6-2021

Expressive bilateral access control for Internet-of-Things in cloud-fog computing

Shengmin XU

Jianting NING

Jinhua MA

Xinyi HUANG

Hwee Hwa PANG
Singapore Management University, hhpang@smu.edu.sg

See next page for additional authors

Citation

XU, Shengmin; NING, Jianting; MA, Jinhua; HUANG, Xinyi; PANG, Hwee Hwa; and DENG, Robert H.. Expressive bilateral access control for Internet-of-Things in cloud-fog computing. (2021). *Proceedings of the 26th ACM Symposium on Access Control Models and Technologies (SACMAT 2021), Virtual Conference, June 16-18*. 143-154.
Available at: https://ink.library.smu.edu.sg/sis_research/6739

Author

Shengmin XU, Jianting NING, Jinhua MA, Xinyi HUANG, Hwee Hwa PANG, and Robert H. DENG

Expressive Bilateral Access Control for Internet-of-Things in Cloud-Fog Computing

Shengmin Xu
Fujian Normal University
Fuzhou, Fujian, China
smxu1989@gmail.com

Jianting Ning*
Fujian Normal University
Fuzhou, Fujian, China
jtning88@gmail.com

Jinhua Ma†
Fujian Normal University
Fuzhou, Fujian, China
jinhuama55@hotmail.com

Xinyi Huang
Fujian Normal University
Fuzhou, Fujian, China
xyhuang81@gmail.com

Hwee Hwa Pang
Singapore Management University
Singapore
hhpang@smu.edu.sg

Robert H. Deng
Singapore Management University
Singapore
robertdeng@smu.edu.sg

ABSTRACT

As a versatile system architecture, cloud-fog Internet-of-Things (IoT) enables multiple resource-constrained devices to communicate and collaborate with each other. By outsourcing local data and immigrating expensive workloads to cloud service providers and fog nodes (FNs), resource-constrained devices can enjoy data services with low latency and minimal cost. To protect data security and privacy in the untrusted cloud-fog environment, many cryptographic mechanisms have been invented. Unfortunately, most of them are impractical when directly applied to cloud-fog IoT computing, mainly due to the large number of resource-constrained end-devices (EDs). In this paper, we present a secure cloud-fog IoT data sharing system with bilateral access control based on a new cryptographic tool called *lightweight matchmaking encryption*. Our system enforces both sender access control and receiver access control simultaneously and adapts to resource-constrained EDs by outsourcing costly workloads to FNs. We conduct extensive experiments to demonstrate the superior performance of our system to the most relevant solutions in the literature.

CCS CONCEPTS

• **Security and privacy** → **Public key encryption**; • **Computer systems organization** → *Cloud computing*; • **Information systems** → Data management systems.

KEYWORDS

Internet-of-Things; Bilateral Access Control; Cloud-Fog Computing

ACM Reference Format:
Shengmin Xu, Jianting Ning, Jinhua Ma, Xinyi Huang, Hwee Hwa Pang, and Robert H. Deng. 2021. Expressive Bilateral Access Control for Internet-of-Things in Cloud-Fog Computing. In *Proceedings of the 26th ACM Symposium on Access Control Models and Technologies (SACMAT '21), June 16–18, 2021, Virtual Event, Spain.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3450569.3463561

1 INTRODUCTION

Cloud computing is a widely accepted paradigm with elastic services and low maintenance cost for many real-world applications, such as infrastructure management, home automation, and environmental monitoring [18, 32, 34]. However, the strict system architecture of cloud computing limits its flexibility [2] for applications that demand immediate or real-time responses, such as intelligent transportation, smart home, and augmented reality. To address this limitation, fog computing [25, 27] was introduced to bring intelligence closer to data sources [8, 14, 19, 26]. Fog computing enables users to enjoy a variety of customized services, such as location awareness, mobility support, and geographic distribution.

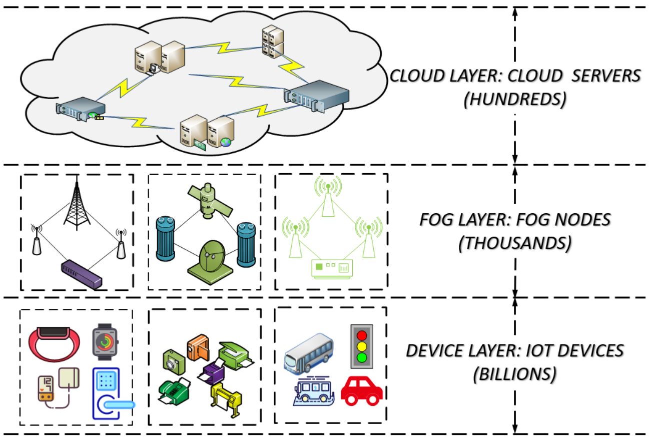


Figure 1: System Architecture of Cloud-Fog IoT

To make full use of fog computing, a new system architecture namely *cloud-fog IoT*, as shown in Fig. 1, was introduced [20] to

[*] Jianting Ning is also with State Key Laboratory of Information Security (Institute of Information Engineering, Chinese Academy of Sciences), Beijing, 100093.
[†] Jinhua Ma is the corresponding author.

SACMAT '21, June 16–18, 2021, Virtual Event, Spain
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8365-3/21/06…$15.00
https://doi.org/10.1145/3450569.3463561

provide interconnections of billions of IoT devices to collect and exchange data. The architecture is a three-layer pyramid, in which (1) cloud service providers (CSPs) offer global access control; (2) FNs provide regional access control and customized services; and (3) IoT devices upload and request data from FNs. FNs play an essential role in this ecosystem that offers a variety of data services by storing frequently used data as caches to provide machine-to-machine communication with low latency. Besides, FNs offer on-demand data services and collaborate for data intelligence to reduce the consumption of computation and bandwidth for data analysis and transmissions. For example, in intelligent transportation, the cloud-fog IoT system enables efficient and secure data interactions between application servers and vehicles. A CSP receives messages from an application server and shares them with FNs, e.g., roadside units, which in turn process customized data services and broadcast these messages to onboard IoT devices, e.g., on-board units. However, an open and untrusted cloud-fog IoT system hinders the widespread development of fog computing, and in particular data security and privacy has been a serious concern.

A common approach to achieving data security and privacy protection is encryption. However, encryption should not impede data sharing among authorized services and devices. Sahai and Waters [24] introduced a cryptographic primitive, called attribute-based encryption (ABE), to share data with fine-grained access control. ABE is a type of one-to-many public-key encryption in which the secret key of a user and a ciphertext are dependent upon attributes. There are two flavors of ABE: key-policy ABE (KP-ABE) and ciphertext-policy ABE (CP-ABE). KP-ABE was introduced [15] for providing fine-grained content-based access control. In KP-ABE, a user's secret key is generated based on an access policy expressed in terms of attributes (aka keywords) that defines the privileges of the user, and data are encrypted over a set of attributes. A user can decrypt a ciphertext if the set of attributes associated with the ciphertext satisfies the user's access policy. CP-ABE was introduced [5] for providing flexible role-based access control. In CP-ABE, the secret key of a user is bound to a set of attributes to represent the user's privileges, and an access policy is associated with a ciphertext. A user can decrypt a ciphertext if the user's set of attributes satisfies the access policy of the ciphertext. Both KP-ABE and CP-ABE achieve access control of encrypted data at a fine-grained level, even if the server holds the data is untrusted and multiple unauthorized users launch collusion attacks.

To facilitate retrieval and search of ciphertexts by keywords, attribute-based keyword search (ABKS) was proposed [28, 35] as a cryptographic tool to conduct expressive keyword search over ciphertexts without costly data decryption. Unfortunately, it has been shown that the existing ABKS solutions suffer from searchable pattern leakage attacks [9, 21, 33].

Ateniese et al. [3] introduced a novel cryptographic primitive, dubbed matchmaking encryption, to provide bilateral access control and presented a concrete instantiation in an identity-based setting. In a bilateral access control system, a sender can specify a decryption policy for receivers and a receiver can specify a source identification policy for senders. Fig. 2 gives a sketch of bilateral access control. Each sender has an encryption key and each receiver has a decryption key, where both keys are issued by a trusted key generation center (KGC). Each sender can encrypt data by specifying a decryption policy and embedding her/his encryption key, so that only receivers with decryption keys satisfy the decryption policy can decrypt the ciphertext. Each receiver can specify a source identification policy for incoming ciphertexts so that only those generated by senders whose encryption keys satisfy the source identification policy are accepted. To provide bilateral access control at the fine-grained level, Xu et al. [30] proposed a concrete instantiation of the matchmaking encryption in the attribute-based setting. However, both solutions [3, 30] suffer from a costly data decryption process and cannot adapt to resource-constrained IoT devices.

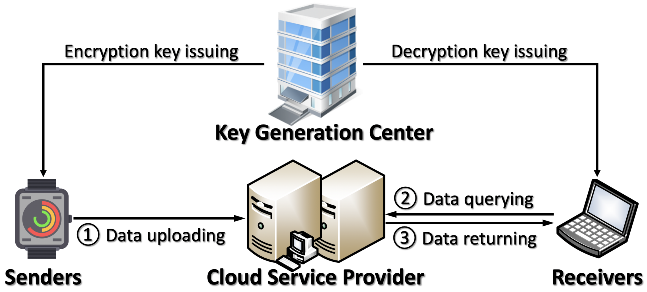


Figure 2: Sketch of Bilateral Access Control

Bilateral access control offers sender access control and receiver access control simultaneously and hence is an important technical solution to meet the various requirements of data security and privacy regulations, including the collection limitation principle and the data quality principle, as mentioned in General Data Protection Regulation (GDPR) [12]. The collection limitation principle requires that sensitive data must be protected and should not be abused by any unauthorized party. For example, in intelligent transportation, an application server encrypts data with a specified decryption policy (e.g., "*(MICRO OR SEDAN) AND Los Angeles*") for secure communications with authorized vehicles of type MICRO or SEDAN registered in Los Angeles. The data quality principle requires that data sources be identifiable to ensure reliability, consequently preserving the interests of receivers. For example, the application server attaches its attributes to a ciphertext (e.g., "*Road Info AND Log Angeles*"), and roadside units can help vehicles to identify useful ciphertexts from a substantial amount of ciphertexts without performing costly data decryption.

