing in the cloud environment," *Future Generation Comput. Syst.*, vol. 43/44, pp. 74–86, 2015.

[11] Y. Yang, X. Zheng, W. Guo, X. Liu, and V. I. Chang, "Privacy-preserving fusion of iot and big data for e-health," *Future Generation Comput. Syst.*, vol. 86, pp. 1437–1455, 2018.

[12] Y. Yang, X. Zheng, X. Liu, S. Zhong, and V. Chang, "Cross-domain dynamic anonymous authenticated group key management with symptom-matching for e-health social system," *Future Generation Comput. Syst.*, vol. 84, pp. 160–176, 2018.

[13] Y. Yang, X. Liu, R. H. Deng, and Y. Li, "Lightweight sharable and traceable secure mobile health system," *IEEE Trans. Dependable Secure Comput.*, 2018, doi: 10.1109/TDSC.2017.2729556.

[14] L. Yeh, P. Chiang, Y. Tsai, and J. Huang, "Cloud-based fine-grained health information access control framework for lightweightiot devices with dynamic auditing andattribute revocation," *IEEE Trans. Cloud Comput.*, vol. 6, no. 2, pp. 532–544, Apr.-Jun. 2018.

[15] M. Green, S. Hohenberger, and B. Waters, "Outsourcing the decryption of ABE ciphertexts," in *Proc. 20th USENIX Conf. Secur.*, 2011, pp. 34–34.

[16] Y. Rouselakis and B. Waters, "Practical constructions and new proof methods for large universe attribute-based encryption," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2013, pp. 463–474.

[17] N. Attrapadung and S. Yamada, "Duality in ABE: Converting attribute based encryption for dual predicate and dual policy via computational encodings," in *Proc. Cryptographers Track RSA Conf.* vol. 9048, pp. 87–105, 2015.

[18] Y. Dodis, L. Reyzin, and A. D. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *Proc. Int. Conf. Theory Appl. Cryptographic Tech.*, 2004, pp. 523–540.

[19] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proc. 24th Annu. Int. Conf. Theory Appl. Cryptographic Tech.*, vol. 3494, pp. 457–473, 2005.

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

[21] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based encryption with non-monotonic access structures," in *Proc. 14th ACM Conf. Comput. Commun. Secur.*, 2007, pp. 195–203.

[22] N. Attrapadung, B. Libert, and E. de Panafieu, "Expressive key-policy attribute-based encryption with constant-size ciphertexts," in *Proc. Int. Workshop Public Key Cryptography*, vol. 6571, pp. 90–108, 2011.

[23] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Secur. Privacy*, 2007, pp. 321–334.
[24] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *Proc. Int. Workshop Public Key Cryptography*, vol. 6571, pp. 53–70, 2011.
[25] A. B. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in *Proc. Annu. Int. Conf. Theory Appl. Cryptographic Tech.*, vol. 6110, pp. 62–91, 2010.
[26] B. Waters, "Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions," in *Proc. Annu. Int. Cryptology Conf.*, 2009, pp. 619–636.
[27] Y. Zhang, D. Zheng, X. Chen, J. Li, and H. Li, "Efficient attribute-based data sharing in mobile clouds," *Pervasive Mobile Comput.*, vol. 28, pp. 135–149, 2016.
[28] V. Koppula and B. Waters, "Realizing chosen ciphertext security generically in attribute-based encryption and predicate encryption," *IACR Cryptology ePrint Archive*, vol. 2018, 2018, Art. no. 847.
[29] B. Qin, R. H. Deng, Y. Li, and S. Liu, "Server-aided revocable identity-based encryption," in *Proc. Eur. Symp. Res. Comput. Secur.*, 2015, pp. 286–304.
[30] H. Cui, R. H. Deng, Y. Li, and B. Qin, "Server-aided revocable attribute-based encryption," in *Proc. Eur. Symp. Res. Comput. Secur.*, 2016, pp. 570–587.
[31] B. Qin, Q. Zhao, D. Zheng, and H. Cui, "Server-aided revocable attribute-based encryption resilient to decryption key exposure," in *Proc. Int. Conf. Cryptology Netw. Secur.*, 2017, pp. 504–514.
[32] S. Xu, G. Yang, and Y. Mu, "Revocable attribute-based encryption with decryption key exposure resistance and ciphertext delegation," *Inf. Sci.*, vol. 479, pp. 116–134, 2018.
[33] Y. Rouselakis and B. Waters, "New constructions and proof methods for large universe attribute-based encryption," *IACR Cryptology ePrint Archive*, vol. 2012, 2012, Art. no. 583.
[34] B. Lynn, "Pbc library manual 0.5. 11," 2006. [Online]. Available: https://crypto.stanford.edu/pbc/manual

