3-2022

Match in my way: Fine-grained bilateral access control for secure cloud-fog computing

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

Jianting NING
Fujian Normal University

Yingjiu LI
University of Oregon

Yinghui ZHANG
Xi'an Institute of Posts and Telecommunications

Guowen XU
University of Electronic Science and Technology of China

See next page for additional authors

Citation

XU, Shengmin; NING, Jianting; LI, Yingjiu; ZHANG, Yinghui; XU, Guowen; HUANG, Xinyi; and DENG, Robert H.. Match in my way: Fine-grained bilateral access control for secure cloud-fog computing. (2022). *IEEE Transactions on Dependable and Secure Computing*. 19, (2), 1064-1077.
Available at: https://ink.library.smu.edu.sg/sis_research/5176

Author

Shengmin XU, Jianting NING, Yingjiu LI, Yinghui ZHANG, Guowen XU, Xinyi HUANG, and Robert H. DENG

Match in My Way: Fine-Grained Bilateral Access Control for Secure Cloud-Fog Computing

Shengmin Xu, Jianting Ning, Yingjiu Li, *Member, IEEE,* Yinghui Zhang, *Member, IEEE,* Guowen Xu, *Student Member, IEEE,* Xinyi Huang, and Robert H. Deng, *Fellow, IEEE.*

***Abstract*—Cloud-fog computing is a novel paradigm to extend the functionality of cloud computing to provide a variety of on-demand data services via the edge network. Many cryptographic tools have been introduced to preserve data confidentiality against the untrustworthy network and cloud servers. However, how to efficiently identify and retrieve useful data from a large number of ciphertexts without a costly decryption mechanism remains a challenging problem. In this paper, we introduce a cloud-fog-device data sharing system (CFDS) with data confidentiality and data source identification simultaneously based on a new cryptographic primitive named matchmaking attribute-based encryption (MABE) by extending matchmaking encryption in CRYPTO'19. Our solution offers a secure fine-grained bilateral access control that includes (1) fine-grained sender access control; (2) fine-grained receiver access control; (3) sender privacy; and (4) performance optimization via outsourcing data source identification to fog nodes. We give the formal definition and security models of MABE, and present a concrete construction with formal security proofs. We also offer a detailed security analysis of our proposed CFDS against real-world security threats. The extensive comparison and experimental simulation demonstrate that, by immigrating heavy workload to fog nodes, our scheme has better functionalities and performances than the most related solutions.**

***Index Terms*—Cloud computing, fog computing, bilateral access control, fine-grained access control**

I. Introduction

CLOUD computing has been widely used to enable cloud users to store and share data to enjoy on-demand data services with elastic resources at low maintenance cost. However, there are still unsolved problems since many devices, especially resource-limited IoT devices, require customized and elaborate supports. Cloud-fog computing [1] as a novel paradigm has been recently introduced to extend the functionality of cloud computing. By empowering edge devices to carry out a substantial amount of computation, storage, and communication locally over the network, cloud-fog computing offers a number of advantages such as geographical distribution, mobility support, location-awareness, and low latency. According to the report from Statista [2], the market for worldwide cloud-fog computing will reach $13 billion in 2022.

To make full use of cloud-fog computing, Stojmenovic and Wen [3] introduced a cloud-fog-device architecture to provide various applications, including smart city, smart grid, intelligent transportation, and industrial automation. The architecture is a three-layer pyramid shown in Fig. 1, where (1) clouds in thousands of magnitude are managed to store data; (2) fog nodes in millions of magnitude are utilized to reduce the workload and bandwidth during data transmissions; and (3) end-devices in billions of magnitude are used to request and upload data. In this architecture, fog nodes play an important role in maintaining frequently used data in caches and collaborate for data intelligence. The purpose of fog nodes is for offering immediate and customized on-demand data services such as processing power and memory to reduce the costs of bandwidth during data transmissions. Taking the case of intelligent transportation for example, the cloud-fog-device architecture enables efficient data communications between the application server and vehicles. Specifically, the cloud receives the message from the application server and forwards to the corresponding fog nodes (e.g., roadside units), and fog nodes transfer the messages to end-devices (e.g., on-board units).

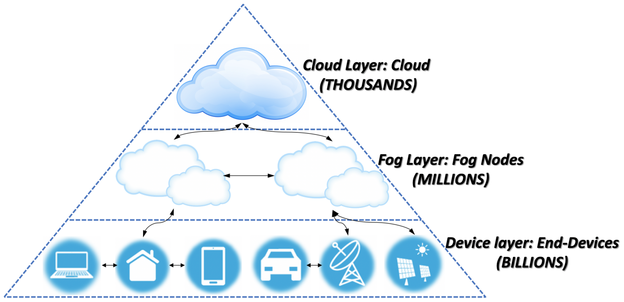


Fig. 1: Cloud-Fog-Device Architecture

Although cloud-fog-device architecture significantly reduces

S. Xu is with the Fujian Provincial Key Laboratory of Network Security and Cryptology, College of Mathematics and Informatics, Fujian Normal University, Fuzhou 350007, P.R. China, and the Information Systems Technology and Design, Singapore University of Technology and Design, Singapore 487372 (e-mail: shengmin_xu@sutd.edu.sg).

J. Ning is with the Fujian Provincial Key Laboratory of Network Security and Cryptology, College of Mathematics and Informatics, Fujian Normal University, Fuzhou 350007, P.R. China, and the School of Information Systems, Singapore Management University, Singapore 178902 (e-mail: jtning88@gmail.com).

Y. Li is with the Computer and Information Science Department, University of Oregon, Eugene, Oregon 97403, U.S. (e-mail: yingjiul@uoregon.edu).

Y. Zhang is with the School of Cyberspace Security, Xi'an University of Posts and Telecommunications, Xi'an 710121, P.R. China (email: yhzhaang@163.com).

G. Xu is with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Sichuan 611731, P.R. China (email: guowen.xu@foxmail.com).

X. Huang is with the Fujian Provincial Key Laboratory of Network Security and Cryptology, College of Mathematics and Informatics, Fujian Normal University, Fuzhou 350007, P.R. China (e-mail: xyhuang81@gmail.com).

R. H. Deng is with the School of Information Systems, Singapore Management University, Singapore 178902 (e-mail: robertdeng@smu.edu.sg).

X. Huang is the corresponding author.

the cost of data transmission and improves the performance of cloud computing, it also inherits many weaknesses of cloud computing. The data is outsourced to the untrustworthy parties, and senders (e.g., end-devices) cannot physically control their outsourced data. Moreover, the unreliable network environment also poses threats to data security. More importantly, the General Data Protection Regulation (GDPR) and similar data regulations demand secure data communications and processing in an untrustworthy environment. Such regulations and concerns would substantially limit the practical applications of cloud-fog computing unless the following three challenges are addressed.

The first challenge is *receiver access control for data confidentiality*. According to the collection limitation principle in GDPR, sensitive data, especially personal information (e.g., healthcare record, bank account information, and social security number), must be protected and cannot be abused by any unauthorized users. The second challenge is *sender access control for data source identification*. According to the data quality principle in GDPR, data source identification must be provided to ensure data accuracy. Besides, data source identification is a desirable property to prevent various attacks (e.g., denial-of-service attack and impersonate attack) by discarding invalid messages (e.g., spam email and data flooding). However, how to efficiently identify the data source from encryption data without a costly decryption mechanism is a challenging problem. The third challenge is *sender privacy for free-speech rights*. The data source identification somehow leaks sender privacy. However, leaking personal information is not a desirable property, especially in authoritarian countries without free-speech rights. Therefore, how to preserve sender privacy and also provide reasonable data source identification is a challenging problem. Unfortunately, there is no existing solution to address the above three challenges simultaneously.

A. Current Research States

In cloud-fog computing, the confidential data usually is shared with multiple users who have required attributes. To provide fine-grained access control, one of the most promising tools is attribute-based encryption (ABE) [4], [5], which enforces receiver access control that allows one ciphertext to be shared with multiple receivers. However, standard ABE does not support sender access control and hence cannot be directly applied in our cloud-fog computing setting.

ABE with keyword search (ABKS) [6] was introduced to allow receivers to search useful ciphertexts based on certain keywords without revealing messages. However, the keyword search incurs a heavy workload. Hence, many solutions have outsourced the workload of the search operation to a third party. To protect privacy about the searching keywords, it usually requires multiple rounds of interactions between the receiver and the third party to generate searchable queries. Unfortunately, many top-level works [7]–[9] pointed out that many ABKS schemes suffer from passive (e.g., leakage attack) and active attacks (e.g., file-injection attack), where the privacy of searchable pattern is easy to compromise. Therefore, there is no efficient and secure solution for fine-grained data source identification from encrypted data.

To provide receiver access control and sender access control simultaneously, matchmaking encryption [10] as a cryptographic primitive was introduced in CRYPTO'19. Matchmaking encryption enables the sender to specify the access structure for receivers to reveal the messages (data confidentiality, receiver access control), and it allows the receiver to determine whether ciphertexts are from approved senders (data source identification, sender access control). The possible application of matchmaking encryption as mentioned in [10] is two-party (e.g., spy and FBI) confidentially communications. However, current matchmaking encryption schemes have certain restrictions in real-world applications, especially cloud-fog applications. There are two matchmaking encryption schemes in the literature. The first one is a generic construction based on functional encryptions and zero-knowledge techniques, which incurs large overhead. Although the generic construction offers flexible access control, no concrete construction was provided for fine-grained access control. Besides, the privacy requirement of the proposed generic construction is too strong in cloud-fog computing, which requires data decryption to hide the reason for decryption failure (e.g., who fails to meet the policy). The second matchmaking encryption scheme is a concrete construction in the identity-based setting (matchmaking identity-based encryption, MIBE for short) without sender privacy. By exhaustively searching all possible identities, a receiver can always identify the sender. Neither of the two schemes support outsourced ciphertext identification due to the strong privacy requirement, which can be a performance bottleneck for resource-limited devices.

B. Contributions

In this paper, we first introduce a new primitive named matchmaking attribute-based encryption (MABE) based on attribute-based encryption and collision-resistant hash function. We then apply MABE to construct a secure fine-grained bilateral access control data sharing system in cloud-fog computing. Specifically, the paper makes the following contributions.

Fine-Grained Bilateral Access Control. We solve the above challenging problems by providing bilateral access control at a fine-grained level in an architecture as shown in Fig. 2 by extending the fine-grained access control in both sender-side and receiver-side. In fine-grained bilateral access control, end-devices require data from a set of authorized senders by defining the corresponding access structure (See ①) to fog nodes. Fog nodes retrieve ciphertexts from the cloud (See ②) if they do not have local copies, and then identify the ciphertexts from authorized senders to end-devices (See ③). The end-devices then reveal the underlying messages if they have a valid decryption key. In our proposed scheme, the senders are required to bind their attributes and specify their policies of authorized receivers in building ciphertexts. The receivers also define their policies for identifying desired ciphertexts from authorized senders and then decrypt ciphertexts if they have a valid decryption key. Therefore, many useless messages are discarded before costly data decryption so as to save the cost of bandwidth and computational resources.