Therefore, it is desirable to build an efficient solution to address security threats and satisfy realistic requirements in a cloud-fog IoT system. To address this problem, in this paper, we introduce a new cloud-fog IoT system with the following contributions:

- We design a secure bilateral access control cloud-fog IoT system with rigorous system definition via a system model and a threat model. The starting point of our design is to meet the security requirements demanded by most real-world applications of IoT devices that are constrained with memory, computation, and battery.
- We introduce the notion and a concrete construction of lightweight matchmaking encryption (LME) and formally prove its

security to support the security of our proposed system. LME offers secure data sharing with the following properties:

- *Bilateral access control at a fine-grained level.* A sender can specify a decryption policy to control receivers. A receiver can specify a source identification policy to only accept ciphertexts from certain types of senders. Both policy specifications are at a fine-grained level.
- *Data source identification with sender anonymity and unlinkability.* A receiver can specify a source identification policy to discard undesirable ciphertexts without costly data decryption. Meanwhile, sender anonymity and unlinkability are preserved. In LME, a sender's encryption key is associated with a set of attributes. The policy specification allows the receiver to identify ciphertexts generated by senders whose attributes associated with their encryption keys satisfy the source identification policy. In practice, different senders may possess same attributes. Hence, the receiver cannot determine who is the real generator of a ciphertext or link multiple ciphertexts generated by the same sender.
- *Lightweight data decryption and outsourced data source identification.* To adapt to resource-constrained devices, some costly workloads, such as data decryption and data source identification, are outsourced to a semi-trusted FN. By outsourcing the main workload of the expensive data decryption, a receiver only takes one exponentiation and one multiplication for revealing a message. FN can help a receiver to perform data source identification, hence, the receiver is relieved from performing any operation for data source identification.

2 OVERVIEW

In this section, we give an overview of the proposed bilateral access control system in terms of system sketch, technical sketch, and function realization.

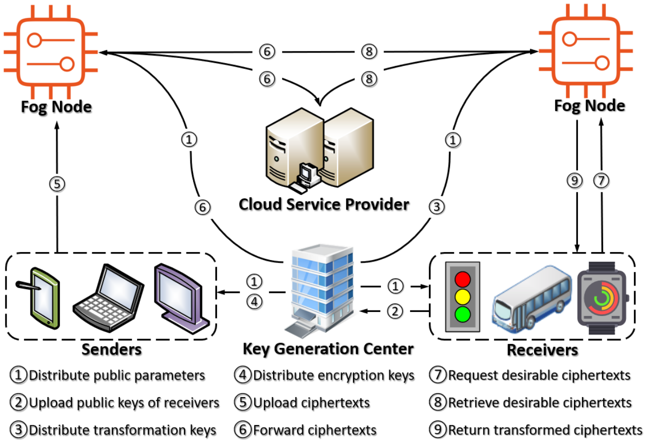

Fog Node
Fog Node
⑥
⑧
⑥
⑧
①
①
⑤
⑥
Cloud Service Provider
③
⑨
⑦
①
④
①
②
Senders
Key Generation Center
Receivers
①Distribute public parameters
②Upload public keys of receivers
③Distribute transformation keys
④Distribute encryption keys
⑤Upload ciphertexts
⑥Forward ciphertexts
⑦Request desirable ciphertexts
⑧Retrieve desirable ciphertexts
⑨Return transformed ciphertexts

Figure 3: System Model of Cloud-Fog IoT

System Overview. Fig. 3 illustrates the system model of the cloud-fog IoT system. The cloud-fog IoT system consists of four typical entities: a KGC, a CSP, FNs, and EDs, where EDs can be further classified into senders and receivers. The detailed analysis of characteristics in each entity is given below:

- The KGC is responsible for initializing system parameters (See ①) and issuing keys, including transformation keys (See ③) and encryption keys (See ④).
- The CSP accommodates ciphertexts from FNs (See ⑥) and shares these ciphertexts to other FNs (See ⑧).
- FNs are facilities or infrastructures to offer on-demand data services with low latency. FNs are responsible for aggregating ciphertexts from senders (See ⑤) and interacting with other FNs and the CSP (See ⑥ and ⑧). Besides, FNs are responsible for receiving data requests from receivers (See ⑦) and transferring ciphertexts from FNs and CSP to receivers (See ⑨).
- EDs are IoT devices constrained with memory, physical size, and battery. An ED could be either a sender or a receiver.
  - As the sender, the ED has an encryption key issued by the KGC (See ④), where the encryption key is cryptographically bound to a set of attributes of the sender and is kept secret by the sender. The ED collects data from surrounding areas, encrypts and uploads them to FNs (See ⑤). The uploaded ciphertext associated with a decryption policy specified by the sender, where the receiver satisfies the decryption policy can reveal the data.
  - As the receiver, the ED generates a public-secret key pair for lightweight decryption (See ②) and has a transformation key (See ③) issued by the KGC, where the transformation key is bound to a set of attributes of the receiver and the secret key is kept secret by the receiver. The ED can specify a source identification policy to an FN to acquire ciphertexts from certain types of senders (See ⑦), and receives a transformed ciphertext from the FN (See ⑨). The transformed ciphertexts must be short, and the decryption progress must be inexpensive.

To ensure secure data sharing, an adversary cannot (1) derive any valid encryption key, (2) generate any valid ciphertext without an encryption key issued by the KGC, and (3) reveal any ciphertext without a valid secret key that satisfies the access policy associated with the ciphertext. More detailed system model and threat model are given in Section 4.

Technical Overview. Our LME scheme is compatible with the advantages of matchmaking attribute-based encryption (MABE) [30] and outsourced ABE [16]. MABE [30] realizes *bilateral access control* and *data source identification*, but it cannot adapt to the IoT ecosystem due to its costly ABE decryption process. Outsourced ABE [16, 17, 22] achieves *lightweight decryption*, but the data access control is unilateral since senders can specify the access policy of receivers only.

We now sketch the technical ideas behind the design of LME. We apply ABE [23] to achieve receiver access control, where the security can be reduced to the decisional q-1 assumption. Then, we modify ABE [23] to the signature version to achieve sender access control, where the security can be reduced to computational q-1 assumption. By issuing the encryption key that specifies a set of attributes, senders can attach their (part of) attributes to endorse the data source, and anyone can perform data source identification. Inspired [16], we apply the outsourced decryption. By outsourcing the public key of an ED, any semi-trusted party, such as FNs, can help the ED to operate the ciphertext transformation to the form of

ElGamal ciphertext. Hence, the communication and computation costs at the ED are at the ElGamal level.

Functional Overview. We sketch the importance of functionality our system achieved:

Bilateral access control at a fine-grained level. Our proposed solution achieves bilateral access control at the fine-grained level. Many existing cryptographic tools only consider receiver access control, such as public key infrastructure (PKI) [13], identity-based encryption (IBE) [7], ABE [24] and dual-policy ABE (DP-ABE) [4]. While access control encryption (ACE) [11] and matchmaking encryption [3] offer bilateral access control, they are impractical in the cloud-fog IoT computing due to the following reasons. ACE requires a third party called sanitizer to monitor the data transmission from senders to receivers. By the inspector mechanism, it enforces access policy "*no read-up, no write-down*" [6], which is not suitable in the cloud-fog IoT system since dynamic and abundant cloud users. Matchmaking encryption allows receivers to identify data sources without revealing underlying messages, and also preserve data and sender privacy. However, the existing solution of matchmaking encryption is either in an identity-based setting (MIBE is short for matchmaking identity-based encryption), or in an attribute-based setting with a large workload at the ED. We introduce the first MABE with outsourced data sourced identification and outsourced decryption and provide a formal definition with security proofs.

Identifying data source without revealing messages. Identifying data sources without costly data decryption is a desirable property that mitigates security threats, such as impersonate attacks and denial-of-service attacks, in the untrustworthy network. Our cloud-fog IoT system deploys MABE to identify data sources without performing data decryption. Some existing solutions achieve a similar goal. One such solution is ABKS, which allows receivers to search for ciphertexts that contain receiver-specified keywords without revealing messages. To preserve keyword privacy and search pattern privacy, it requires multiple rounds of interactions to generate searchable queries. However, a variety of attacks, including passive attacks (e.g., leakage-abuse attacks [10, 21]) and active attacks (e.g., file-injection attacks [33]), threaten current keyword search solutions. In comparison with ABKS, our solution provides a novel and promising strategy for retrieving useful information from a substantial amount of ciphertexts with privacy preservation.

Privacy-preserving data sharing with sender anonymity and unlinkability. Data source identification may breach sender privacy. The existing solutions, such as ACE and MIBE, suffer from this problem. ACE requires a third party to reveal sender privacy. MIBE is vulnerable to brute force attacks as sender identity can be discovered by the receiver through an exhaustive search on all possible identities in data decryption. Hence, it remains challenging to preserve sender privacy during the data source identification process. Our system enables the data sender to pick a non-unique attribute set to encrypt data such that the sender's privacy is preserved among multiple users who share the same set of attributes. More importantly, our scheme applies re-randomization technology to achieve ciphertext (including the transformed ciphertext) unlinkability, which means that no one except the data sender himself/herself can link two different ciphertexts from the same sender.

Lightweight data decryption and outsourced data source identification. IoT devices are usually resource-constrained with memory, physical size, and battery. It is desirable to design a secure and efficient cloud-fog IoT scheme with lightweight data decryption and outsourced data source identification. Although lightweight data decryption has been widely applied in IBE and ABE [16], there is no formal treatment in MABE. Besides, the outsourced data source identification is an important property to immigrate the workload of EDs. In this paper, we achieve lightweight data decryption and outsourced data source identification simultaneously. By outsourcing the heavy workload of the costly data decryption and data source identification to semi-trusted FNs, EDs perform lightweight operations only (one exponentiation and one multiplication for decrypting each ciphertext) and are relieved of performing any operation for data source identification. Therefore, our scheme is particularly suitable for the cloud-fog system with resource-constrained EDs.

3 PRELIMINARIES

In this section, we introduce bilinear map, linear secret sharing scheme, hard assumption, and the definition of LME, which are used in our proposed cloud-fog IoT system.

3.1 Notation

Let $\mathbb{N}$ be a set of natural numbers. For $n \in \mathbb{N}$, let $[n]$ represent integers from 1 to n. If a and b are strings, $a||b$ denotes the concatenation of a and b. If a and b are two ciphertexts, $a \equiv b$ means they have the same distribution, e.g., encrypting the identical messages with different randomnesses. If $\mathcal{S}$ be an attribute set and $\mathbb{S}$ be a policy, $\mathcal{S} \models \mathbb{S}$ denotes the attribute set $\mathcal{S}$ satisfies the policy $\mathbb{S}$, and $\mathcal{S} \not\models \mathbb{S}$ represents the attribute set $\mathcal{S}$ does not satisfy the policy $\mathbb{S}$.

3.2 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 $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is said to be an admissible bilinear pairing if the following properties hold.

(1) Bilinearity: for all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p$, $e(u^a, v^b) = e(u, v)^{ab}$.

(2) Non-degeneration: $e(g, g) \neq 1$.

(3) Computability: it is efficient to compute $e(u, v)$ for any $u, v \in \mathbb{G}$.