Shengmin Xu received the BSc degree from the School of Computing and Information Technology, University of Wollongong, Australia, in 2014 and the PhD degree in cryptography from the University of Wollongong, Australia, in 2018. He is currently a research fellow with the School of Information System, Singapore Management University, Singapore. His research interests include cryptography and information security.

Yingjiu Li is an associate professor with the School of Information Systems, Singapore Management University. His research interests include RFID security and privacy, mobile and system security, data application security and privacy. He has served on the editorial boards (and committee chair) of many information security international journals (and conferences). He is a member of the IEEE.

Robert H. Deng is AXA chair professor of cybersecurity and director of the Secure Mobile Centre, School of Information Systems, Singapore Management University (SMU). His research interests include data security and privacy, cloud security and Internet of Things security. He received the Outstanding University Researcher Award from National University of Singapore, Lee Kuan Yew Fellowship for Research Excellence from SMU, and Asia-Pacific Information Security Leadership Achievements Community Service Star from International Information Systems Security Certification Consortium. His professional contributions include an extensive list of positions in several industry and public services advisory boards, editorial boards and conference committees. These include the editorial boards of the *IEEE Security & Privacy Magazine*, the *IEEE Transactions on Dependable and Secure Computing*, the *IEEE Transactions on Information Forensics and Security*, the *Journal of Computer Science and Technology*, and Steering Committee Chair of the ACM Asia Conference on Computer and Communications Security. He is a fellow of the IEEE.

Yinghui Zhang is a professor of National Engineering Laboratory for Wireless Security (NELWS), Xi'an University of Posts & Telecommunications since 2018. He is also a research fellow with Singapore Management University. His research interests include public key cryptography, cloud security and wireless network security. He has published more than 80 research articles including ASIACCS, the *IEEE Transactions on Services Computing*, the *Computer Networks*, the *IEEE Internet of Things Journal*, the *Computers & Security*. He is a member of the IEEE.

Xiangyang Luo received the MS and PhD degrees from Zhengzhou Information Science and Technology Institute, Zhengzhou, China, in 2004 and 2010, respectively. From 2006 to 2007, he was a visiting scholar of the Department of Computer Science and Technology of Tsinghua University. His main research interests include network and information security. He is the author or co-author of more than 150 refereed international journal and conference papers. He is currently a professor of State Key Laboratory of Mathematical Engineering and Advanced Computing, China. In addition, he also recently served as guest editor of some special issues of International Journal the *Multimedia Tools and Applications*, the *Security and Communication Networks*, the *Journal of Universal Computer Science* and the *International Journal of Internet*, etc.

Ximeng Liu received the BSc degree in electronic engineering from Xidian University, Xian, China, in 2010 and the PhD degrees in cryptography from Xidian University, China, in 2015. Now, he is a full professor with the College of Mathematics and Computer Science, Fuzhou University, China. Also, he is a research fellow with the School of Information System, Singapore Management University, Singapore. His research interests include cloud security, applied cryptography and big data security. He has published more than 100 research articles include the *IEEE Transactions on Information Forensics and Security*, the *IEEE Transactions on Dependable and Secure Computing*, the *IEEE Transactions on Computers*, the *IEEE Transactions on Industrial Informatics*, the *IEEE Transactions on Services Computing*, the *IEEE Transactions on Cloud Computing*, and IEEE INFOCOM. He awards "Minjiang Scholars Distinguished Professor, "Qishan Scholars in Fuzhou University, and ACM SIGSAC China Rising Star Award (2018). He served as a leader guest editor for Wireless Communications and Mobile Computing. He a member of the IEEE, ACM, CCF.