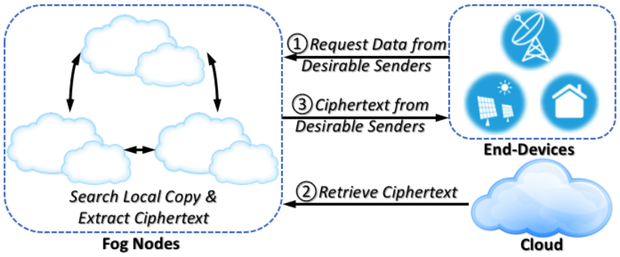

①Request Data from Desirable Senders
③Ciphertext from Desirable Senders
End-Devices
②Retrieve Ciphertext
Search Local Copy & Extract Ciphertext
Fog Nodes
Cloud

Fig. 2: Fine-Grained Bilateral Access Control

Outsourced Data Source Identification. To identify relevant data from a vast amount of ciphertexts, it is impractical if the receivers have to process every ciphertext, especially for resource-limited IoT devices. Our proposed system allows receivers to offload the heavy workloads of ciphertext identification to a semi-trusted third party (e.g., a fog node). A receiver only outsources an access structure to a fog node for identifying authorized senders. The fog node helps receivers to filter ciphertexts that do not satisfy this access structure. Note that MIBE cannot immigrate the heavy workload from the users to any third party due to the strict privacy requirement, and our ciphertext identification is at a fine-grained level rather than the coarse-grained level compared with MIBE.

Data Source Identification with Sender Privacy. As the data regulations such as GDPR require that data source should be identifiable to guarantee the data reliability, consequent to preserve receivers' interests. The existing MIBE solution binds the sender's identity and ciphertexts together to support the GDPR's data quality requirement but suffers from privacy leakage. To preserve sender privacy, our solution has the encryption key being associated with a set of attributes rather than a unique identity. The sender can choose a set of non-unique attributes to generate a ciphertext for preventing sender privacy. Moreover, by applying re-randomization technology, no third party can link two ciphertexts from the same sender.

Security and Efficiency. We provide a stronger security model than the MIBE [10] solution to thwart various attacks, including sender guessing attack, eavesdropping attack, impersonate attack, and collision attack. Note that MIBE suffers the sender guessing attack and impersonate attack. In MIBE, a valid receiver who meets the requirements specified by the sender can launch a brute force attack by trying every possible identity to find the sender's identity. This valid receiver can also change the underlying messages from the sender to forge new ciphertexts under the sender's identity. In our solution, we use a set of shared attributes rather than a unique identity as sender information in encryption against the sender guessing attack. We also apply a collision-resistant hash function to link ciphertexts to the sender's encryption key to prevent the impersonate attack. We provide efficiency analysis in terms of theoretical complexity and experimental data, which shows that, by immigrating heavy workload to fog nodes, our scheme enjoys superior functionalities and performances than the most relevant solutions in the literature.

C. Outline

Section II presents preliminaries for our proposed scheme. Section III defines system model and threat model of CFDS. The formal definition and security models of MABE are presented in Section IV. In Section V, we provide the detailed workflow of CFDS, a concrete construction of MABE with formal proofs, and security analysis of CFDS. In Section VI, we give the efficiency analysis, including theoretical complexity analysis and experimental performance. Section VII summarizes the related work and Section VIII concludes this paper.

II. Preliminaries

A. Notations

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 x and y are strings, $x \| y$ denotes the concatenation of x and y. If a and b are two ciphertexts from a probabilistic encryption algorithm, $a \equiv b$ means they have the same distribution, e.g., encrypting the same message with the different randomnesses. Besides, we give some frequently used notations in Table I.

TABLE I: Frequently Used Notations

Notation	*Description*
Ω_{snd}	*universe of the sender's attributes*
Ω_{rcv}	*universe of the receiver's attributes*
$\mathcal{P}_{\mathsf{snd}}$	*policies of the sender*
$\mathcal{P}_{\mathsf{rcv}}$	*policies of the receiver*
$\mathcal{S}, \mathcal{R}$	*attribute sets of a sender and a receiver*
$\mathbb{S}, \mathbb{R}$	*policies associated with a sender and a receiver*
$\mathcal{S} \models \mathbb{S}$	*a sender's attribute set satisfies a specific policy*
$\mathcal{R} \models \mathbb{R}$	*a receiver's attribute set satisfies a specific policy*

B. Bilinear Map

Definition 1 (Bilinear Map)**.** *Let $\mathbb{G}$ and $\mathbb{G}_T$ be two cyclic multiplicative groups of some prime order p and g be a generator of $\mathbb{G}$. We call e a bilinear map if $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is a map with the following properties:*

- *Bilinear: $e(g^a, g^b) = e(g, g)^{ab}$ for all $a, b \in \mathbb{Z}_p$.*
- *Non-degenerate: $e(g, g) \neq 1$.*
- *Computable: it is efficient to compute $e(g_1, g_2)$ for all $g_1, g_2 \in \mathbb{G}$.*

C. Assumptions

The security of our concrete construction bases on two complexity assumptions called the decisional Bilinear Diffie-Hellman (BDH) assumption and computational BDH assumption. We recall the formal definition of them as follows:

Definition 2 (Decisional BDH Assumption)**.** *Let $a, b, c, z \in \mathbb{Z}_p$ and g be a generator of bilinear group $\mathbb{G}$ of prime order p. The decisional BDH assumption is that no probabilistic polynomial-time algorithm can distinguish the tuple $(A = g^a, B = g^b, C = g^c, D = e(g, g)^{abc})$ from the tuple $(A = g^a, B = g^b, C = g^c, Z = e(g, g)^z)$ with more than a negligible advantage.*

Definition 3 (Computational BDH Assumption)**.** *Let $a, b, c \in \mathbb{Z}_p$ and g be a generator of bilinear group $\mathbb{G}$ of prime order* *p. The computational BDH assumption is that no probabilistic polynomial-time algorithm can compute the term* $e(g,g)^{abc}$ *from the tuple* $(A = g^a, B = g^b, C = g^c)$ *with more than a negligible advantage.*

D. Linear Secret Sharing Scheme

Definition 4 (Linear Secret Sharing Scheme (LSSS) [11])**.** *An LSSS over a set of parties* $\mathcal{P}$ *is called linear (over* $\mathbb{Z}_p$*) if*

1) *The shares for each party from a vector over* $\mathbb{Z}_p$.
2) *There exists a matrix* $\mathbb{M}$ *called share generating matrix with* ℓ *rows and n columns. For all* $i \in [\ell]$*, the* i^{th} *row of* $\mathbb{M}$ *is labeled with a party name* $x_i \in \mathcal{P}$*. When we consider the column vector* $\vec{v} = (s, r_2, ..., r_n)$*, where* $s \in \mathbb{Z}_p$ *is the secret to be shared, and* $r_2, ..., r_n \in \mathbb{Z}_p$ *are randomly chosen, then* $\mathbb{M}\vec{v}$ *is the vector of* ℓ *shares of the secret s. The share* $\mathbb{M}_i v_i$ *belongs to party* x_i*, where* $\mathbb{M}_i$ *denotes the* i^{th} *row of the matrix* $\mathbb{M}$ *and* v_i *is the* i^{th} *term in* $\vec{v}$*.*

III. System Model and Threat Model

In this section, we introduce the system model and threat model of our proposed CFDS. We define various entities and describe their interactions in our system model. We analysis the security requirement of each entity and present some possible attacks in our threat model.

A. System Model

Fig. 3 presents the system model of our proposed CFDS. Our system only requests a secure channel (as the imaginary lines in Fig. 3) to distribute secret keys. Our CFDS includes four types of entities: a key generation center (KGC), a CSP, fog nodes (FNs) and end-devices (EDs). The detailed description of characteristics and functionalities of each entity is given below:

- The KGC is responsible for initializing the system parameters and distribute them to all the entities. Besides, the KGC generates encryption keys and decryption keys based on the properties of the end-devices and distributes these keys to them via the secure channel (See ①).
- EDs could be resource-limited IoT devices. In our CFDS, EDs have the following responsibilities:
  1) EDs require data generated by the desirable senders (e.g., device owner remotely controls EDs or communication to other EDs) (See ②).
  2) EDs aggregate data from their surrounding environment, encrypt the data, and outsource the encrypted data to FNs. (See ③).
- FNs are edge servers and are responsible for the following:
  1) FNs act as caches to store frequently used data and information with the short-term purposes such as the data for communicating with other EDs.
  2) FNs forward the data with long-term purposes, such as historical records, to the CSP.
  3) After receiving data queries from EDs (See ②), FNs first search the local storage and interacts with other FNs (See ③). If no answers to the queries can be found, FNs request answers from the CSP (See ④).

  Note that although EDs are allowed to communicate with the CSP directly, it takes more resources since FNs are closer to EDs in the real-world scenarios (e.g., RSU closes to vehicles and gateway closes to smart devices).
- The CSP is a remote server which has a vast amount of storage to accommodate the data and also share the encrypted data to FNs via the public channel (See ④).

Remark. Our CFDS employs MABE to provide fine-grained bilateral access control for data sharing. Specifically, the KGC initializes MABE system and distribute encryption keys and decryption keys to EDs. For secure communication via public channels, each ED uses its encryption key and attributes of authorized receivers to encrypt data. Because the underlying message of ciphertext is unknown before decryption, the receivers can only receive the data from desirable senders and discard some useless data by identifying the encryption key in ciphertexts to reduce costs. For a more detailed description please refer to the workflow of CFDS in Section V.

B. Threat Model

We assume the KGC is a fully trusted entity. The KGC generates the system parameters and issues encryption keys and decryption keys to other entities via the secure channel. CSP and FNs are semi-trusted, who faithfully carries out system operations but may launch any passive attacks. EDs are untrustworthy, who can launch any attacks. As the receiver, an ED may try to decrypt any unauthorized ciphertexts. As the sender, an ED may try to pretend any unauthorized senders from generating messages to others. For simplicity, let unauthorized parties denote the collusive parties among CSP, FNs, and EDs without the valid decryption and encryption keys. In the following, we summarize the possible attacks in our CFDS system.

Sender guessing attack: The encryption key contains the sender's attributes, e.g., name, identity, gender, title, occupation and so on. The sender is allowed to choose any of them to generate the ciphertexts. The sender guessing attack means that an untrusted entity tries to know who is the sender or to link two ciphertexts, without personal identifiers (e.g., user identity and unique attributes), from the same sender.

Eavesdropping attack: An unauthorized party may eavesdrop the messages in the public channels, and try to learn the sensitive information from the unauthorized ciphertexts.

Impersonate attack: Impersonate attack means that any party may impersonate the encryption key with unauthorized attributes or generate the ciphertexts by attaching unauthorized sender's attributes to misleading the receivers. Besides, by obtaining a valid ciphertext, the unauthorized parties may try to replace or modify the underlying message to impersonate the corresponding sender.

Collusion attack: The unauthorized parties can get together to launch the above attacks. For example, unauthorized users combine multiple decryption keys to decrypt unauthorized ciphertexts, exchange encryption keys to generate the ciphertext without authorizing the sender's attributes.