We say that $(\mathbb{G}, \mathbb{G}_T)$ are bilinear map groups if there exists a bilinear pairing $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ as above.

3.3 Linear Secret Sharing Scheme (LSSS)

Let $\mathbb{M}$ denote an $\ell \times n$ matrix over the base field $\mathbb{F}$ and ρ be a mapping function from the set $[\ell]$ to an attribute universe. An LSSS [31] is of the type $(\mathbb{M}, \rho)$, where it satisfies attributes ψ if $(1, 0, ..., 0) \in \mathbb{F}^n$ is contained in $\text{Span}_{\mathbb{F}}(\mathbb{M}_i : \rho(i) \in \psi)$, where $\mathbb{M}_i$ is the i^{th} row of $\mathbb{M}$.

3.4 Assumptions

Definition 3.1 (Decisional q-1 Assumption [23]). Let $a, s, b_1, b_2, ..., b_q \in \mathbb{Z}_p$ be random terms and $g \in \mathbb{G}$ be a group generator of bilinear group $\mathbb{G}$ with prime order p. Decisional q-1 assumption is that no probabilistic polynomial-time algorithm can distinguish the term $e(g, g)^{sa^{q+1}}$ from any random term in $\mathbb{G}_T$ with more than a negligible advantage by giving the following terms:

$$\begin{array}{ll} g, g^s & \\ g^{a^i}, g^{b_j}, g^{sb_j}, g^{a^i b_j}, g^{a^i/b_j^2} & \forall (i,j) \in [q,q]; \\ g^{a^i b/b_{j'}^2} & \forall (i,j,j') \in [2q,q,q] \text{ with } j \neq j'; \\ g^{a^i/b_j} & \forall (i,j) \in [2q,q] \text{ with } i \neq q+1; \\ g^{sa^i b_j/b_{j'}}, g^{sa^i b_j/b_{j'}^2} & \forall (i,j,j') \in [q,q,q] \text{ with } j \neq j'. \end{array}$$

Based on decisional q-1 assumption, we propose computational q-1 assumption. In decisional q-1 assumption, the adversary wins the game if he can distinguish $e(g,g)^{sa^{q+1}}$ from any random term in $\mathbb{G}_T$. In computational q-1 assumption, the adversary wins the game if he can output $e(g,g)^{sa^{q+1}}$. Following is the formal definition of computational q-1 assumption.

Definition 3.2 (Computational q-1 Assumption). Let $a, s, b_1, b_2, \dots, b_q \in \mathbb{Z}_p$ be random terms and $g \in \mathbb{G}$ be a group generator of bilinear group $\mathbb{G}$ with prime order p. The computational q-1 assumption is that no probabilistic polynomial-time algorithm can output the term $e(g,g)^{sa^{q+1}}$ with more than a negligible advantage by giving the terms as in decisional q-1 assumption.

3.5 Definition of Lightweight Matchmaking Encryption

In the following, we present LME based on the definitions of matchmaking encryption [3] and outsourced ABE [16]. Our scheme offers not only expressive bilateral access control but also a lightweight decryption mechanism for resource-constrained devices.

Definition 3.3 ($\mathcal{LME}$). A lightweight matchmaking encryption $\mathcal{LME}$ with attribute universes Ω_{snd} and Ω_{rcv} that supports policies $\mathcal{P}_{\mathsf{snd}}$ and $\mathcal{P}_{\mathsf{rcv}}$, an identity space $\mathcal{I}$ and a message space $\mathcal{M}$. An $\mathcal{LME}$ involves five typical entities: a KGC, senders, receivers, FNs, and a CSP. It consists of the following eight algorithms:

$\mathsf{Setup}(1^\lambda) \rightarrow (mpk, msk)$: The probabilistic setup algorithm is run by the KGC. It takes a security parameter $\lambda \in \mathbb{N}$ as input, and outputs a master public key mpk and a master secret key msk. The KGC publishes mpk and keeps msk secret. We implicitly assume that all other algorithms take mpk as input.

$\mathsf{KeyGen}(id) \rightarrow (pk_{id}, sk_{id})$: The probabilistic key generation algorithm is run by each sender. It takes an identity of a sender $id \in \mathcal{I}$ as input, and outputs a public key pk_{id} and a secret key sk_{id}. Each sender sends pk_{id} to the KGC and keeps sk_{id} secret.

$\mathsf{TKGen}(msk, pk_{id}, \mathcal{R}) \rightarrow tk_{id}$: The probabilistic transformation key generation algorithm is run by the KGC. It takes a master secret key msk, a public key pk_{id} and a set of receiver's attributes $\mathcal{R} \subseteq \Omega_{\mathsf{rcv}}$ as input, and outputs a transformation key tk_{id}. The KGC sends tk_{id} to FNs.

$\mathsf{EKGen}(msk, \mathcal{S}) \rightarrow ek_{\mathcal{S}}$: The probabilistic encryption key generation algorithm is run by the KGC. It takes a master secret key msk and a set of sender's attributes $\mathcal{S} \subseteq \Omega_{\mathsf{snd}}$ as input, and outputs an encryption key $ek_{\mathcal{S}}$. The KGC sends $ek_{\mathcal{S}}$ to senders.

$\mathsf{Enc}(ek_{\mathcal{S}}, \hat{\mathcal{S}}, \mathbb{R}, m) \rightarrow c$: The probabilistic encryption algorithm is run by each sender. It takes an encryption key $ek_{\mathcal{S}}$, a set of sender's attributes $\hat{\mathcal{S}} \subseteq \mathcal{S}$, a policy of receivers $\mathbb{R} \in \mathcal{P}_{\mathsf{rcv}}$ and a message $m \in \mathcal{M}$ as input, and outputs a ciphertext c. The sender outsources c to the CSP.

$\mathsf{Verify}(\mathbb{S}, c) \rightarrow \{0,1\}$: The deterministic verification algorithm is run by each FN. It takes a policy of a receiver $\mathbb{S} \in \mathcal{P}_{\mathsf{snd}}$ and a ciphertext c associated $\hat{\mathcal{S}}$ as input, and outputs a bit 1 if $\hat{\mathcal{S}} \models \mathbb{S}$; otherwise, outputs 0. The FN returns c to the receiver if it outputs 1.

$\mathsf{Transfer}(tk_{id}, c) \rightarrow \hat{c}$: The deterministic transformation algorithm is run by each FN. It takes a transformation key tk_{id} and a ciphertext c as input, and outputs a transformed ciphertext $\hat{c}$. The FN returns $\hat{c}$ to the receiver. Note that transformation algorithm could be a probabilistic algorithm by re-randomizing $\hat{c}$.

$\mathsf{Dec}(sk_{id}, \hat{c}) \rightarrow m$: The deterministic decryption algorithm is run by each receiver. It takes a secret key sk_{id} and a transformed ciphertext $\hat{c}$ as input, and outputs a message $m \in \mathcal{M}$.

Ateniese et al. [3] and Green et al. [16] introduced the security models for matchmaking encryption and outsourced ABE, respectively. We refine these models to define models called selectively indistinguishable against chosen plaintext attacks (sIND-CPA) and existential unforgeability under a chosen message attack (EU-CMA) for our proposed LME.

*Definition 3.4 (*sIND-CPA *in $\mathcal{LME}$).* Let $\mathcal{O}$ denote a set of oracles: a key generation oracle $\mathcal{O}_{\mathsf{KeyGen}}(\cdot,\cdot)$, a corrupt oracle $\mathcal{O}_{\mathsf{Corrupt}}(\cdot,\cdot)$, a transformation key generation oracle $\mathcal{O}_{\mathsf{TKGen}}(\cdot,\cdot)$, and an encryption key generation oracle $\mathcal{O}_{\mathsf{EKGen}}(\cdot)$. The security definition of sIND-CPA in LME is based on the following experiment:

```
Experiment $\mathbf{Exp}^{\mathsf{sIND\text{-}CPA}}_{\mathcal{LME},\mathcal{A}}(1^\lambda)$
    $\mathbb{R}^* \leftarrow \mathcal{A}(1^\lambda)$;
    $\mathcal{D}_{id,\mathcal{R}} = \emptyset$;
    $(mpk, msk) \leftarrow \mathsf{Setup}(1^\lambda)$;
    $(m_0, m_1, \mathcal{S}_0, \mathcal{S}_1) \leftarrow \mathcal{A}^{\mathcal{O}}(mpk)$;
    $b \in \{0, 1\}$;
    $ek_{\mathcal{S}_b} \leftarrow \mathsf{EKGen}(msk, \mathcal{S}_b)$;
    $\hat{\mathcal{S}} = \mathcal{S}_b \cap \mathcal{S}_{1-b}$;
    $c^* \leftarrow \mathsf{Enc}(ek_{\mathcal{S}_b}, \hat{\mathcal{S}}, \mathbb{R}^*, m_b)$;
    $b' \leftarrow \mathcal{A}^{\mathcal{O}}(c^*)$;
    return 1 if $b = b'$ and
        $\mathcal{S}_b \cap \mathcal{S}_{1-b} \neq \emptyset$ and
        $\forall (id, \mathcal{R}) \in \mathcal{D}_{id,\mathcal{R}} : \mathcal{R} \not\models \mathbb{R}$.
```

```
Oracle $\mathcal{O}_{\mathsf{KeyGen}}(id, \mathcal{R})$
    $(pk_{id}, sk_{id}) \leftarrow \mathsf{KeyGen}(id)$;
    return $pk_{id}$.
Oracle $\mathcal{O}_{\mathsf{Corrupt}}(id, \mathcal{R})$
    $\mathcal{D}_{id,\mathcal{R}} \leftarrow \mathcal{D}_{id,\mathcal{R}} \cup \{id, \mathcal{R}\}$;
    $(pk_{id}, sk_{id}) \leftarrow \mathsf{KeyGen}(id)$;
    return $sk_{id}$.
Oracle $\mathcal{O}_{\mathsf{TKGen}}(pk_{id}, \mathcal{R})$
    $tk_{id} \leftarrow \mathsf{TKGen}(msk, pk_{id}, \mathcal{R})$;
    return $tk_{id}$.
Oracle $\mathcal{O}_{\mathsf{EKGen}}(\mathcal{S})$
    $ek_{\mathcal{S}} \leftarrow \mathsf{EKGen}(msk, \mathcal{S})$;
    return $ek_{\mathcal{S}}$.
```

An $\mathcal{LME}$ is said to be sIND-CPA secure if for any probabilistic polynomial-time adversary $\mathcal{A}$, the following advantage is negligible: $\mathbf{Adv}^{\mathsf{sIND\text{-}CPA}}_{\mathcal{LME},\mathcal{A}}(1^\lambda) = \left|\Pr[\mathbf{Exp}^{\mathsf{sIND\text{-}CPA}}_{\mathcal{LME},\mathcal{A}}(1^\lambda) = 1] - 1/2\right|$.