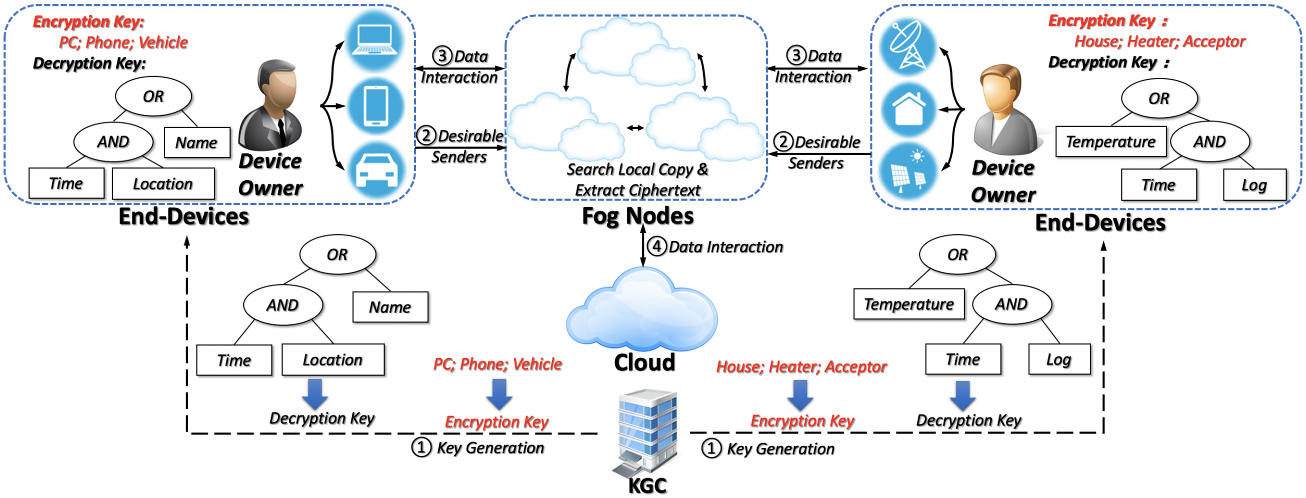


Fig. 3: System Model: Cloud-Fog-Device Data Sharing System

IV. Definition of MABE

In this section, we give the formal definition of MABE, which covers the core algorithms in the system model. Then, we propose the security model, which includes all possible attacks in the threat models.

A. Formal Definition

Definition 5 (Matchmaking Attribute-Based Encryption)**.** *A matchmaking attribute-based encryption $\mathcal{MABE}$ with the attribute universes Ω_{snd} and Ω_{rcv} that support the policies $\mathcal{P}_{\mathsf{snd}}$ and $\mathcal{P}_{\mathsf{rcv}}$ and message space $\mathcal{M}$ involves five types of entities: a KGC, senders, receivers, fog nodes, and a cloud service provider, and consists of the following six algorithms:*

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

$\mathsf{EKGen}(msk, \mathcal{S}) \rightarrow ek$*: The encryption key generation algorithm is run by the KGC. It takes the master secret key msk, and the set of the sender's attributes $\mathcal{S} \in \Omega_{\mathsf{snd}}$ as input, and outputs the encryption key ek.*

$\mathsf{DKGen}(msk, \mathbb{R}) \rightarrow dk$*: The decryption key generation algorithm is run by the KGC. It takes the master secret key msk and the policy of the receiver $\mathbb{R} \in \mathcal{P}_{\mathsf{rcv}}$ as input, and outputs the decryption key dk.*

$\mathsf{Enc}(ek, \mathcal{R}, \mathcal{S}', m) \rightarrow c$*: The encryption algorithm is run by the sender. It takes the encryption key ek, the set of the receiver's attributes $\mathcal{R} \in \Omega_{\mathsf{rcv}}$, the set of the sender's attributes $\mathcal{S}' \in \Omega_{\mathsf{snd}}$ and a message $m \in \mathcal{M}$ as input, and outputs a ciphertext c. We require that the set of the sender's attributes $\mathcal{S}'$ be a subset of the set of the sender's attributes $\mathcal{S}$ associated with encryption key ek, i.e., $\mathcal{S}' \subseteq \mathcal{S}'$, to achieve privacy of senders.*

$\mathsf{Verify}(\mathbb{S}, c) \rightarrow \{0,1\}$*: The verification algorithm is run by the fog node. It takes the policy of the sender $\mathbb{S} \in \mathcal{P}_{\mathsf{snd}}$ and ciphertext c associated with a set of the sender's attributes $\mathcal{S} \in \Omega_{\mathsf{snd}}$ as input, and outputs a bit 1 if and only if $\mathcal{S} \models \mathbb{S}$; otherwise, outputs 0.*

$\mathsf{Dec}(dk, c) \rightarrow m$ or $\perp$*: The decryption algorithm is run by the receiver. It takes the decryption key dk associated with a policy of the receiver $\mathbb{R} \in \mathcal{P}_{\mathsf{rcv}}$ and the ciphertext c associated with a set of the receiver's attributes $\mathcal{R} \in \Omega_{\mathsf{rcv}}$ as input, and outputs the message m if and only if $\mathcal{R} \models \mathbb{R}$; otherwise, outputs an error symbol $\perp$.*

B. Security Models

In this subsection, we describe two security models. The first one is indistinguishability under a chosen plaintext attack (IND-CPA) in the random oracle model to cover sender guessing attack, collusion attack, and eavesdropping attack. The second one is existential unforgeability under a chosen message attack (EU-CMA) in the random oracle model to cover the replay attack and impersonate attack.

Definition 6 (IND-CPA)**.** *Let $\mathcal{O}$ denote a set of oracles: a hash oracle $\mathcal{O}_{\mathcal{R}}(\cdot)$, a encryption key generation oracle $\mathcal{O}_{\mathsf{EKGen}}(\cdot)$, and a decryption key generation oracle $\mathcal{O}_{\mathsf{DKGen}}(\cdot)$. The* IND-CPA *security definition of a matchmaking attribute-based encryption scheme $\mathcal{MABE}$ is based on the following experiment:*

```
$\mathbf{Exp}^{\mathsf{IND\text{-}CPA}}_{\mathcal{MABE},\mathcal{A}}(1^\lambda)$
  $\mathcal{R}^* \leftarrow \mathcal{A}(1^\lambda)$;
  $\mathcal{D}_{\mathbb{R}} = \emptyset$;
  $(mpk, msk) \leftarrow \mathsf{Setup}(1^\lambda)$;
  $(m_0, m_1, \mathcal{S}_0, \mathcal{S}_1) \leftarrow \mathcal{A}^{\mathcal{O}}(mpk)$;
  $b \in \{0,1\}$;
  $ek \leftarrow \mathsf{EKGen}(msk, \mathcal{S}_b)$;
  $\mathcal{S}' \in \mathcal{S}_b \cap \mathcal{S}_{1-b}$;
  $c \leftarrow \mathsf{Enc}(ek, \mathcal{R}^*, \mathcal{S}', m_b)$;
  $b' \leftarrow \mathcal{A}^{\mathcal{O}}(c)$;
  return 1 iff $b = b'$ and $\mathcal{R} \not\models \mathcal{D}_{\mathbb{R}}$.

$\mathbf{Oracle}\ \mathcal{O}_{\mathcal{R}}(att)$
  return $\mathcal{H}_2(att)$.
$\mathbf{Oracle}\ \mathcal{O}_{\mathsf{EKGen}}(\mathcal{S})$
  $ek \leftarrow \mathsf{EKGen}(msk, \mathcal{S})$;
  return $ek$.
$\mathbf{Oracle}\ \mathcal{O}_{\mathsf{DKGen}}(\mathbb{R})$
  $\mathcal{D}_{\mathbb{R}} \leftarrow \mathcal{D}_{\mathbb{R}} \cup \{\mathbb{R}\}$;
  $dk \leftarrow \mathsf{DKGen}(msk, \mathbb{R})$;
  return $dk$.
```

A matchmaking attribute-based encryption scheme $\mathcal{MABE}$ is said to be IND-CPA *secure if for any probabilistic polynomial-time adversary $\mathcal{A}$, the following advantage is negligible:*

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

Our IND-CPA security model covers sender guessing attack, collusion attack and eavesdropping attack. The detailed analysis is described as follows:

Sender guessing attack: $\mathcal{A}$ is allowed to output two attribute sets of senders $\mathcal{S}_0$ and $\mathcal{S}_1$ $s.t.$ $(m_0, m_1, \mathcal{S}_0, \mathcal{S}_1) \leftarrow \mathcal{A}^{O}(mpk)$. In the challenge phase, a random bit b is chosen to pick a sender associated $\mathcal{S}_b$. In the guessing phase, $\mathcal{A}$ outputs a bit b' to guess the bit b and wins the game if $b = b'$. If $\mathcal{A}$ can launch the sender guessing attack, the probability of $\mathcal{A}$ wins the game is more than random guess.

Collusion attack: $\mathcal{A}$ is allowed to query any encryption key via the encryption key generation oracle $O_{\mathsf{EKGen}}(\cdot)$ and any decryption key except the one satisfying the ciphertext c via the the decryption key generation oracle $O_{\mathsf{DKGen}}(\cdot)$. Therefore, $\mathcal{A}$ can launch collusion attack as the group of any senders and authorised receivers without associating $\mathbb{R}$ $s.t. \mathcal{R}^* \models \mathbb{R}$ and $\mathcal{R}^*$ is used to derive the challenge ciphertext c.

Eavesdropping attack: $\mathcal{A}$ is allowed to output two messages m_0 and m_1. In the challenge phase, a random bit b is chosen to generate a challenge ciphertext depending on m_b. In the guessing phase, $\mathcal{A}$ outputs a bit b' to guess the bit b and wins the game if $b = b'$. If $\mathcal{A}$ can launch the eavesdropping attack, the probability of $\mathcal{A}$ wins the game is more than random guess.

Definition 7 (EU-CMA)**.** *Let O denote a set of oracles: a hash oracle $O_{\mathcal{S}}(\cdot)$, a encryption key generation oracle $O_{\mathsf{EKGen}}(\cdot)$, a decryption key generation oracle $O_{\mathsf{DKGen}}(\cdot)$, and an encryption oracle $O_{\mathsf{Enc}}(\cdot,\cdot,\cdot,\cdot)$. The* EU-CMA *security definition of a matchmaking attribute-based encryption scheme $\mathcal{MABE}$ is based on the following experiment:*

```
$\mathbf{Exp}^{\mathsf{EU\text{-}CMA}}_{\mathcal{MABE},\mathcal{A}}(1^{\lambda})$
    $\mathbb{S}^* \leftarrow \mathcal{A}(1^{\lambda})$
    $\mathcal{D}_{\mathcal{S}} = \emptyset;$
    $\mathcal{D}_c = \emptyset;$
    $(mpk, msk) \leftarrow \mathsf{Setup}(1^{\lambda});$
    $c^* \leftarrow \mathcal{A}^{O}(mpk);$
    $return\ 1\ iff\ \mathsf{Verify}(\mathbb{S}^*, c) = 1$, 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 $O_{\mathcal{S}}(att)$
    $return\ \mathcal{H}_1(att).$
Oracle $O_{\mathsf{EKGen}}(\mathcal{S})$
    $\mathcal{D}_{\mathcal{S}} \leftarrow \mathcal{D}_{\mathcal{S}} \cup \{\mathcal{S}\};$
    $ek \leftarrow \mathsf{EKGen}(msk, \mathcal{S});$
    $return\ ek.$
Oracle $O_{\mathsf{DKGen}}(\mathbb{R})$
    $dk \leftarrow \mathsf{DKGen}(msk, \mathbb{R});$
    $return\ dk.$
Oracle $O_{\mathsf{Enc}}(ek, \mathcal{R}, \mathcal{S}, m)$
    $c \leftarrow \mathsf{Enc}(ek, \mathcal{R}, \mathcal{S}, m)$
    $\mathcal{D}_c \leftarrow \mathcal{D}_c \cup \{c\}$
    $return\ c.$
```

A matchmaking attribute-based encryption scheme $\mathcal{MABE}$ is said to be EU-CMA *secure if for any probabilistic polynomial-time adversary $\mathcal{A}$, the following advantage is negligible:*

$$\mathbf{Adv}^{\mathsf{EU\text{-}CMA}}_{\mathcal{MABE},\mathcal{A}}(1^{\lambda}) = \Pr[\mathbf{Exp}^{\mathsf{EU\text{-}CMA}}_{\mathcal{MABE},\mathcal{A}}(1^{\lambda}) = 1].$$

Our EU-CMA security model covers impersonate attack and collusion attack. The detailed analysis is described as follows:

Impersonate attack: $\mathcal{A}$ is allowed to query information of senders by querying the encryption key generation oracle $O_{\mathsf{EKGen}}(\cdot)$ to gain the encryption key and querying encryption oracle $O_{\mathsf{Enc}}(\cdot,\cdot,\cdot,\cdot)$ to gain the ciphertext. If $\mathcal{A}$ can then modify the decryption keys and the ciphertexts to output the ciphertext c under the policy $\mathbb{S}$. $\mathcal{A}$ wins the game if c is a message has not been queried and no encryption key associated $\mathcal{S}$ $s.t.$ $\mathcal{S} \models \mathbb{S}$ has been queried. If $\mathcal{A}$ can launch the impersonate attack, the probability of $\mathcal{A}$ wins the game is more than negligible.

Collusion attack: $\mathcal{A}$ is allowed to query any decryption key via the the decryption key generation oracle $O_{\mathsf{DKGen}}(\cdot)$ and any encryption key except the one satisfying the ciphertext c via the encryption key generation oracle $O_{\mathsf{EKGen}}(\cdot)$. Therefore, $\mathcal{A}$ can launch collusion attack as the group of any receivers and authorized senders without associating $\mathcal{S}$ $s.t.$ $\mathcal{S} \models \mathbb{S}$ and $\mathbb{S}$ is used to derive the ciphertext c.

Therefore, our IND-CPA and EU-CMA security models cover all possible attacks in the threat model.

V. Proposed Scheme

A. Workflow of CFDS

With reference to Fig. 3, we now describe the workflow of our CFDS which employs $\mathcal{MABE} = \{\mathsf{Setup}, \mathsf{EKGen}, \mathsf{DKGen}, \mathsf{Enc}, \mathsf{Verify}, \mathsf{Dec}\}$. Our system has three phases: system initialization, data uploading, and data downloading.

System Initialization: Fig. 4 shows the initialization phase of CFDS. The KGC runs the Setup algorithm to generate the system parameters and distributes the public parameters to each entity. The KGC runs EKGen and DKGen to generate an encryption key and a decryption key, then distribute them to the ED which is usually controlled by a device owner (See ①).

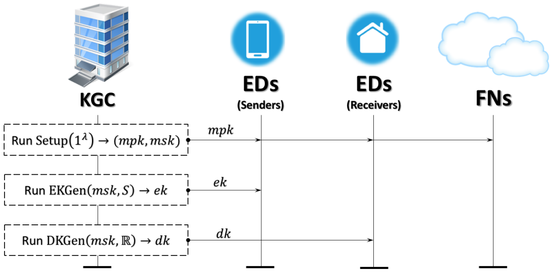

KGC
EDs
(Senders)
EDs
(Receivers)
FNs
Run Setup$(1^{\lambda}) \rightarrow (mpk, msk)$
mpk
Run EKGen$(msk, S) \rightarrow ek$
ek
Run DKGen$(msk, \mathbb{R}) \rightarrow dk$
dk

Fig. 4: System Initialization in CFDS

Data Uploading: Fig. 5 shows the phase of message uploading in CFDS, where FN_i and FN_j denote two different fog nodes. An ED runs the Enc algorithm to generate ciphertexts and outsource them to FNs (See ③). FNs analyze the propose of ciphertexts based on the sender's attributes. If the ciphertexts have short-term purposes (e.g., comment to machine-to-machine communications), FNs store the ciphertexts locally or forward to another FNs. If the ciphertexts have long-term purposes (e.g., historical records and log documents), FNs forward the ciphertexts to the CSP (See ④).

Data Downloading: Fig. 5 depicts the phase of message downloading in CFDS. When an ED requests information from the desirable senders (See ②), FNs returns the ciphertexts from desirable senders if the ciphertexts are stored locally (See ③); otherwise, FNs request ciphertexts from the CSP (See ④) and other FNs to extract the desirable ciphertexts to EDs (See ③). Note that the interaction between FNs is for machine-to-machine communication [12], FNs collaborate with each other for data intelligence and sharing resources like processing power and memory.

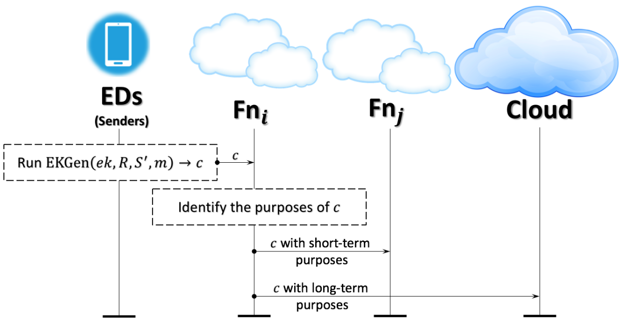

EDs
(Senders)
Fn$_i$
Fn$_j$
Cloud
Run EKGen$(ek, R, S', m) \to c$
c
Identify the purposes of c
c with short-term purposes
c with long-term purposes

Fig. 5: System Uploading in CFDS

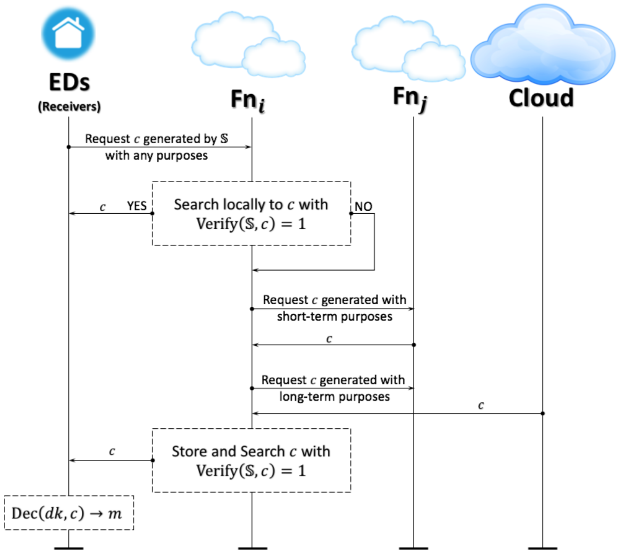

EDs
(Receivers)
Fn$_i$
Fn$_j$
Cloud
Request c generated by $\mathbb{S}$ with any purposes
c
YES
Search locally to c with Verify$(\mathbb{S}, c) = 1$
NO
Request c generated with short-term purposes
c
Request c generated with long-term purposes
c
c
Store and Search c with Verify$(\mathbb{S}, c) = 1$
Dec$(dk, c) \to m$

Fig. 6: System Downloading in CFDS

B. Concrete Construction of MABE

Our concrete construction is based on the idea of matchmaking encryption [10] for bilateral access control. Specifically, we apply the ABE scheme proposed by Rouselakis and Waters [20] enabling the sender to specify the access structure of receiver and design a variety of ABE derived from [20] ensuring the receiver to verify the sender. The details of each algorithm are given below.

Setup(1^λ): The setup algorithm runs the pairing group generator $\mathcal{G}(1^\lambda)$ to generate the description of bilinear group $(p, \mathbb{G}, \mathbb{G}_T, e, g)$, and randomly picks $\alpha, \beta \in \mathbb{Z}_p$ and three hash functions $\mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3$:

$$\mathcal{H}_1 : \Omega_{\mathsf{snd}} \to \mathbb{G}, \quad \mathcal{H}_2 : \Omega_{\mathsf{rcv}} \to \mathbb{G}, \quad \mathcal{H}_3 : \{0,1\}^* \to \mathbb{G},$$

where $\mathcal{H}_3$ is a collision-resistant hash function, and $\mathcal{H}_1$ and $\mathcal{H}_2$ are modelled as random oracles. The algorithm returns the master public key mpk and the master private key mpk.

$$mpk = (p, \mathbb{G}, \mathbb{G}_T, e, g, \mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3, e(g,g)^\alpha, e(g,g)^\beta),$$
$$msk = (g^\alpha, g^\beta).$$

EKGen$(msk, \mathcal{S})$: Parse the set of sender's attributes $\mathcal{S} = (att_{\mathsf{snd},1}, att_{\mathsf{snd},2}, \dots, att_{\mathsf{snd},k})$. The encryption key generation algorithm randomly picks $r \in \mathbb{Z}_p$ and for $i \in [k]$, it computes:

$$ek_{1,i} = g^\alpha \mathcal{H}_1(att_{\mathsf{snd},i})^r, \quad ek_2 = g^r.$$

The algorithm returns the encryption key $ek = (\mathcal{S}, \{ek_{1,i}\}_{i \in [k]}, ek_2)$.

DKGen$(msk, \mathbb{R})$: Parse the access structure of the receiver $\mathbb{R} = (\mathbb{N}, \pi)$, where $\mathbb{N} \in \mathbb{Z}_p^{\ell_{\mathbb{N}} \times n_{\mathbb{N}}}$ is a matrix and $\pi : [\ell_{\mathbb{N}}] \to \Omega_{\mathsf{rcv}}$ is a mapping function. The decryption key generation algorithm randomly picks $\vec{y} = (\beta, y_2, \dots, y_{n_{\mathbb{N}}})^\perp \in \mathbb{Z}_p^{n_{\mathbb{N}} \times 1}$ and computes $\vec{\lambda} = (\lambda_1, \lambda_2, \dots, \lambda_{\ell_{\mathbb{N}}}) = \mathbb{N}\vec{y}$. For $i \in [\ell_{\mathbb{N}}]$, it randomly chooses $r_i \in \mathbb{Z}_p$ and computes:

$$dk_{1,i} = g^{\lambda_i} \mathcal{H}_2(\pi(i))^{r_i}, \quad dk_{2,i} = g^{r_i}$$

The algorithm returns the decryption key $dk = ((\mathbb{N}, \pi), \{dk_{1,i}, dk_{2,i}\}_{i \in [\ell_{\mathbb{N}}]})$.

Enc$(ek, \mathcal{R}, \mathcal{S}', m)$: Parse the set of receiver's attributes $\mathcal{R} = (att_{\mathsf{rcv},1}, att_{\mathsf{rcv},2}, \dots, att_{\mathsf{rcv},l})$ and the set of sender's attributes $\mathcal{S}' = (att_{\mathsf{snd},1}, att_{\mathsf{snd},2}, \dots, att_{\mathsf{snd},k'})$, where $\mathcal{S}'$ is the subset of sender's attributes $\mathcal{S}$ defined in the encryption key ek, *s.t.* $\mathcal{S}' \subseteq \mathcal{S}$. The encryption algorithm randomly picks $s, r', t \in \mathbb{Z}_p$, and for $i \in [l]$, it computes:

$$c_0 = m \cdot e(g,g)^{\beta s}, \quad c_1 = g^s, \quad c_{2,i} = \mathcal{H}(att_{\mathsf{rcv},i})^s,$$
$$c_3 = ek_2 \cdot g^{r'} = g^{r+r'}, \quad c_4 = g^t.$$

Let $c_{1\text{-}4}$ denote a binary string as $c_{1\text{-}4} = c_0 \| c_1 \| c_{2,i} \| \dots \| c_{2,l} \| c_3 \| c_4$. For $i' \in [k']$, it finds j such that $att_{\mathsf{snd},i'} = att_{\mathsf{snd},j}$ (j exists due to $\mathcal{S}' \subseteq \mathcal{S}$) and computes:

$$ek_{1,i'} = ek_{1,j} \cdot \mathcal{H}_1(att_{\mathsf{snd},i'})^{r'} = g^\alpha \mathcal{H}_1(att_{\mathsf{snd},i'})^{r+r'},$$
$$c_{5,i'} = ek_{1,i'} \cdot \mathcal{H}_3(c_{1\text{-}4})^t = g^\alpha \mathcal{H}_1(att_{\mathsf{snd},i'})^{r+r'} \mathcal{H}_3(c_{1\text{-}4})^t.$$

The algorithm returns the ciphertext $c = ((\mathcal{S}, \mathcal{R}'), c_0, c_1, \{c_{2,i}\}_{i \in [l]}, c_3, c_4, \{c_{5,i'}\}_{i' \in [k']})$.