*Definition 3.5 (*EU-CMA *in $\mathcal{LME}$).* Let $\mathcal{O}$ denote a set of oracles: a key generation oracle $\mathcal{O}_{\mathsf{KeyGen}}(\cdot,\cdot)$, a corrupt oracle $\mathcal{O}_{\mathsf{Corrupt}}(\cdot,\cdot)$, a transformation key generation oracle $\mathcal{O}_{\mathsf{TKGen}}(\cdot,\cdot)$, an encryption key generation oracle $\mathcal{O}_{\mathsf{EKGen}}(\cdot)$, and an encryption oracle $\mathcal{O}_{\mathsf{Enc}}(\cdot,\cdot,\cdot,\cdot)$. The security definition of EU-CMA in LME is based on the following experiment:

An $\mathcal{LME}$ is said to be EU-CMA secure if for any probabilistic polynomial-time adversary $\mathcal{A}$, the following advantage is negligible: $\mathbf{Adv}^{\mathsf{EU\text{-}CMA}}_{\mathcal{LME},\mathcal{A}}(1^\lambda) = \Pr[\mathbf{Exp}^{\mathsf{EU\text{-}CMA}}_{\mathcal{LME},\mathcal{A}}(1^\lambda) = 1]$.

Remark. In our security models, $(id, \mathcal{R})$ and $(pk_{id}, \mathcal{R})$ are the bundled pairs, which means that one id only has a unique set of attributes $\mathcal{R}$. Hence, the oracles, $\mathcal{O}_{\mathsf{KeyGen}}(\cdot,\cdot)$, $\mathcal{O}_{\mathsf{Corrupt}}(\cdot,\cdot)$, and $\mathcal{O}_{\mathsf{TKGen}}(\cdot,\cdot)$, reject invalid queries when one id has been bound to different sets of attributes $\mathcal{R}$ and $\mathcal{R}'$ with $\mathcal{R} \neq \mathcal{R}'$.

```
Experiment $\mathbf{Exp}^{\text{EU-CMA}}_{\mathcal{LME},\mathcal{A}}(1^\lambda)$
    $\mathbb{S}^* \leftarrow \mathcal{A}(1^\lambda)$;
    $\mathcal{D}_\mathcal{S} = \emptyset$;
    $\mathcal{D}_c = \emptyset$;
    $(mpk, msk) \leftarrow \text{Setup}(1^\lambda)$;
    $c^* \leftarrow \mathcal{A}^{\mathcal{O}}(mpk)$;
    return 1 if $\text{Verify}(\mathbb{S}, c) = 1$, and
        $\forall \mathcal{S} \in \mathcal{D}_\mathcal{S} : \mathcal{S} \not\models \mathbb{S}$, and
        $\forall c \in \mathcal{D}_c : c \not\equiv c^*$

Oracle $\mathcal{O}_{\text{Enc}}(\hat{\mathcal{S}}, \mathbb{R}, m)$
    $c \leftarrow \text{EKGen}(msk, \hat{\mathcal{S}})$;
    $\mathcal{D}_c \leftarrow \mathcal{D}_c \cup \{c\}$;
    return $c$.

Oracle $\mathcal{O}_{\text{KeyGen}}(id, \mathcal{R})$
    $(pk_{id}, sk_{id}) \leftarrow \text{KeyGen}(id)$;
    return $pk_{id}$.

Oracle $\mathcal{O}_{\text{Corrupt}}(id, \mathcal{R})$
    $(pk_{id}, sk_{id}) \leftarrow \text{KeyGen}(id)$;
    return $sk_{id}$.

Oracle $\mathcal{O}_{\text{TKGen}}(pk_{id}, \mathcal{R})$
    $tk_{id} \leftarrow \text{TKGen}(msk, pk_{id}, \mathcal{R})$;
    return $tk_{id}$.

Oracle $\mathcal{O}_{\text{EKGen}}(\mathcal{S})$
    $\mathcal{D}_\mathcal{S} \leftarrow \mathcal{D}_\mathcal{S} \cup \{\mathcal{S}\}$;
    $ek_\mathcal{S} \leftarrow \text{EKGen}(msk, \mathcal{S})$;
    return $ek_\mathcal{S}$.
```

4 CLOUD-FOG IOT SYSTEM

We give the system model and threat model for our proposed cloud-fog IoT system. Specifically, we present the interactions between each entity, and analyse the security requirement of each entity to show several potential attacks.

4.1 System Model

Recall $\mathcal{LME}$ = {Setup, KeyGen, TKGen, EKGen, Enc, Verify, Transfer, Dec} as in Definition 3. Based on our system model, as shown in Fig. 3, the workflow of the cloud-fog IoT system consists of three phases: system initialization, data uploading, and data retrieving.

System Initialization: Fig. 4 shows the cloud-fog IoT initialization. This phase can be further classified into *system parameter initialization* (See ①), *decryption key initialization* (See ② and ③) and *encryption key initialization* (See ④).

- *System Parameter Initialization*: The KGC generates the master pubic and secret key pair (mpk, msk) by running the setup algorithm $\text{Setup}(1^\lambda)$, then distributes the master public key mpk to FNs and EDs. The master secret key msk is kept secret.
- *Decryption Key Initialization*: The receiver generates the public and secret key pair (pk_{id}, sk_{id}) by running the key generation algorithm $\text{KeyGen}(id)$, and sends the public key pk_{id} to the KGC. The secret key sk_{id} is kept secret. The KGC specifics an attribute set $\mathcal{R}$ of that receiver id and runs the transformation key generation algorithm $\text{TKGen}(msk, pk_{id}, \mathcal{R})$ to derive the transformation key tk_{id}. The KGC sends the transformation key tk_{id} to the FNs. Note that the decryption capability is the combination of the transformation key tk_{id} and the decryption key sk_{id}, where the transmission of the transformation key tk_{id} is via a public channel.
- *Encryption Key Initialization*: The KGC specifics an attribute set $\mathcal{S}$ of a sender and generates an encryption key $ek_\mathcal{S}$ by running the encryption key generation algorithm $\text{EKGen}(msk, \mathcal{S})$. The KGC sends the encryption key $ek_\mathcal{S}$ to that sender.

Data Uploading: Fig. 5 shows the data uploading phase. A sender aggregates data from surrounding areas or user inputs and runs the encryption algorithm $\text{Enc}(mpk, ek_\mathcal{S}, \mathbb{R}, m)$ to derive a ciphertext c to FNs (See ⑤). The FN forwards the ciphertext c to the other FN or the CSP depending on the purpose of the message (See ⑥).

Data Retrieving: Fig. 6 shows the phase of data retrieving in the cloud-fog IoT system. This phase can be further classified into *ciphertext retrieving* (See ⑦ and ⑧) and *data revealing* (See ⑨).

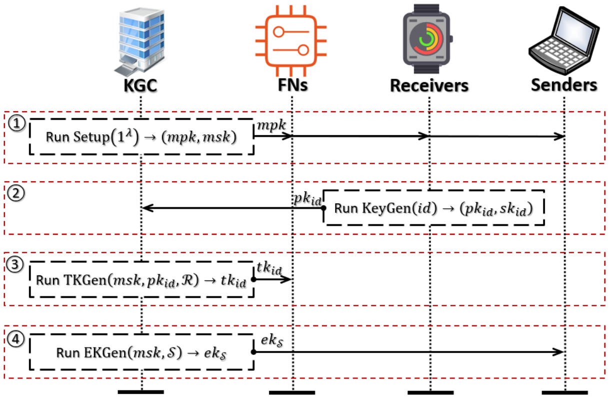

KGC
FNs
Receivers
Senders
①
Run Setup$(1^\lambda) \rightarrow (mpk, msk)$
mpk
②
pk_{id}
Run KeyGen$(id) \rightarrow (pk_{id}, sk_{id})$
③
Run TKGen$(msk, pk_{id}, \mathcal{R}) \rightarrow tk_{id}$
tk_{id}
④
Run EKGen$(msk, \mathcal{S}) \rightarrow ek_\mathcal{S}$
$ek_\mathcal{S}$

Figure 4: System Initialization in the Cloud-Fog IoT System

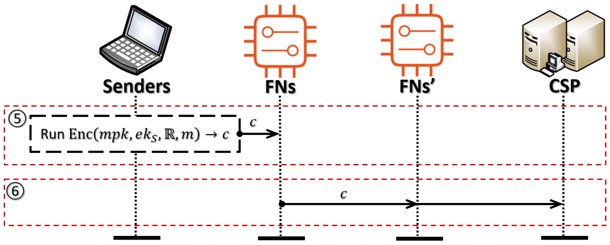


Figure 5: Data Uploading in the Cloud-Fog IoT System

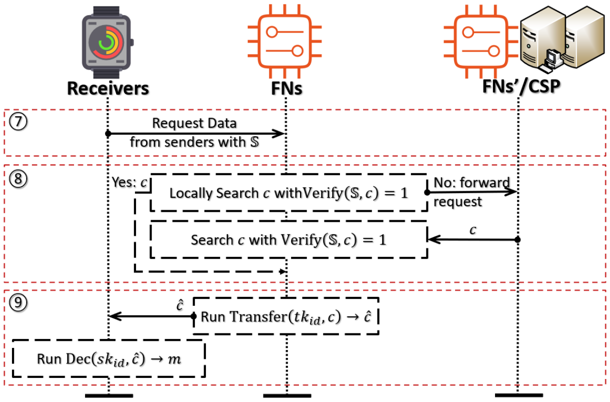

Receivers
FNs
FNs'/CSP
⑦
Request Data
from senders with $\mathbb{S}$
⑧
Yes: c
Locally Search c withVerify$(\mathbb{S}, c) = 1$
No: forward
request
Search c with Verify$(\mathbb{S}, c) = 1$
c
⑨
$\hat{c}$
Run Transfer$(tk_{id}, c) \rightarrow \hat{c}$
Run Dec$(sk_{id}, \hat{c}) \rightarrow m$

Figure 6: Data Retrieving in the Cloud-Fog IoT System

- *Ciphertext Retrieving*: A receiver specifies a sender's access policy $\mathbb{S}$ and sends $\mathbb{S}$ to FNs. FNs first search the local storage to find ciphertexts c with $\text{Verify}(\mathbb{S}, c) = 1$ if they exist; otherwise, FNs require data from the CSP (or the other FNs).
- *Data Revealing*: After finding/receiving ciphertexts, the FN runs the transformation algorithm $\text{Transfer}(tk_{id}, c)$ to generate transferred ciphertexts $\hat{c}$ and send $\hat{c}$ to receivers, who run the decryption algorithm $\text{Dec}(sk_{id}, \hat{c})$ to reveal message m.