Verify$(\mathbb{S}, c)$: Parse the access structure of the sender $\mathbb{S} = (\mathbb{M}, \rho)$, where $\mathbb{M} \in \mathbb{Z}_p^{\ell_{\mathbb{M}} \times n_{\mathbb{M}}}$ is a matrix and $\rho : [\ell_{\mathbb{M}}] \to \Omega_{\mathsf{snd}}$ is a mapping function. The verification algorithm randomly picks $\vec{x} = (1, x_2, \dots, x_{n_{\mathbb{M}}})^\perp \in \mathbb{Z}_p^{n_{\mathbb{M}} \times 1}$ and computes $\vec{\kappa} = (\kappa_1, \kappa_2, \dots, \kappa_{\ell_{\mathbb{M}}}) = \mathbb{M}\vec{x}$. Let I be the set *s.t.* $I = \{i | i \in [\ell_{\mathbb{M}}], \rho(i) = \mathcal{S}\}$. It finds $\{\omega_i\}_{i \in I}$ such that $\sum_{i \in I} \omega \mathbb{M}_i = (1, 0, \dots, 0)$ and returns 1 if the following equality holds:

$$\prod_{i \in I} \left(\frac{e(c_{5,i}, g)}{e(\mathcal{H}_1(att_{\mathsf{snd},i}), c_3) \cdot e(\mathcal{H}_3(c_{1\text{-}4}), c_4)} \right)^{\kappa_i \omega_i} \overset{?}{=} e(g,g)^\alpha;$$

otherwise, the algorithm returns 0.

Dec(dk, c): Let J be the set such that $J = \{j | j \in [\ell_{\mathbb{N}}], \pi(j) = \mathcal{R}\}$. The decryption algorithm takes terms $\{\eta\}_{j \in J}$ such that $\sum_{j \in J} \eta \mathbb{N}_j = (1, 0, \dots, 0)$ and computes:

$$c_0 \cdot \prod_{j \in J} \left(\frac{e(dk_{2,i}, c_{2,j})}{e(dk_{1,i}, c_1)} \right)^{\eta_j} = m,$$

where i is the index of the attribute $\pi(i)$ in $\mathcal{R}$ *s.t.* $\pi(i) = att_{\mathsf{rcv},j}$. The algorithm returns the message m.

C. Correctness

If $\mathcal{S} \models \mathbb{S}$, the ciphertext can pass the verification according to the verification algorithm:

$$\begin{aligned} & \prod_{i \in I} \left(\frac{e(c_{5,i}, g)}{e(\mathcal{H}_1(att_{\mathsf{snd},i}), c_3) \cdot e(\mathcal{H}_3(c_{1\text{-}4}), c_4)} \right)^{\kappa_i \omega_i} \\ = & \prod_{i \in I} \left(\frac{e(g^{\alpha} \mathcal{H}_1(att_{\mathsf{snd},i})^{r+r'} \mathcal{H}_3(c_{1\text{-}4})^t, g)}{e(\mathcal{H}_1(att_{\mathsf{snd},i}), g^{r+r'}) \cdot e(\mathcal{H}_3(c_{1\text{-}4}), g^t)} \right)^{\kappa_i \omega_i} \\ = & \prod_{i \in I} e(g^{\alpha}, g)^{\kappa_i \omega_i} = e(g,g)^{\alpha \sum_{i \in I} \kappa_i \omega_i} = e(g,g)^{\alpha}. \end{aligned}$$

If $\mathcal{R} \models \mathbb{R}$, the message can be recovered according to the decryption algorithm:

$$\begin{aligned} & c_0 \cdot \prod_{j \in J} \left(\frac{e(dk_{2,i}, c_{2,j})}{e(dk_{1,i}, c_1)} \right)^{\eta_j} \\ = \ & m \cdot e(g,g)^{\beta s} \cdot \prod_{j \in J} \left(\frac{e(g^{r_j}, \mathcal{H}(att_{\mathsf{rcv},j})^s)}{e(g^{\lambda_j} \mathcal{H}_2(\pi(i))^{r_j}, g^s)} \right)^{\eta_j} \\ = \ & m \cdot e(g,g)^{\beta s} \cdot \prod_{j \in J} e(g^{\lambda_j}, g^s)^{-\eta_j} \\ = \ & m \cdot e(g,g)^{\beta s} \cdot e(g,g)^{-s \sum_{j \in J} \lambda_j \eta_j} \\ = \ & m \cdot e(g,g)^{\beta s} \cdot e(g,g)^{-\beta s} = m. \end{aligned}$$

Therefore, our verification algorithm and decryption algorithm are correct.

D. Security Analysis of MABE

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

The sketch of security proof for theorem 1: we can build an algorithm $\mathcal{B}$ interacting with $\mathcal{A}$ who can break our proposed scheme with non-negligible advantage to break the decisional BDH problem for a tuple $(p, \mathbb{G}, \mathbb{G}_T, e, g, A = g^a, B = g^b, C = g^c, T)$ to guess $T = e(g,g)^{abc}$ or a random term. To simulate our proposed scheme, $\mathcal{B}$ sets the master public key mpk to $(p, \mathbb{G}, \mathbb{G}_T, e, B = g^b, e(B,B)^{\alpha}, e(A,B))$. Hence, the group generator is B and the master secret key is g^{α} with $\alpha \in \mathbb{Z}_p$ and g^{α} with $\beta = a/b$, where a and b from $A = g^a$ and $B = g^b$. To make challenge ciphertext for m_0, m_1 from $\mathcal{A}$, $\mathcal{B}$ uses $B = g^c$ as the random term c_1 and chooses a random bit $b \in \{0,1\}$. Hence, the message hiding component is $m_b \cdot e(g,g)^{abc}$ if the giving challenge information is a valid BDH tuple, such as $T = e(g,g)^{abc}$; otherwise, the probability of $\mathcal{A}$ winning the game is no more than random guessing the bit b. The details of the proof for Theorem 1 is in Appendix A.

Theorem 2. *If the computational BDH assumption holds, then all probabilistic polynomial-time adversaries have a negligible advantage in breaking* EU-CMA *security of our scheme.*

The sketch of security proof for theorem 2: we can build an algorithm $\mathcal{B}$ interacting with $\mathcal{A}$ who can break our proposed scheme with non-negligible advantage to break the computational BDH problem for a tuple $(p, \mathbb{G}, \mathbb{G}_T, e, g, A = g^a, B = g^b, C = g^c)$ to output $e(g,g)^{abc}$. To simulate our proposed scheme, $\mathcal{B}$ set the master public key mpk to $(p, \mathbb{G}, \mathbb{G}_T, e, B = g^b, e(A,B), e(A,B)^{\beta})$. Hence, the group generator is B and the master secret key is g^{α} with $\alpha = a/b$ and g^{β} with $\beta \in \mathbb{Z}$, where a and b from $A = g^a$ and $B = g^b$. To ensure $\mathcal{A}$ can return the useful ciphertext, $\mathcal{B}$ embeds C for all the queries on the challenge sender's policy on the random oracle. Hence, $\mathcal{B}$ can extract the message $e(g,g)^{abc}$ at the end. The details of the proof for Theorem 2 is in Appendix B.

E. Security Analysis of CFDS

Theorem 3. *If the underlying MABE is* IND-CPA *and* EU-CMA *secure, then our proposed CFDS is secure against all possible attacks defined in the threat model.*

According to Theorem 1 and Theorem 2, the security of CFDS can be inferred by the MABE IND-CPA and EU-CMA secure. In the following, we analyze the possible attacks as defined in the threat model.

Sender guessing attack: this attack can be reduced to the attacks in the IND-CPA model and the adversary is any party except the message owner. Our scheme allows the senders to pick a random number in the domain $\mathbb{Z}_p$ to re-random the encryption key. Thus, this random number is unknown to any party except the sender itself. Therefore, the adversary cannot identify the sender or link two ciphertexts to one sender when the sender's attributes are not unique.

Eavesdropping attack: this attack can be reduced to the attacks in the IND-CPA model and the adversary is any party except the message owner and the valid receivers. Hence, we consider following two types of eavesdropping attack:

- Outsider attack: the outsiders do not have the valid decryption key, they can obtain the ciphertext and cannot learn anything beyond the ciphertext. Note that the outsiders have less capability than the others, we only consider the powerful adversaries in our security proof since they cover the attack capability of the other entities.
- Insider attack: we consider two types of insider attack: one is the malicious CSP and FNs without any decryption key, and the other one is the EDs with the invalid decryption key.
  - Malicious CSP and FNs: they cannot learn anything beyond the ciphertext as the reasons explained in the case of outsider attack.
  - EDs with the invalid decryption key: although these EDs does not have the valid decryption key, they can try combining the decryption keys to make a new decryption key satisfying the policy to decrypt the ciphertext. However, this decryption key is invalid since every decryption key has a unique random seed. Therefore, this combined decryption key has at least two random seeds leading to decryption fail. Therefore, the ciphertext is still secure.

Impersonate attack: this attack can be reduced to the attacks in the EU-CMA model. The adversary may launch impersonate attack via following two strategies:

1) Forge an encryption key: Similar to the reason in eavesdropping attack. The outsiders, malicious CSP, and FNs do not have any encryption key. Hence, they cannot forge

any encryption key. The EDs with the invalid encryption key also cannot forge a valid encryption key since every decryption key is derived from a unique random seed, encryption key with at least two random seeds leads to verification process fails.

2) Derive a valid ciphertext from existing ciphertexts. The collision-resistant hash function prevents any manipulation to the ciphertexts. The attackers cannot find two messages with different information but the same hash value because of the collision-resistant property of the hash function. Therefore, the changed ciphertext is an invalid ciphertext which cannot pass the verification process.

Collusion attack: this attack can be reduced to the attacks in the IND-CPA and EU-CMA model. The purpose of collision attack is to derive the valid ciphertext without the valid encryption key and decrypt the ciphertext without a valid decryption key.

- Derive the valid ciphertext without the valid encryption key: similar as the reasoning in eavesdropping attack, every encryption key is derived from unique randomness. The combination of multiple encryption keys cannot help the attacker to derive a valid encryption key. Hence, the collusion attack cannot help to obtain valid ciphertexts.
- Decrypt the ciphertext without valid decryption key: as the reasons in impersonate attack, every decryption key is derived from unique randomness. The combination of multiple decryption keys cannot help attack to derive valid decryption key. Hence, the collusion attack cannot help to decrypt unauthorized ciphertexts.

VI. Efficiency Analysis

In this section, we give the efficiency analysis from two aspects: theoretical complexity and experimental performance. For theoretical complexity, we analyze computational complexity and space complexity. For experimental performance, we compare running time and data storage among DP-ABE [13], MIBE [10], and ours since DP-ABE and modified MIBE are two comparable solutions in terms of functionality. We also modify the MIBE (modified MIBE, short for mMIBE) to achieve comparable performance to MABE, where our modified MIBE with access structure using AND gate only by dividing message into various pieces and encrypted each piece under different identities to simulate the set of attributes in MABE.

A. Theoretical Complexity

Table II presents the comparison of computational complexity and space complexity among some most relevant solutions [10], [13]–[16] and ours.

For computational complexity, our scheme is comparable to DP-ABE and ACE, and much better than mMIBE. Our scheme takes a constant time to setup the system. The computational complexity of the encryption key generation bases on the set of sender's attributes, which is comparable to ACE depending on the number of senders/receivers specified by the policy. The decryption key generation algorithm only relates to the policy of receivers rather than attributes of senders and policies of receivers simultaneously in DP-ABE. The encryption algorithm is about the number and senders' attributes and receivers' attribute rather than the multiplicity of the number and senders' attributes and receivers' attribute. The sender verification algorithm only checks the validate of senders, which takes the computational complexity based on the specified policy. The decryption algorithm only relates to the policy of receivers since the sender verification can be outsourced to the fog nodes.

For space complexity, our scheme is much better than other schemes since it has a constant-size system parameter by applying collision-resistant hash functions. The encryption key generation and decryption key generation relate to the number of sender's attributes and the number of receiver's policy, respectively. The space complexity of ciphertext is comparable to others since the ciphertext depends on the number of sender's attributes and the receiver's policy.

B. Experimental Performance

For experimental analysis, we evaluate the two most related works [10], [13] and ours. Specifically, XLD+19 [13] is DP-ABE with complex access control in ciphertexts (e.g., ciphertext-policy access control and key-policy access control) instead of entities (e.g., sender access control and receiver access control) as in our MABE and AFNV19 [10] represents mMIBE derived from MIBE for simulating the access control in our MABE. Although ACE provides access control on data flow from a sender to a receiver, the rules "*no read up* and *no write down*" in ACE is quite different in our proposed system. Besides, ACE requires a fully trusted sanitizer always online to ensure the rule runs in each ciphertext and prevent attacks from the malicious senders and receivers.

Our experimental simulation was performed on a PC running 64-bit Windows 10 with 3.60GHz Intel(R) Core(TM) i7-4790 CPU and 24GB memory. The implementation is based on JPBE 2.0.0 with Type A elliptic curve based on the standard parameters from "a.properties" from JPBC library. The experimental performance is demonstrated in Fig. 7 and Fig. 8.

Fig. 7 gives the experimental performances about the algorithm running time.

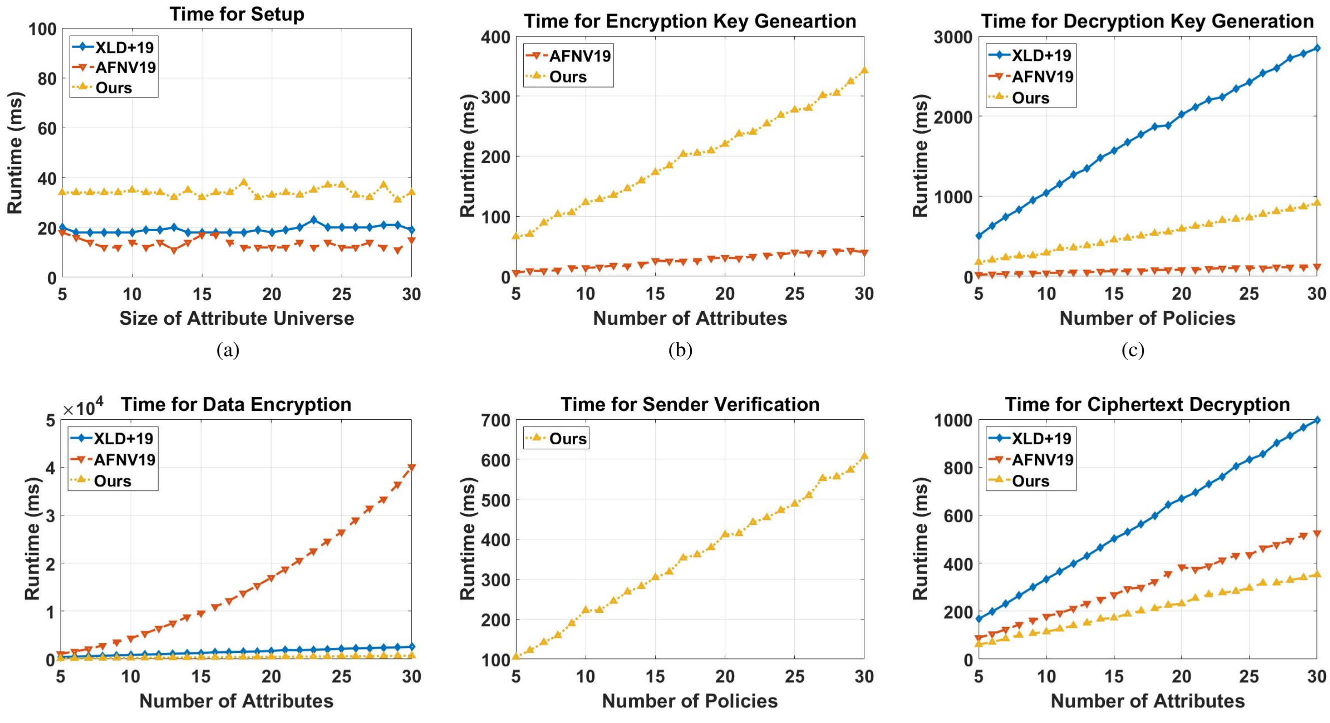
Fig. 7a presents the running time for system setup as a function of the size of the attribute universe. The time consumption is irrelevant to the size of the attribute universe. Our scheme takes much more time than other schemes since our scheme requires two pairing computation to build the bilateral fine-grained access control. XLD+19 only offers one-way access control (e.g., the sender specifies the receivers), and AFNV19 only provides the bilateral access control in a coarse-grained level (or AND gate only in a fine-grained level).

Fig. 7b illustrates the time for encryption key generation as a function of the number of attributes. Our scheme takes more time than mMIBE since our scheme can achieve the fine-grained access control with LSSS, while mMIBE only has access structure using AND gate, which cannot support OR gate. DP-ABE only has one-way access control to limit

TABLE II: Theoretical Comparison among Existing Solutions Related to Bilateral Access Control in the Prime-Order Group

	Type of	*Computational Complexity*						*Space Complexity*			
	Scheme	***Setup***	***EKGen***	***DKGen***	***Enc***	***SV***	***Dec***	***SP***	***EK***	***DK***	***C***
***AI09** [14]*	*DP-ABE*	$O(\Omega_{\mathsf{snd}}+\Omega_{\mathsf{rec}})$	*N/A*	$O(\mathcal{S}+\mathbb{R})$	$O(\mathbb{S}+\mathcal{R})$	*N/A*	$O(\mathcal{S}+\mathbb{R})$	$O(\Omega_{\mathsf{snd}}+\Omega_{\mathsf{rec}})$	*N/A*	$O(\mathcal{S}+\mathbb{R})$	$O(\mathbb{S}+\mathcal{R})$
XLD+19 [13]	*DP-ABE*	$O(1)$	*N/A*	$O(\mathcal{S}+\mathbb{R})$	$O(\mathbb{S}+\mathcal{R})$	*N/A*	$O(\mathcal{S}+\mathbb{R})$	$O(1)$	*N/A*	$O(\mathcal{S}+\mathbb{R})$	$O(\mathbb{S}+\mathcal{R})$
DHO16 [15]	*ACE*	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(1)$	$O(n)$	$O(n)$	$O(n)$	$O(2^n)$
KW17 [16]	*ACE*	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$\mathsf{poly}(2^n)$
AFNV19 [10]	*MIBE*	$O(1)$	$O(1)$	$O(1)$	$O(1)$	*N/A*	$O(1)$	$O(1)$	$O(1)$	$O(1)$	$O(1)$
AFNV19 [10]	*mMIBE*	$O(1)$	$O(\mathcal{S})$	$O(\mathbb{R})$	$O(\mathcal{S}\cdot\mathcal{R})$	*N/A*	$O(\mathbb{S}+\mathbb{R})$	$O(1)$	$O(\mathcal{S})$	$O(\mathbb{R})$	$O(\mathcal{S}+\mathbb{R})$
Ours	*MABE*	$O(1)$	$O(\mathcal{S})$	$O(\mathbb{R})$	$O(\mathcal{S}+\mathcal{R})$	$O(\mathbb{S})$	$O(\mathbb{R})$	$O(1)$	$O(\mathcal{S})$	$O(\mathbb{R})$	$O(\mathcal{S}+\mathbb{R})$

EKGen: the encryption key generation algorithm; ***DKGen***: the decryption key generation algorithm; ***Enc***: the encryption algorithm;
SV: the sender verification algorithm, where this algorithm is run by untrustworthy fog nodes in ours and by fully trusted sanitizer in ACE;
Dec: the decryption algorithm; ***SP***: the system parameter; ***EK***: the encryption key;
DK: the decryption key; ***C***: the ciphertext; $\mathsf{poly}(2^n)$: poly-logarithmic in n;
n: the number of senders/receivers specified by the policy, where the policy is defined by the rule "*no read up, no write down*" in ACE;
mMIBE: modified MIBE to achieve comparable functionality (access structure with AND gate only) to MABE.



(a) (b) (c)

(d) (e) (f)

Fig. 7: Experimental Performances about the Algorithm Running Time

the receiver to reveal the ciphertext. There is no encryption key generation algorithm for DP-ABE.

Fig. 7c displays the time for decryption key generation vs the size of policies. Our scheme has better performance than DP-ABE since the decryption key in our scheme focuses on the simple fine-grained access control for the receivers rather than complex access control (combing the access control of KP-ABE and CP-ABE simultaneously). mMIBE has better performance in the decryption key generation due to its simple access control policy.

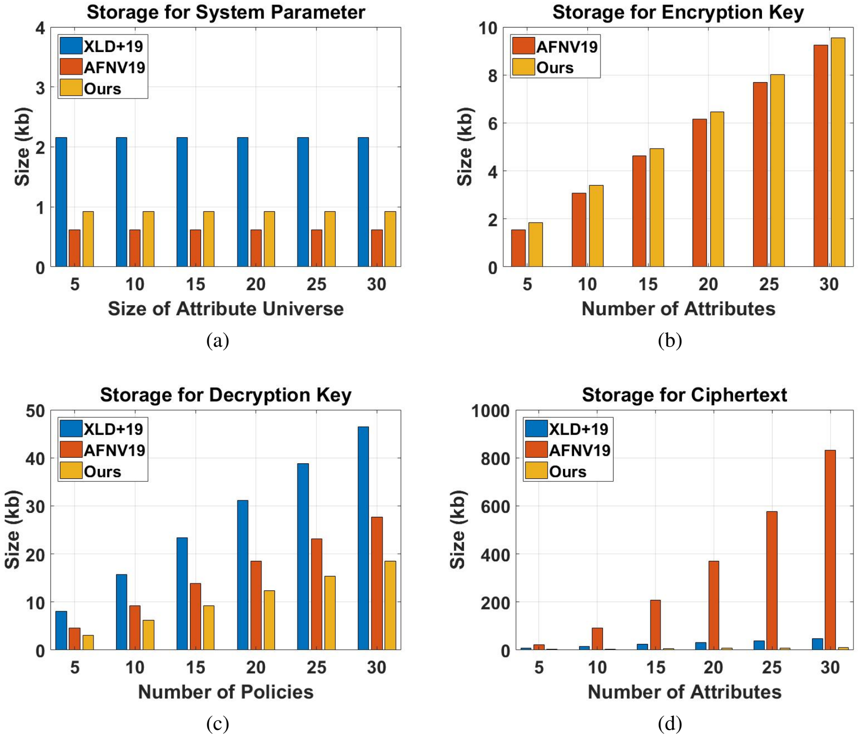
Fig. 7d presents the time for data encryption vs the number of attributes. The mMIBD takes much more time than others since the direct extension from an identity-based setting to an attribute-based setting incurs large overheads. Our scheme has better performance than others due to collision-resistant hash functions are used to generate the ciphertexts.