4.2 Threat Model

The KGC is trusted, who initializes the system and issues encryption keys and transformation keys honestly. The CSP and FNs are semi-trusted, who follow our protocol but can launch passive attacks to learn any information beyond available (e.g., revealing messages from unauthorized ciphertexts and identifying senders). EDs are untrustworthy, who follow our protocol but can launch passive and active attacks to learn unauthorized information and impersonate others to share data (e.g., sending messages as unauthorized senders). Outsiders are untrusted, who can launch any attacks including collusion attacks, and impersonate any entity. In the following, we present the potential attacks in our cloud-fog IoT system. We refer the unauthorized party to any party without valid keys, including the CSP, FNs, outsiders, and EDs without valid decryption keys.

In our threat model, we consider the following passive attacks.

- *Release of message contents*: Any unauthorized party reveals plaintext information of transferred messages by observing all messages via the public channel.
- *Traffic analysis*: Any party except the message sender themselves reveals the sender's unique identifier or links two ciphertexts from the same sender.

Besides passive attacks, we consider the following active attacks.

- *Masquerade attack*: This attack can be launched by any sender, receiver, or outsider. Because outsider has neither encryption key nor decryption key, we focus on sender and receiver (who are stronger attackers) in addressing the masquerade attack.
  - The sender with an attribute set $\mathcal{S}_1$ impersonates another sender with an attribute set $\mathcal{S}$ $s.t.$ $\mathcal{S} \nsubseteq \mathcal{S}_1$; the sender may collude with unauthorized senders with attribute sets $\mathcal{S}_2, \mathcal{S}_3, ..., \mathcal{S}_n$ $s.t.$ $\mathcal{S} \subseteq \mathcal{S}_1 \cup \mathcal{S}_2 \cup \cdots \cup \mathcal{S}_n$ and $\forall i \in [n] : \mathcal{S} \nsubseteq \mathcal{S}_i$. Our EU-CMA security model captures this attack by offering an encryption key generation oracle which provides various encryption keys to simulate a sender multiple unauthorized senders.
  - The receiver with an attribute set $\mathcal{R}_1$ impersonates another receiver with an attribute set $\mathcal{R}$ $s.t.$ $\mathcal{R} \nsubseteq \mathcal{R}_1$; the receiver may collude with unauthorized receivers with attribute sets $\mathcal{R}_2, \mathcal{R}_3, ..., \mathcal{R}_n$ $s.t.$ $\mathcal{R} \subseteq \mathcal{R}_1 \cup \mathcal{R}_2 \cup \cdots \cup \mathcal{R}_n$ and $\forall i \in [n]$: $\mathcal{R} \nsubseteq \mathcal{R}_i$. Our sIND-CPA security model captures this attack by offering a corrupt oracle and a transformation key generation oracle which provides various keys to simulate a sender in colluding with multiple unauthorized receivers.
- *Modification attack*: This attack can be launched by any unauthorized party, who can intercept any message via the public channel, and modify these messages. An attacker may collude with other parties to launch the modification attack. Our sIND-CPA and EU-CMA security models capture this attack by allowing adversaries to get various messages via the public channel. The purpose of a modification attack is to generate a valid ciphertext from invalid parties, which breaks the sIND-CPA and EU-CMA security.

Remark. All the above attacks can be reduced to at least one of sIND-CPA and EU-CMA security. Hence, if there exists a probabilistic polynomial-time adversary that can break our scheme, we can then use this adversary to break sIND-CPA and EU-CMA as shown in Definition 4 and Definition 5. Therefore, the security of our proposed cloud-fog IoT system is established based on the security of LME with sIND-CPA and EU-CMA security models.

5 LIGHTWEIGHT MATCHMAKING ENCRYPTION

We propose a concrete construction of LME which is shown in Appendix A. Following is the high-level processes for achieving the important functionalities, including *lightweight decryption*, *bilateral access control* and *data source identification*, in our proposed scheme.

Lightweight decryption allows EDs to decrypt a ciphertext with the cost of one group exponentiation and one group multiplication as shown in the decryption algorithm in Appendix A. To support secure lightweight data decryption, we consider following four steps:

(1) Each ED executes the key generation algorithm KeyGen to generate a key pair (pk_{id}, sk_{id}), then outsources pk_{id} to the KGC and keeps sk_{id} secure.
(2) Based on pk_{id}, the KGC runs the transformation key generation algorithm TKGen to derive a transformation key tk_{id} and sends tk_{id} to the corresponding FN.
(3) The FN operates the transfer algorithm Transfer for removing an attribute-based component in ciphertext to derive a transformed ciphertext $\hat{c}$, and forwards to the corresponding ED. Note that $\hat{c}$ is secure since sk_{id} is unknown to the ED.
(4) The ED executes the decryption algorithm Dec to reveal the message m. The cost in the ED is only one group exponentiation and one group multiplication.

Therefore, we have lightweight decryption due to the efficient decryption mechanism.

Bilateral access control means *sender access control* and *receiver access control* simultaneously.

- *Sender access control* is based on the encryption key. The encryption key is issued by the KGC. By processing the verification algorithm Verify, any ciphertext from unauthorized senders will be identified and discarded to reduce the workload of computational and communication resources. Our scheme applies a collision-resistant function to assure the ciphertext integrity, which prevents unauthorized parties from modifying the ciphertext to launch impersonate attacks.
- *Receiver access control* is based on the ABE [23]. ABE ensures that the ciphertext associated with an access policy $\mathbb{R}$ only can be decrypted by authorized receivers who have a secret key specified a set of attribute $\mathcal{R}$, $s.t.$ $\mathcal{R} \models \mathbb{R}$. Note that, in ABE, each secret key associated with a unique randomness, any secret key with more than one randomnesses will be invalid for preventing collision attacks.

Outsourced ciphertext identification allows the semi-trusted FN to operate data source identification. To support secure identification, we consider the following two steps:

(1) Each ED uploads an access policy $\mathbb{S}$ to the corresponding FN, where $\mathbb{S}$ specifies an access policy for desirable senders.
(2) FNs run the verification algorithm Verify to identify the data source associated with an attribute set $\mathcal{S}$ $s.t.$ $\mathcal{S} \models \mathbb{S}$. FNs then forward valid ciphertexts to the corresponding ED.

Therefore, we have outsourced data source identification since the workload of ciphertext identification immigrates from EDs to FNs.

5.1 Security Analysis

Theorem 5.1. *If the decisional* q-1 *assumption holds, then all probabilistic polynomial-time adversaries have a negligible advantage in breaking* sIND-CPA *security of our scheme.*

Theorem 5.2. *If the computational* q-1 *assumption holds, then all probabilistic polynomial-time adversaries have a negligible advantage in breaking* EU-CMA *security of our scheme.*

The detailed security proofs of **Theorem 1** and **Theorem 2** are omitted to conserve space. Please contact the authors for them.

5.2 Efficiency Analysis

To the best of our knowledge, no formal solution is introduced for matchmaking encryption with lightweight decryption in an attribute-based setting. Hence, in this section, we give a comparison between the most relevant solutions, including ABE, DP-ABE, ACE, MIBE, MABE, and ours, as shown in Table 1. There is a variety of extensions of ABE. One of the most relevant to our work is the outsourced ABE. The seminal work of the outsourced ABE [16] only considered the basic access control and cannot handle complex policy as demonstrated in DP-ABE [4, 29] which provides key-policy and ciphertext-policy access control simultaneously. DP-ABE [4] cannot provide lightweight decryption and fails to support resource-constrained IoT devices. The ACE scheme [11] has been used to achieve complex access control policy "no read-up, no write-down", which is different from the attribute-based access control and also requires a third party online to supervise all communications between senders and receivers. The existing solution for matchmaking encryption [3] is impractical since the access policy is in an identity-based setting. Hence, only lightweight DP-ABE [29] has comparable functionality to ours, which provides the bilateral access control in ciphertexts (cf. entities) and outsourced decryption.

Theoretical Complexity. We give comparisons in terms of computational complexity and space complexity among [1, 3, 4, 11, 16, 29] and ours. These comparisons illustrate that our scheme has superior performance to the most relevant solutions.

Table 2 shows the comparison about computational complexity. Our scheme has fixed time to set up the system, which is better than AI09 [4] and DHO16 [11]. By applying the technology in the lightweight decryption, our scheme has the constant time to key generation, and the cost of transformation key generation is based on the attribute set of receivers, which is comparable to the relevant solutions with the outsourced property. Our scheme has bilateral access control in an attribute-based setting, which takes the cost depending on the attribute set of senders. Because the verification process in AFNV19 [3] cannot be outsourced, we consider the cost in DHO16 [11] and ours. DHO16 [11] has "no read-up, no write down" policy incurring the cost based on the number of system users, which is different from ours with the attribute-based setting that takes the cost depending on the policy of senders. Similar to the other schemes with lightweight decryption, our scheme only has a fixed cost to process ciphertext decryption, and the heavy workload of ciphertext transformation is outsourced to a third party (e.g., edge devices). Hence, our scheme has comparable computational complexity to the related bilateral access control solutions.

Table 3 shows the comparison about space complexity. Our scheme applies the large universe ABE [23] to achieve the constant-size public parameter in the standard model, which is quite different from other solutions [1, 3, 4, 11, 16] by applying a random oracle, but the random oracle is an ideal model and nonexistent in the real world. To support resource-constrained devices, our scheme only has the constant-size user key to process ciphertext decryption and constant-size transformed ciphertext to reduce the storage requirement, which has much better performance than other schemes without considering lightweight data process. Hence, our scheme has comparable space complexity to the existing solutions related to solutions with lightweight data decryption, and much better than the schemes without considering lightweight data decryption.

Based on the above analysis, only the outsourced DP-ABE as XLD+19 [29] has comparable performance in terms of functionality (e.g., bilateral access control in ciphertexts) and theoretical complexity (e.g., lightweight decryption progress) to ours. In the next, we focus on the experimental simulation between XLD+19 [29] and ours since [29] applies DP-ABE which can be redacted to realize the same functionalities in ours.

Experimental Simulation. Our experimental simulation was performed on a personal computer equipped with 64-bit Windows 10, 3.60GHz Intel(R) Core(TM) i7-4790 CPU and 24GB memory, and ciphertext decryption process was simulated on MI 5s running in Android 10 with Quad-core Max 2.15 GHz Snapdragon 821 and 4 GB 2400 MHz LPDDR4. The implementation is based on the Type A elliptic curve with around 80-bit security from the standard parameters in "a.properties" via JPBC library. Hence, in our implementation, p is a 160-bit prime number, and elements in $\mathbb{G}$ and $\mathbb{G}_T$ are 512 bits and 1024 bits, respectively. The experimental performance is demonstrated in Fig. 7 and Fig. 8.

Fig. 7 presents algorithm running time. Overall, our solution has comparable performance in the term of computation compared to XLD+19, and better performance in the transformation key generation algorithm and the ciphertext transformation algorithm.

Fig. 7a gives the running time about the system setup versus the attribute universe. The time cost is irrelevant to attributes in XLD+19 and ours since the constant-size parameter ABE is the basic construction of both schemes. Fig. 7b illustrates the running time of the key generation versus the number of attributes. The time cost is irrelevant to the size of attributes in XLD+19 and ours since each user only requires to generate a key pair for processing lightweight decryption. Fig. 7c shows the running time about the transformation key generation versus the policy size. Our scheme has much better performance than XLD+19 since the dual-policy control in XLD+19 has more cost. Fig. 7d displays the running time of the encryption key generation versus the number of attributes. Because our scheme focuses on bilateral access control rather than unilateral access control in XLD+19, we only provide the performance of ours. The running time is growth linearly to the number of attributes. Fig. 7e gives the running time about the encryption versus the number of attributes. Our scheme has a similar performance to that of XLD+19. Fig. 7f illustrates the running time about the verification versus

Table 1: Functionality Comparison

	Functionality						
	Type of Scheme	***Ct. Iden.***	***Snd. Anon.***	***Out. Iden.***	***Lig. Dec.***	***Security Model***	***Access Policy***
GHW11 [16]	*ABE*	✗	✓	✗	✓	*Random Oracle*	*Unilateral LSSS*
AC17 [1]	*ABE*	✗	✓	✗	✗	*Random Oracle*	*Unilateral LSSS*
AI09 [4]	*DP-ABE*	✗	✓	✗	✗	*Standard Model*	*Unilateral LSSS*
XLD+19 [29]	*DP-ABE*	✗	✓	✗	✓	*Standard Model*	*Unilateral LSSS*
DHO16 [11]	*ACE*	✓	✗	✗	✗	*Standard Model*	*No Read-Up, No Write-Down*
AFNV19 [3]	*MIBE*	✓	✗	✗	✗	*Random Oracle*	*Bilateral ID-Based Control*
Ours	*MABE*	✓	✓	✓	✓	*Standard Model*	*Bilateral LSSS*

Ct. Iden. (Ciphertext Identification): The quality of the ciphertext generator can be identified without revealing messages.
Snd. Anon. (Sender Anonymity): The ciphertexts do not leak the personal information of the sender.
Out. Iden. (Outsourced Verification): The workload of ciphertext identification can be outsourced to a third party.
Lig. Dec. (Lightweight Decryption): The cost of data decryption is lightweight and can be operated in the resource-constrained device.
LSSS: An access policy has a fine-grained access control as shown in Section 3.3.

Table 2: Computational Complexity Comparison

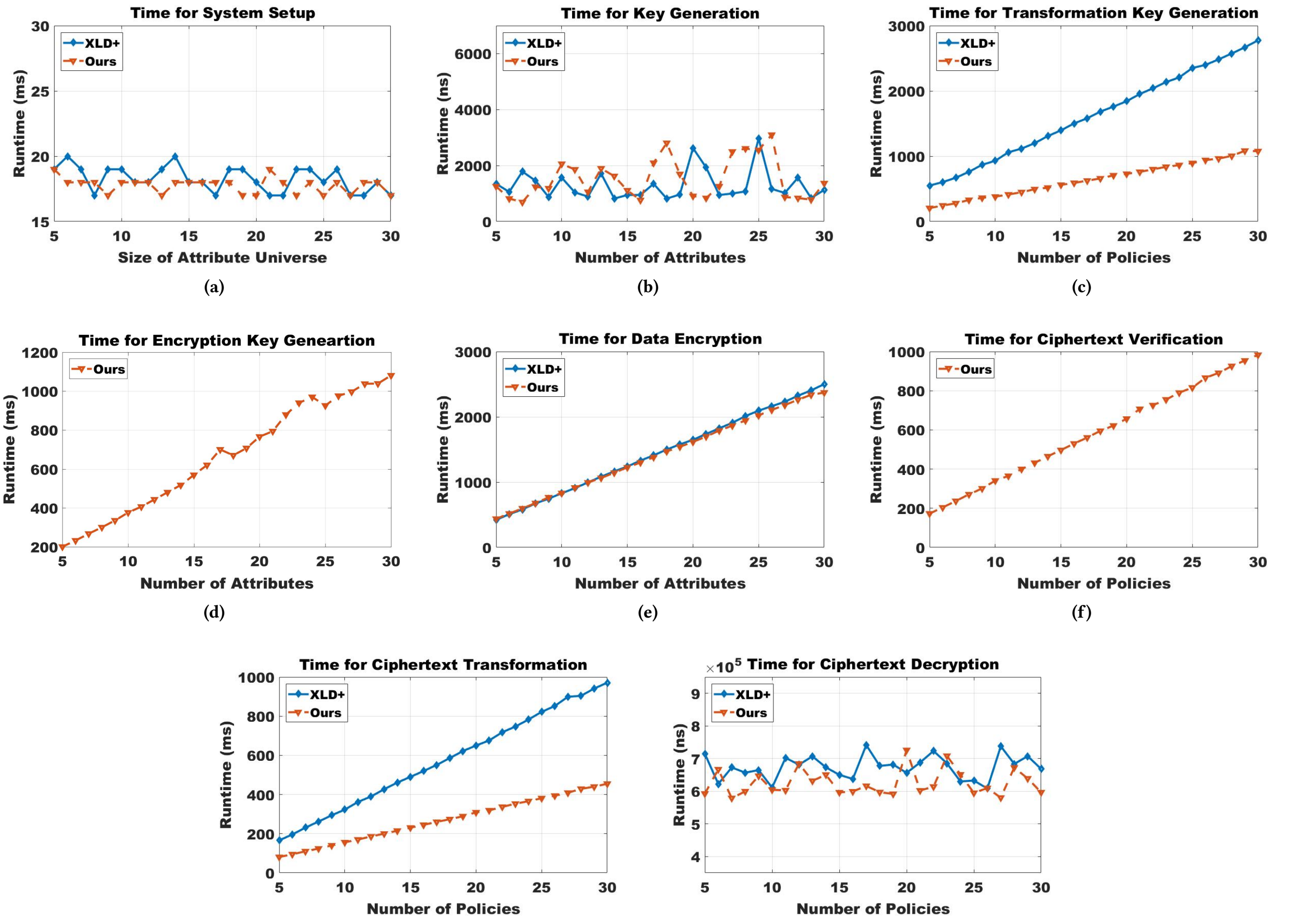
	Computational Complexity							
	Setup	***KeyGen***	***TKGen***	***EKGen***	***Enc***	***Verify***	***Transfer***	***Dec***
GHW11 [16]	$O(1)$	$O(1)$	$O(\mathbb{R})$	*N/A*	$O(\mathcal{R})$	*N/A*	$O(\mathbb{R})$	$O(1)$
AC17 [1]	$O(1)$	$O(\mathcal{R})$	*N/A*	*N/A*	$O(\mathbb{R})$	*N/A*	*N/A*	$O(\mathcal{R})$
AI09 [4]	$O(max_{\Omega,\mathcal{P}})$	$O(\mathcal{R}+\mathbb{R}')$	*N/A*	*N/A*	$O(\mathbb{R}+\mathcal{R}')$	*N/A*	*N/A*	$O(\mathcal{R}+\mathbb{R}')$
XLD+19 [29]	$O(1)$	$O(1)$	$O(\mathcal{R}+\mathbb{R}')$	*N/A*	$O(\mathbb{R}+\mathcal{R}')$	*N/A*	$O(\mathcal{R}+\mathbb{R}')$	$O(1)$
DHO16 [11]	$O(n)$	$O(n)$	*N/A*	*N/A*	$O(n)$	$O(n)$	$O(n)$	$O(1)$
AFNV19 [3]	$O(1)$	$O(1)$	*N/A*	$O(1)$	$O(1)$	*N/A*	*N/A*	$O(1)$
Ours	$O(1)$	$O(1)$	$O(\mathcal{R})$	$O(\mathcal{S})$	$O(\mathcal{S}+\mathbb{R})$	$O(\mathbb{S})$	$O(\mathbb{R})$	$O(1)$

$\mathbb{R}$: The access policy of receivers $\mathcal{R}$: The attribute set of receivers *N/A: Not applicable*
$\mathbb{S}$: The access policy of senders $\mathcal{S}$: The attribute set of senders
$max_{\Omega,\mathcal{P}}$: The bounded number of attributes or policies for describing a key or ciphertext.
n: The number of senders/receivers specified by the policy as the definition in ACE.

Table 3: Space Complexity Comparison

	Space Complexity					
	Public Parameter	***User Key***	***Transformation Key***	***Encryption Key***	***Ciphertext***	***Transformed Ciphertext***
GHW11 [16]	$O(1)$	$O(1)$	$O(\mathbb{R})$	*N/A*	$O(\mathbb{R})$	$O(1)$
AC17 [1]	$O(1)$	$O(\mathcal{R})$	*N/A*	*N/A*	$O(\mathcal{R})$	*N/A*
AI09 [4]	$O(max_{\Omega,\mathcal{P}})$	$O(\mathcal{R}+\mathbb{R}')$	*N/A*	*N/A*	$O(\mathbb{R}+\mathcal{R}')$	*N/A*
XLD+19 [29]	$O(1)$	$O(1)$	$O(\mathcal{R}+\mathbb{R}')$	*N/A*	$O(\mathcal{R}+\mathbb{R}')$	$O(1)$
DHO16 [11]	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(2^n)$	*N/A*
AFNV19 [3]	$O(1)$	$O(1)$	*N/A*	$O(1)$	$O(1)$	*N/A*
Ours	$O(1)$	$O(1)$	$O(\mathcal{R})$	$O(\mathcal{S})$	$O(\mathcal{S}+\mathbb{R})$	$O(1)$

the size of policies. Because XLD+19 does not support outsourced ciphertext identification, we only provide the performance in ours. The running time is growth linearly to the size of policies. Fig. 7g shows the running time about the ciphertext transformation versus the size of policies. Because the verification process can be outsourced, our scheme has a much better performance than XLD+19. Fig. 7h displays the running time about the ciphertext decryption versus the size of policies. Because of similar technology for lightweight decryption, our scheme has a similar performance to that of XLD+19. Note that Fig. 7b, Fig. 7h, Fig. 7g and Fig. 7f demonstrate the computational costs in a receiver, a sender, an edge server and a fog node, respectively. We find that the costs of the receiver and the sender are in the nanosecond order of magnitude for adopting to resource-constrained EDs.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

Figure 7: Experimental Performances about Algorithm Running Time

Fig. 8 presents the cost of the data storage. Overall, our solution has comparable performance in the term of storage to XLD+19, and the shorter space of the transformation key.