Fig. 7e illustrates time for verifying senders in our scheme. Note that other schemes do not support outsourcing sender verification by a third party. The time of sender verification increases linearly in the size of policies in our scheme.

Fig. 7f displays the encryption running time vs the number of attributes. Our scheme has better performance than others since the workload of sender verification can be outsourced.

Fig. 8 gives the experimental performances on the storage for the system parameter, the encryption key, the decryption key, and the ciphertext.

Fig. 8a presents the size of the system parameter as a function of the attribute universe. The storage of the system parameter is irrelevant to the size of the attribute universe. Our scheme has better performance than DP-ABE since the latter requires additional 6 elements in $\mathbb{G}$. mMIBE has better performance since modifying MIBE to mMIBE does not



(a)

(b)

(c)

(d)

Fig. 8: Experimental Results about Storage Overhead

extend the size of the system parameter.

Fig. 8b illustrates the size of the encryption key vs the size of the attribute universe. Our scheme and mMIBE have comparable performances. The size of the encryption key grows linear in the number of attributes.

Fig. 8c displays the size of decryption key vs the size of policies. The size of the decryption key is linear to the size of policies. Our scheme has smaller decryption key size than others since the simple fine-grained access control policy is used to control the receivers.

Fig. 8d gives the size of ciphertext vs the size of the attribute universe. The size of the ciphertext key is linear to the size of policies. Our scheme has smaller ciphertext size than others since the ciphertext in mMIBE is the multiply of sender's attributes and receiver's attributes, and DP-ABE has complex access control (combing the access control of KP-ABE and CP-ABE simultaneously).

Remark. Our experimental simulation follows the key encapsulation mechanism (KEM) as many cloud-based applications to improve the performance. A symmetric encryption algorithm (e.g., AES) with key domain $\mathcal{K}$ is chosen. To encrypt a message, we first pick a random key $k \in \mathcal{K}$, then encrypt the message under the key k and encrypt k under corresponding asymmetric encryption scheme (e.g., DP-ABE, mMIBE and ours). To reveal a message, we first run the asymmetric decryption scheme to reveal the symmetric encryption key k and reveal the message by using k to decrypt the symmetric ciphertext.

VII. Related Works

Table III give functionality comparison among existing solutions related to bilateral access control and ours. In the following, we give the detailed descriptions of them.

Sahai and Waters [4] first introduced fuzzy identity-based encryption (FIBE) as the rudiment of ABE, but it only supports threshold access control and is selectively secure. Following FIBE, there are many ABE schemes proposed. There are two primary flavors: key-policy ABE (KP-ABE) [11] and ciphertext-policy ABE (CP-ABE) [17]. In KP-ABE, the user's secret key is based on the access structure, and ciphertexts are encrypted over a set of attributes. In CP-ABE, the access structure specifics ciphertexts, and the receiver's secret key associates a set of attributes. To aggregate them, dual-policy ABE (DP-ABE) [13], [14] has been introduced to offer KP-ABE and CP-ABE simultaneously.

Goyal et al. [18] proposed the first KP-ABE scheme with a tree-based access structure. To enrich expressiveness, Ostrovsky et al. [11] introduced a KP-ABE scheme with non-monotonic access policies. Lewko and Waters [19] then proposed a KP-ABE with the large universe in the composite-order group without hash function and predefined maximum number of attributes to express a ciphertext. Rouselakis and Waters [20] proposed the first large universe KP-ABE with prime-order groups and constant-size parameters. After that, many KP-ABE schemes [21]–[25] have been proposed to improve performance in terms of the efficiency (e.g., constant-size ciphertexts) and security (e.g., from non-standard hard problems to standard hard problems).

TABLE III: Functionality Comparison among Existing Solutions Related to Bilateral Access Control

	Data Privacy	***Ciphertext Identification***	***Ciphertext Filter***	***Sender Anonymity***	***Outsourced Verification***	***Access Policy***
DP-ABE	✓	✗	✗	✓	✗	***one-way LSSS***
ABKS	✓	✗	✓	✓	✗	***one-way LSSS***
ACE	✓	✗	✓	✗	✓	***no read up, no write down***
MIBE	✓	✓	✗	✗	✗	***one-to-one data sharing***
Ours	✓	✓	✓	✓	✓	***bilateral LSSS***

Ciphertext Filter: the ciphertexts with specific information can be extracted without revealing the messages of ciphertexts;
Sender Anonymity: the ciphertexts cannot reveal the sensitive information (e.g., a unique ID) of the sender;
Outsourced Verification: the workload of ciphertext filter can be outsourced to a third party.
Access Policy: the approach controls the data flow in the system.
LSSS: linear secret sharing system (refers to *Definition 4* for details).

Bethencourt et al. [26] proposed the first CP-ABE scheme with a tree-based access policy. To support flexible expressiveness, Cheung and Newport [27] proposed a basic CP-ABE scheme which supports AND gate policies with positive and negative attributes simultaneously. To further improve expressiveness, Goyal et al. [28] proposed a tree-based CP-ABE scheme. Herranz et al. [29] presented a CP-ABE scheme in which the ciphertext size is constant. Waters [30] proposed a CP-ABE scheme supporting access policies expressed as LSSS. Rouselakis and Waters [20] proposed a large universe CP-ABE scheme with the LSSS access policy. After that, many CP-ABE schemes [17], [24], [31]–[34] have been proposed to improve performance in terms of the efficiency (e.g., fast decryption and constant computational cost) and security (e.g., from selectively secure to fully secure).

Attrapadung and Imai [14] introduced the first DP-ABE scheme with access controls in KP-ABE and CP-ABE simultaneously. Miyaji and Tran [35] proposed a DP-ABE with constant-size ciphertexts. Rao and Dutta [36] proposed a DP-ABE with efficient computation and constant-size ciphertexts simultaneously. Attrapadung and Yamada [37] proposed fully secure DP-ABE in the composite-order group. After that, many KP-ABE schemes [13], [36] have been proposed to improve performance in terms of efficiency and security.

ABKS was first introduced by Zheng et al. [6]. For search ciphertext without revealing messages, but it cannot identify the senders information for the ciphertext and request multiple rounds to search the messages. The senders first define keywords with a policy that specify the authorized keyword searchers and outsource the data to a cloud. The authorized receiver can search the ciphertexts based on different keywords. However, the keywords should be secrecy for any party and even the authorized receiver to keep the keyword secrecy, denoted keyword guessing attack [38]. To prevent this attack, Qiu et al. [39] introduced hidden policy ciphertext-policy attribute-based encryption with keyword search.

Damgard et al. [15] proposed the first ACE scheme. ACE provides access control over information flow. This information flow is from a sender to a sanitizer, and a sanitizer to a receiver, where the sanitizer acts as a fully trusted party to ensure secure data flow. The purpose of ACE is enforcing the *no read up* and *no write down* access rule. Specifically, each user has a different authorization to read and write a message. The receiver cannot read the message generated by a user with higher reading rights, and the senders cannot write a message as a user with lower writing rights. Hence, the policy in ACE is suitable to the system with a hierarchical access policy (e.g., army and enterprise) rather than the access control in the cloud-fog-device architecture. To improve the security, Kim and Wu [16] proposed a secure ACE with the standard assumptions.

Matchmaking encryption as a novel paradigm protects the data communications with bilateral access control user privacy proposed by Ateniese et al. [10]. The current solution in matchmaking encryption is only in identity-based setting, which cannot provide fine-grained access control.

VIII. Conclusion

In this work, we investigated problems of data sharing in cloud-fog computing and produced a secure cloud-fog-device data sharing system with fine-grained bilateral access control. The system model and the threat model were presented for our proposed system. To provide secure data sharing, we introduced a matchmaking attribute-based encryption with a formal definition, and security models, which supports the receiver identifying ciphertexts from undesirable senders without costly data decryption. Moreover, the workload of senders verification can be outsourced to the fog nodes. We believe that our proposed matchmaking attribute-based encryption finds many applications for providing data privacy and ciphertext identification simultaneously. The future work could be investigating a more efficiency MABE system with some practical properties (e.g., revocability, traceability, etc.).

Acknowledgement

This research is supported by Singapore National Research Foundation (Grant No. NRF2018NCR-NSOE004-0001) and AXA Research Fund, National Natural Science Foundation of China (Grant No. 61972094, 61822202 and 61872089), Key Research and Development Program of Shaanxi (Grant No. 2019KW-053), and the New Star Team of Xi'an University of Posts and Telecommunications (Grant No. 2016-02).

References

[1] S. Yi, Z. Qin, and Q. Li, "Security and privacy issues of fog computing: A survey," in *WASA*, 2015, pp. 685–695.

[2] Sstatista, "Forecast of fog computing market revenue worldwide from 2018 to 2022." [Online]. Available: https://www.statista.com/statistics/830485/world-fog-computing-revenue-by-vertical/

[3] I. Stojmenovic and S. Wen, "The fog computing paradigm: Scenarios and security issues," in *FedCSIS*, 2014, pp. 1–8.

[4] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *EUROCRYPT*, vol. 3494, 2005, pp. 457–473.

[5] 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. Information Forensics and Security*, vol. 7, no. 2, pp. 743–754, 2012.

[6] Q. Zheng, S. Xu, and G. Ateniese, "VABKS: verifiable attribute-based keyword search over outsourced encrypted data," in *INFOCOM*, 2014, pp. 522–530.

[7] D. Cash, P. Grubbs, J. Perry, and T. Ristenpart, "Leakage-abuse attacks against searchable encryption," *IACR Cryptology ePrint Archive*, vol. 2016, p. 718, 2016.

[8] J. Ning, J. Xu, K. Liang, F. Zhang, and E. Chang, "Passive attacks against searchable encryption," *IEEE Trans. Information Forensics and Security*, vol. 14, no. 3, pp. 789–802, 2019.

[9] Y. Zhang, J. Katz, and C. Papamanthou, "All your queries are belong to us: The power of file-injection attacks on searchable encryption," in *USENIX*, 2016, pp. 707–720.

[10] G. Ateniese, D. Francati, D. Nuñez, and D. Venturi, "Match me if you can: Matchmaking encryption and its applications," in *CRYPTO*, 2019, pp. 701–731.

[11] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based encryption with non-monotonic access structures," in *CCS*, 2007, pp. 195–203.

[12] P. K. D. Pramanik, S. Pal, A. Brahmachari, and P. Choudhury, "Processing iot data: From cloud to fog—it's time to be down to earth," in *SMAC*, 2018, pp. 124–148.

[13] S. Xu, Y. Li, R. H. Deng, Y. Zhang, X. Luo, and X. Liu, "Lightweight and expressive fine-grained access control for healthcare internet-of-things," *IEEE Transactions on Cloud Computing*, 2019, [Online].

[14] N. Attrapadung and H. Imai, "Dual-policy attribute based encryption," in *ACNS*, vol. 5536, 2009, pp. 168–185.

[15] I. Damgård, H. Haagh, and C. Orlandi, "Access control encryption: Enforcing information flow with cryptography," in *TCC*, 2016, pp. 547–576.

[16] S. Kim and D. J. Wu, "Access control encryption for general policies from standard assumptions," in *ASIACRYPT*, 2017, pp. 471–501.

[17] Q. M. Malluhi, A. Shikfa, and V. C. Trinh, "A ciphertext-policy attribute-based encryption scheme with optimized ciphertext size and fast decryption," in *AsiaCCS*, 2017, pp. 230–240.