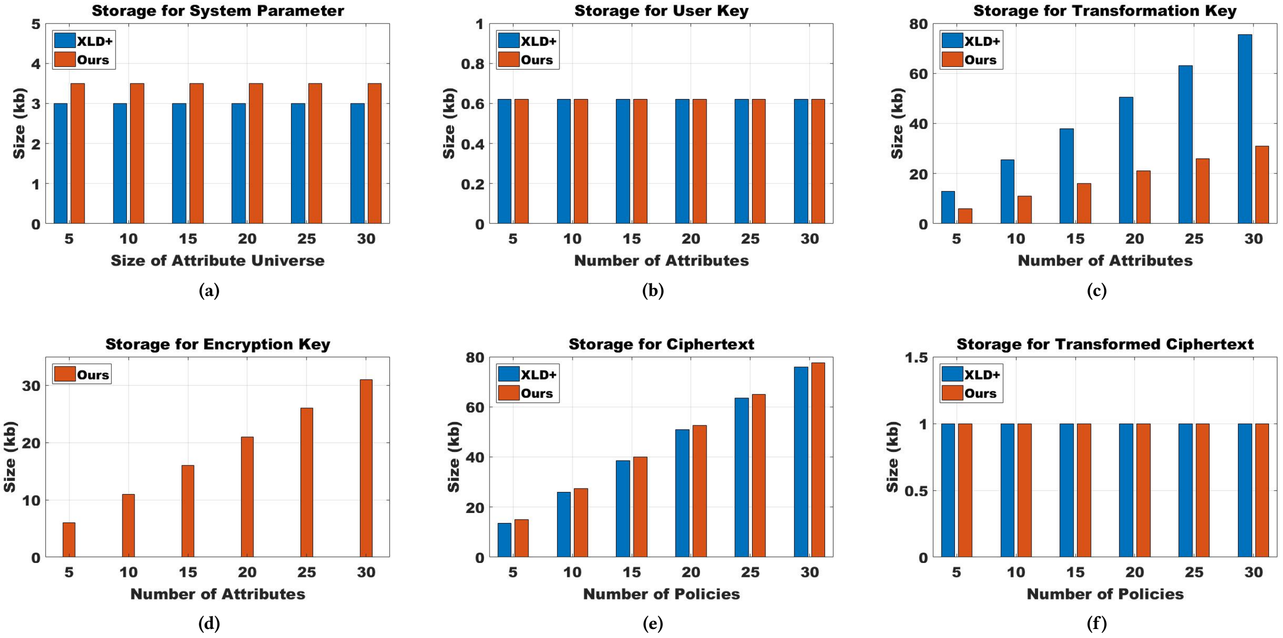
Fig. 8a gives the data storage about the system parameter versus the attribute universe. Our scheme has large storage since it requires one more $\mathbb{G}$ element to initialize the system parameter. Fig. 8b illustrates the data storage about the user key versus the number of attributes. Our scheme has the same data storage as XLD+19. Fig. 8c shows the data storage about the transformation key versus the number of attributes. XLD+19 requires double data storage as ours to generate the transformation key. Fig. 8d gives the data storage about the encryption key versus the number of attributes. Because the encryption key is for realizing the bilateral access control, we focus on the data storage in our scheme, which increases linearly with the number of attributes. Fig. 8e illustrates the data storage about the ciphertext versus policy size. Our scheme has similar performance to XLD+19. Fig. 8f shows the data storage about the transformed ciphertext versus the policy size. Our scheme has the same data storage as XLD+19. Note that Fig. 8b and Fig. 8f show the costs of data storage in the receiver and the sender. We believe that storage costs are lightweight (within 1KB) and suitable resource-constrained EDs.

6 CONCLUSION

In this paper, we presented a secure and efficient cloud-fog IoT system with bilateral access control at a fine-grained level. By applying cloud-fog computing, the heavy workloads of end-devices are relieved and various attacks are mitigated in our system. We believe our system is a promising solution for many large-scale IoT applications requiring data privacy and data source identification.

ACKNOWLEDGMENT

This work is supported in part by Natural Science Foundation of China (Grant Nos. 61872089, 61972094, 62032005), Science Foundation of Fujian Provincial Science and Technology Agency (2020J02016), the young talent promotion project of Fujian Science and Technology Association.



(a) (b) (c)

(d) (e) (f)

Figure 8: Experimental Results about Storage Overhead

REFERENCES

[1] Shashank Agrawal and Melissa Chase. 2017. FAME: Fast Attribute-based Message Encryption. In *CCS*. 665–682.

[2] Ala I. Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. 2015. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys and Tutorials* 17, 4 (2015), 2347–2376.

[3] Giuseppe Ateniese, Danilo Francati, David Nuñez, and Daniele Venturi. 2019. Match Me if You Can: Matchmaking Encryption and Its Applications. In *CRYPTO*. 701–731.

[4] Nuttapong Attrapadung and Hideki Imai. 2009. Dual-Policy Attribute Based Encryption. In *ACNS*, Vol. 5536. 168–185.

[5] John Bethencourt, Amit Sahai, and Brent Waters. 2007. Ciphertext-Policy Attribute-Based Encryption. In *IEEE S&P*. 321–334.

[6] Chiara Bodei, Pierpaolo Degano, Flemming Nielson, and Hanne Riis Nielson. 1999. Static Analysis of Processes for No and Read-Up nad No Write-Down. In *FoSSaCS*. 120–134.

[7] Dan Boneh and Matthew K. Franklin. 2001. Identity-Based Encryption from the Weil Pairing. In *CRYPTO*. 213–229.

[8] Flavio Bonomi, Rodolfo A. Milito, Jiang Zhu, and Sateesh Addepalli. 2012. Fog computing and its role in the internet of things. In *MCC*. 13–16.

[9] David Cash, Paul Grubbs, Jason Perry, and Thomas Ristenpart. 2016. Leakage-Abuse Attacks Against Searchable Encryption. *IACR Cryptology ePrint Archive* 2016 (2016), 718.

[10] David Cash, Paul Grubbs, Jason Perry, and Thomas Ristenpart. 2016. Leakage-Abuse Attacks Against Searchable Encryption. *IACR Cryptology ePrint Archive* 2016 (2016), 718.

[11] Ivan Damgård, Helene Haagh, and Claudio Orlandi. 2016. Access Control Encryption: Enforcing Information Flow with Cryptography. In *TCC*. 547–576.

[12] eugdpr.org. 2016. General Data Protection Regulation. https://eugdpr.org/the-process/how-did-we-get-here

[13] Taher El Gamal. 1984. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. In *CRYPTO*. 10–18.

[14] Luis Miguel Vaquero González and Luis Rodero-Merino. 2014. Finding your Way in the Fog: Towards a Comprehensive Definition of Fog Computing. *Computer Communication Review* 44, 5 (2014), 27–32.

[15] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. 2006. Attribute-based encryption for fine-grained access control of encrypted data. In *CCS*. 89–98.

[16] Matthew Green, Susan Hohenberger, and Brent Waters. 2011. Outsourcing the Decryption of ABE Ciphertexts. In *USENIX*.

[17] Junzuo Lai, Robert H. Deng, Chaowen Guan, and Jian Weng. 2013. Attribute-Based Encryption With Verifiable Outsourced Decryption. *IEEE Trans. Inf. Forensics Secur.* 8, 8 (2013), 1343–1354.

[18] Jianghua Liu, Jinhua Ma, Wei Wu, Xiaofeng Chen, Xinyi Huang, and Li Xu. 2017. Protecting Mobile Health Records in Cloud Computing: A Secure, Efficient, and Anonymous Design. *ACM Trans. Embedded Comput. Syst.* 16, 2 (2017), 57:1–57:20.

[19] Arslan Munir, Prasanna Kansakar, and Samee U. Khan. 2017. IFCIoT: Integrated Fog Cloud IoT: A novel architectural paradigm for the future Internet of Things. *IEEE Consumer Electronics Magazine* 6, 3 (2017), 74–82.

[20] Jianbing Ni, Kuan Zhang, Xiaodong Lin, and Xuemin Sherman Shen. 2018. Securing Fog Computing for Internet of Things Applications: Challenges and Solutions. *IEEE Communications Surveys and Tutorials* 20, 1 (2018), 601–628.

[21] Jianting Ning, Jia Xu, Kaitai Liang, Fan Zhang, and Ee-Chien Chang. 2019. Passive Attacks Against Searchable Encryption. *IEEE Trans. Information Forensics and Security* 14, 3 (2019), 789–802.

[22] Baodong Qin, Robert H. Deng, Shengli Liu, and Siqi Ma. 2015. Attribute-Based Encryption With Efficient Verifiable Outsourced Decryption. *IEEE Trans. Information Forensics and Security* 10, 7 (2015), 1384–1393.

[23] Yannis Rouselakis and Brent Waters. 2013. Practical constructions and new proof methods for large universe attribute-based encryption. In *CCS*. 463–474.

[24] Amit Sahai and Brent Waters. 2005. Fuzzy Identity-Based Encryption. In *EUROCRYPT*, Vol. 3494. 457–473.

[25] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. 2016. Edge Computing: Vision and Challenges. *IEEE Internet of Things Journal* 3, 5 (2016), 637–646.

[26] Jordan Shropshire. 2014. Extending the Cloud with Fog: Security Challenges & Opportunities. In *AMCIS*.

[27] Ivan Stojmenovic and Sheng Wen. 2014. The Fog Computing Paradigm: Scenarios and Security Issues. In *Proceedings of the 2014 Federated Conference on Computer Science and Information Systems, Warsaw, Poland, September 7-10, 2014*. 1–8.

[28] Zhiguo Wan, Jun-e Liu, and Robert H. Deng. 2012. HASBE: A Hierarchical Attribute-Based Solution for Flexible and Scalable Access Control in Cloud Computing. *IEEE Trans. Information Forensics and Security* 7, 2 (2012), 743–754.

[29] Shengmin Xu, Yingjiu Li, Robert H. Deng, Yinghui Zhang, Xiangyang Luo, and Ximeng Liu. 2019. Lightweight and Expressive Fine-grained Access Control for Healthcare Internet-of-Things. *IEEE Transactions on Cloud Computing* (2019).

[30] Shengmin Xu, Jianting Ning, Yingjiu Li, Yinghui Zhang, Guowen Xu, Xinyi Huang, and Robert Deng. 2020. Match in my way: Fine-grained bilateral access control for secure cloud-fog computing. *IEEE Transactions on Dependable and Secure Computing* (2020).

[31] Shengmin Xu, Guomin Yang, and Yi Mu. 2018. Revocable Attribute-Based Encryption with Decryption Key Exposure Resistance and Ciphertext Delegation. *Information Sciences* 479 (2018), 116–134.

[32] Yinghui Zhang, Xiaofeng Chen, Jin Li, Duncan S Wong, Hui Li, and Ilsun You. 2017. Ensuring attribute privacy protection and fast decryption for outsourced data security in mobile cloud computing. *Information Sciences* 379 (2017), 42–61.

[33] Yupeng Zhang, Jonathan Katz, and Charalampos Papamanthou. 2016. All Your Queries Are Belong to Us: The Power of File-Injection Attacks on Searchable Encryption. In *USENIX*. 707–720.

[34] Yinghui Zhang, Dong Zheng, and Robert H Deng. 2018. Security and privacy in smart health: Efficient policy-hiding attribute-based access control. *IEEE Internet of Things Journal* 5, 3 (2018), 2130–2145.

[35] Qingji Zheng, Shouhuai Xu, and Giuseppe Ateniese. 2014. VABKS: Verifiable attribute-based keyword search over outsourced encrypted data. In *INFOCOM*. 522–530.

A CONCRETE CONSTRUCTION

Let $\mathcal{LME}$ be a matchmaking attribute-based encryption with lightweight decryption with attribute universes Ω_{snd} and Ω_{rcv} that supports policies $\mathcal{P}_{\mathsf{snd}}$ and $\mathcal{P}_{\mathsf{rcv}}$, an identity space $\mathcal{I}$ and a message space $\mathcal{M}$. The concrete construction of $\mathcal{LME}$ is given below:

- $\mathsf{Setup}(1^\lambda) \rightarrow (mpk, msk)$: Run the bilinear pairing generator $\mathcal{G}(1^\lambda)$ to get the description $(e, \mathbb{G}, \mathbb{G}_T, g, p)$. Pick $w, v, u, h \in \mathbb{G}$, $\alpha, \beta \in \mathbb{Z}_p$, and a collision-resistant hash function $\mathcal{H} : \{0,1\}^* \rightarrow \mathbb{G}$. Output $mpk = (g, w, v, u, h, e(g,g)^\alpha, e(g,g)^\beta, \mathcal{H})$ and $msk = (\alpha, \beta)$.
- $\mathsf{KeyGen}(id) \rightarrow (pk_{id}, sk_{id})$: Pick $\gamma_{id} \in \mathbb{Z}_p$. Output $pk_{id} = g^{\gamma_{id}}$ and $sk_{id} = \gamma_{id}$.
- $\mathsf{TKGen}(msk, pk_{id}, \mathcal{R}) \rightarrow tk_{id}$: Parse $\mathcal{R} = (\mathcal{R}_1, \mathcal{R}_2, ..., \mathcal{R}_k)$. Pick $r, r_1, r_2, ..., r_k \in \mathbb{Z}_p$. Compute $tk_1 = pk_{id}^{\alpha} w^r, tk_2 = g^r$, $tk_{3,\tau} = g^{r_\tau}$ and $tk_{4,\tau} = (u^{\mathcal{R}_\tau} h)^{r_\tau} v^{-r}$. Output $tk_{id} = (\mathcal{R}, tk_1, tk_2, \{tk_{3,\tau}, tk_{4,\tau}\}_{\tau \in [k]})$.
- $\mathsf{EKGen}(msk, \mathcal{S}) \rightarrow ek_{\mathcal{S}}$: Parse $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_k)$. Pick $s, s_1, s_2, ..., s_k \in \mathbb{Z}_p$. Compute $ek_1 = g^\beta w^s, ek_2 = g^s, ek_{3,\tau} = g^{s_\tau}$ and $ek_{4,\tau} = (u^{\mathcal{S}_\tau} h)^{s_\tau} v^{-s}$. Output $ek_{\mathcal{S}} = (\mathcal{S}, ek_1, ek_2, \{ek_{3,\tau}, ek_{4,\tau}\}_{\tau \in [k]})$.
- $\mathsf{Enc}(ek_{\mathcal{S}}, \hat{\mathcal{S}}, \mathbb{R}, m) \rightarrow c$: Parse $\mathbb{R} = (\mathbb{M}, \rho)$, $\mathbb{M} \in \mathbb{Z}_p^{\ell \times n}$ and $\rho : [\ell] \rightarrow \mathbb{Z}_p$. Pick $\vec{x} = (\phi, x_2, ..., x_n)^\top \in \mathbb{Z}_p^{n \times 1}$. Compute $\vec{\lambda} = (\lambda_1, \lambda_2, ..., \lambda_\ell)^\top = \mathbb{M}\vec{x}$. Pick $t_1, t_2, ..., t_\ell \in \mathbb{Z}_p$. Compute $c_0 = m \cdot e(g,g)^{\alpha\phi}, c_1 = g^\phi, c_{2,\tau} = w^{\lambda_\tau} v^{t_\tau}, c_{3,\tau} = (u^{\rho(\tau)} h)^{-t_\tau}$ and $c_{4,\tau} = g^{t_\tau}$. Parse $\hat{\mathcal{S}} = (\hat{\mathcal{S}}_1, \hat{\mathcal{S}}_2, ..., \hat{\mathcal{S}}_{\hat{k}})$. Pick $\hat{s}, \hat{s}_1, \hat{s}_2, ..., \hat{s}_{\hat{k}}, \kappa \in \mathbb{Z}_p$. Compute $c_5 = ek_2 \cdot g^{\hat{s}} = g^{s+\hat{s}}, c_{6,\hat{\tau}} = ek_{3,\tau} \cdot g^{s_{\hat{\tau}}} = g^{s_\tau + s_{\hat{\tau}}}, c_{7,\hat{\tau}} = ek_{4,\tau} \cdot (u^{\mathcal{S}_\tau} h)^{s_{\hat{\tau}}} v^{-\hat{s}} = (u^{\mathcal{S}_\tau} h)^{(s_\tau + s_{\hat{\tau}})} v^{-(s+\hat{s})}$ and $c_8 = g^\kappa$. Let $\tilde{c}$ be $\tilde{c} = c_0 \| c_1 \| c_{2,1} \| ... \| c_{2,\ell} \| c_{3,1} \| ... \| c_{3,\ell} \| c_{4,1} \| ... \| c_{4,\ell} \| c_5 \| c_{6,1} \| ... \| c_{6,\hat{k}} \| c_{7,1} \| ... \| c_{7,\hat{k}} \| c_8$. Compute $c_9 = ek_1 \cdot w^{\hat{s}} \cdot \mathcal{H}(\tilde{c})^\kappa = g^\beta w^{s+\hat{s}} \mathcal{H}(\tilde{c})^\kappa$. Output $c = ((\hat{\mathcal{S}}, \mathbb{R}), c_0, c_1, \{c_{2,\tau}, c_{3,\tau}, c_{4,\tau}\}_{\tau \in [k]}, c_5, \{c_{6,\hat{\tau}}, c_{7,\hat{\tau}}\}_{\hat{\tau} \in [\hat{k}]}, c_8, c_9)$.
- $\mathsf{Verify}(\mathbb{S}, c) \rightarrow \{0,1\}$: Parse $\mathbb{S} = (\mathbb{N}, \pi)$, $\mathbb{N} \in \mathbb{Z}_p^{\ell \times n}$ and $\pi : [\ell] \rightarrow \mathbb{Z}_p$. Pick $\vec{y} = (1, y_2, ..., y_n)^\top \in \mathbb{Z}_p^{n \times 1}$. Compute $\vec{\mu} = (\mu_1, \mu_2, ..., \mu_\ell)^\top = \mathbb{N}\vec{y}$. Let $\mathcal{I}$ be $\mathcal{I} = \{i : \pi(i) \in \mathcal{S}\}$ for $\{\omega_i \in \mathbb{Z}_p\}_{i \in \mathcal{I}}$ $s.t.$ $\sum_{i \in \mathcal{I}} \omega_i \mathbb{N}_i = (1, 0, ..., 0)$. Check $e(g,g)^\beta \cdot \prod_{i \in \mathcal{I}} (e(c_5, w^{\mu_i} v) \cdot e(c_{6,\tau}, (u^{\pi(i)} h)^{-1}) \cdot e(c_{7,\tau}, g))^{\omega_i} \cdot e(c_8, \mathcal{H}(\tilde{c})) \overset{?}{=} e(c_9, g)$. Output 1 if the above formula is valid, otherwise, output 0.
- $\mathsf{Transfer}(tk_{id}, c) \rightarrow \hat{c}$: Parse $\mathcal{R} = (\mathcal{R}_1, \mathcal{R}_2, ..., \mathcal{R}_k)$, $\mathbb{R} = (\mathbb{M}, \rho)$, $\mathbb{M} \in \mathbb{Z}_p^{\ell \times n}$ and $\rho : [\ell] \rightarrow \mathbb{Z}_p$. Let $\mathcal{J}$ be $\mathcal{J} = \{j : \rho(j) \in \mathcal{R}\}$ for $\{\theta_j \in \mathbb{Z}_p\}_{j \in \mathcal{J}}$ $s.t.$ $\sum_{j \in \mathcal{J}} \theta_j \mathbb{M}_j = (1, 0, ..., 0)$. Compute $\hat{c}_0 = e(c_1, tk_1) / (\prod_{j \in \mathcal{J}} \left(e(c_{2,j}, tk_2) \cdot e(c_{3,j}, tk_{3,\tau}) \cdot e(c_{4,j}, tk_{4,\tau})\right)^{\theta_j}) = e(pk_{id}, g)^{\alpha\phi}$. Output $\hat{c} = (c_0, \hat{c}_0)$. To achieve unlinkability, this algorithm also executes the ciphertext re-randomization process. Pick $\gamma_1, \gamma_2 \in \mathbb{Z}_p$, output $\hat{c} = (c_0^{\gamma_1} \cdot e(g,g)^{\gamma_2}, \hat{c}_0^{\gamma_1} \cdot e(pk_{id}, g)^{\gamma_2})$ instead. We have $c_0^{\gamma_1} \cdot e(g,g)^{\gamma_2} = m \cdot e(g,g)^{\alpha\phi \cdot \gamma_1 + \gamma_2}$ and $\hat{c}_0^{\gamma_1} \cdot e(pk_{id}, g)^{\gamma_2}) = e(pk_{id}, g)^{\alpha\phi \cdot \gamma_1 + \gamma_2}$, where $(c_0, \hat{c}_0) \equiv (c_0^{\gamma_1} \cdot e(g,g)^{\gamma_2}, \hat{c}_0^{\gamma_1} \cdot e(pk_{id}, g)^{\gamma_2}))$ since the randomnesses γ_1 and γ_2 can be canceled in the decryption algorithm.
- $\mathsf{Dec}(sk_{id}, \hat{c}) \rightarrow m$: Output $c_0 / \hat{c}_0^{1/sk_{id}} = m$.