[18] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *CCS*, 2006, pp. 89–98.

[19] A. B. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *EUROCRYPT*, 2011, pp. 568–588.

[20] Y. Rouselakis and B. Waters, "Practical constructions and new proof methods for large universe attribute-based encryption," in *CCS*, 2013, pp. 463–474.

[21] N. Attrapadung, B. Libert, and E. de Panafieu, "Expressive key-policy attribute-based encryption with constant-size ciphertexts," in *PKC*, vol. 6571, 2011, pp. 90–108.

[22] T. Okamoto and K. Takashima, "Fully secure unbounded inner-product and attribute-based encryption," in *ASIACRYPT*, 2012, pp. 349–366.

[23] A. Sahai, H. Seyalioglu, and B. Waters, "Dynamic credentials and ciphertext delegation for attribute-based encryption," in *CRYPTO*, 2012, pp. 199–217.

[24] S. Agrawal and M. Chase, "FAME: fast attribute-based message encryption," in *CCS*, 2017, pp. 665–682.

[25] S. Xu, G. Yang, Y. Mu, and R. H. Deng, "Secure fine-grained access control and data sharing for dynamic groups in the cloud," *IEEE Trans. Information Forensics and Security*, vol. 13, no. 8, pp. 2101–2113, 2018.

[26] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *IEEE S&P*, 2007, pp. 321–334.

[27] L. Cheung and C. C. Newport, "Provably secure ciphertext policy ABE," in *CCS*, 2007, pp. 456–465.

[28] X. Liang, Z. Cao, H. Lin, and D. Xing, "Provably secure and efficient bounded ciphertext policy attribute based encryption," in *ASIACCS*, 2009, pp. 343–352.

[29] J. Herranz, F. Laguillaumie, and C. Ràfols, "Constant size ciphertexts in threshold attribute-based encryption," in *PKC*, 2010, pp. 19–34.

[30] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *PKC*, vol. 6571, 2011, pp. 53–70.

[31] Y. Zhang, D. Zheng, X. Chen, J. Li, and H. Li, "Computationally efficient ciphertext-policy attribute-based encryption with constant-size ciphertexts," in *ProvSec*, 2014, pp. 259–273.

[32] ——, "Efficient attribute-based data sharing in mobile clouds," *Pervasive and Mobile Computing*, vol. 28, pp. 135–149, 2016.

[33] S. Xu, G. Yang, and Y. Mu, "Revocable attribute-based encryption with decryption key exposure resistance and ciphertext delegation," *Information Sciences*, vol. 479, pp. 116–134, 2018.

[34] J. Ning, Z. Cao, X. Dong, K. Liang, H. Ma, and L. Wei, "Auditable σ-time outsourced attribute-based encryption for access control in cloud computing," *IEEE Trans. Information Forensics and Security*, vol. 13, no. 1, pp. 94–105, 2018.

[35] A. Miyaji and P. V. X. Tran, "Constant-ciphertext-size dual policy attribute based encryption," in *CCS*, 2012, pp. 400–413.

[36] Y. S. Rao and R. Dutta, "Computationally efficient dual-policy attribute based encryption with short ciphertext," in *ProvSec*, 2013, pp. 288–308.

[37] N. Attrapadung and S. Yamada, "Duality in ABE: converting attribute based encryption for dual predicate and dual policy via computational encodings," in *CT-RSA*, 2015, pp. 87–105.

[38] J. W. Byun, H. S. Rhee, H. Park, and D. H. Lee, "Off-line keyword guessing attacks on recent keyword search schemes over encrypted data," in *VLDB SDM*, 2006, pp. 75–83.

[39] S. Qiu, J. Liu, Y. Shi, and R. Zhang, "Hidden policy ciphertext-policy attribute-based encryption with keyword search against keyword guessing attack," *SCIENCE CHINA Information Sciences*, vol. 60, no. 5, pp. 052 105:1–052 105:12, 2017.

Shengmin Xu received the Ph.D. degree in Cryptography from University of Wollongong, Australia, in 2018. He is currently a research fellow at Singapore University of Technology and Design, Singapore. Previously, he was a research fellow at Singapore Management University, Singapore. His research interests include information security, cloud computing and blockchain.

Jianting Ning received the Ph.D. degree from the Department of Computer Science and Engineering, Shanghai Jiao Tong University in 2016. He is currently a research fellow at School of Information Systems, Singapore Management University. He will be a Professor with the Fujian Provincial Key Laboratory of Network Security and Cryptology, School of Mathematics and Computer Science, Fujian Normal University, China. Previously, he was a research fellow at Department of Computer Science, National University of Singapore. His research interests include applied cryptography and information security, in particular, public key encryption, secure and privacy-preserving computation.

Yingjiu Li is currently a Ripple Professor in the Computer and Information Science Department at the University of Oregon. His research interests include IoT Security and Privacy, Mobile and System Security, Applied Cryptography and Cloud Security, and Data Application Security and Privacy. He has published over 140 technical papers in international conferences and journals, and served in the program committees for over 80 international conferences and workshops, including top-tier cybersecurity conferences and journals.

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 at Singapore Management University. He has published over 80 research articles including AsiaCCS, IEEE TSC, Computer Networks, IEEE Internet of Things Journal, Computers & Security. His research interests include public key cryptography, cloud security and wireless network security.

Guowen Xu received his B.S. degree in information and computing science from Anhui University of Architecture in 2014. Currently, he is a Ph.D. student at the School of Computer Science and Engineering, University of Electronic Science and Technology of China , Cchina. His research interests include Cryptography, Searchable Encryption, and the Privacy-preserving Deep Learning.

Xinyi Huang received his Ph.D. degree from the School of Computer Science and Software Engineering, University of Wollongong, Australia, in 2009. He is currently a Professor at the Fujian Provincial Key Laboratory of Network Security and Cryptology, College of Mathematics and Informatics, Fujian Normal University, China. His research interests include cryptography and information security. He has published over 130 research papers in refereed international conferences and journals. His work has been cited more than 6000 times at Google Scholar. He is in the Editorial Board of International Journal of Information Security. He has served as the program/general chair or program committee member in over 120 international conferences.

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 are in the areas of 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 IEEE Security & Privacy Magazine, IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Information Forensics and Security, Journal of Computer Science and Technology, and Steering Committee Chair of the ACM Asia Conference on Computer and Communications Security. He is an IEEE Fellow.

Appendix

A. Security Proof for Theorem 1

Proof. Suppose there exists a polynomial-time adversary $\mathcal{A}$ that can break our scheme in IND-CPA model with a non-negligible advantage. Then we can build a simulator $\mathcal{B}$ that can break the decisional BDH assumption with a non-negligible advantage. Specially, $\mathcal{B}$ receives the tuple $(p, \mathbb{G}, \mathbb{G}_T, e, g, A = g^a, B = g^b, C = g^c, T)$ to guess $T = e(g,g)^{abc}$ or a random term interacting with $\mathcal{A}$.

Setup. $\mathcal{B}$ randomly picks $\beta \in \mathbb{Z}_p$ and two collision-resistant hash functions $\mathcal{H}_1 : \Omega_{\mathsf{snd}} \rightarrow \mathbb{G}$ and $\mathcal{H}_3 : \{0,1\}^* \rightarrow \mathbb{G}$ and returns the master public key $mpk = (p, \mathbb{G}, \mathbb{G}_T, e, B = g^b, e(B,B)^\alpha, e(A,B))$ to $\mathcal{A}$. We have the group generator $B = g^b$, and two terms α and $\beta = a/b$ because of

$$e(A,B) = e(g^a, g^b) = e(g,g)^{ab} = e(g,g)^{b^2 \cdot a/b} = e(B,B)^{a/b}.$$

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

$\mathcal{O}_{\mathcal{R}}(att)$: Let $\mathcal{L} = \{att, r, r'\}$ be a hash list. $\mathcal{A}$ queries the random oracle on the attribute att. If att has been queried before, it returns $B^{r_i} g^{r'_i}$; otherwise, $\mathcal{B}$ picks $r = 0$ and $r' \in \mathbb{Z}_p$ if $att \in \mathcal{R}^*$, picks $r, r' \in \mathbb{Z}_p$ if $att \notin \mathcal{R}^*$. Then, $\mathcal{B}$ updates a hash list $\mathcal{L} \leftarrow \mathcal{L} \cup \{att, r, r'\}$. $\mathcal{B}$ returns $B^{r_i} g^{r'_i}$ to $\mathcal{A}$.

$\mathcal{O}_{\mathsf{EKGen}}(\mathcal{S})$: $\mathcal{A}$ queries the encryption key generation oracle on the set of the sender's attributes $\mathcal{S} = (att_1, att_2, \ldots, att_k)$. $\mathcal{B}$ randomly picks $r \in \mathbb{Z}_p$ and for $i \in [k]$, it computes:

$$ek_{1,i} = B^\alpha \cdot \mathcal{H}_1(att_i)^r, \quad ek_2 = B^r.$$

$\mathcal{B}$ returns the encryption key $ek = (\mathcal{S}, \{ek_{1,i}\}_{i \in [k]}, ek_2)$ to $\mathcal{A}$.

$\mathcal{O}_{\mathsf{DKGen}}(\mathbb{R})$: $\mathcal{A}$ queries the encryption key generation oracle on the set of the receiver's policy $\mathbb{R} = (\mathbb{N}, \pi)$. According to Proposition 1 in [18], there exists a vector $\vec{\eta}$ such that $\mathbb{N}\vec{\eta} = \vec{0}$ and $\vec{1} \cdot \vec{\eta} \neq 0 (= h$, say). Let $\vec{\eta} = (\eta_1, \eta_2, \ldots, \eta_\ell)$. Finally define the vector $\vec{\eta}$ as follows:

$$\vec{\eta} = \vec{v} + \psi \vec{w}, \quad \text{where } \psi = \frac{ab - b\sum_{k=1}^{\ell} \lambda_k}{h}, \ \vec{v} = b\vec{\lambda} \text{ for } \lambda_i \in \mathbb{Z}_p.$$

If $\pi(i) \in \mathcal{R}$, we have

$$\mathbb{N}_i \vec{\eta} = \vec{v} + \frac{a - \vec{v}}{h} \cdot \vec{w} = a\mu_1 + \mu_2,$$

If $\pi(i) \notin \mathcal{R}$, we have

$$\mathbb{N}_i \vec{\eta} = b\vec{v} + \frac{ab - b\vec{v}}{h} \cdot \vec{w} = b(a\mu_1 + \mu_2),$$

where $\mu_1 = \mathbb{N}_i \vec{w} \cdot h^{-1}$ and $\mu_2 = \mathbb{N}_i (h\vec{v} - \vec{v}\vec{w})$ are commutable. For $i \in [\ell]$, $\mathcal{B}$ randomly picks $r_i \in \mathbb{Z}_p$ and computes

$$dk_{1,i} = g^{\mathbb{N}_i \vec{\eta}} \mathcal{H}_2(\pi(i))^{r_i}, \quad dk_{2,i} = g^{r_i}.$$

$\mathcal{B}$ returns the decryption key $dk = ((\mathbb{N}, \pi), \{dk_{1,i}, dk_{2,i}\}_{i \in [\ell_{\mathbb{N}}]})$ to $\mathcal{A}$.

Challenge. $\mathcal{A}$ sends $(m_0, m_1, \mathcal{S}_0, \mathcal{S}_1)$. $\mathcal{B}$ picks a random bit $b \in \{0,1\}$. Suppose $\mathcal{S}_b = (att_1, att_2, \ldots, att_k)$, $\mathcal{B}$ randomly chooses $r \in \mathbb{Z}_p$ and for $i \in [k]$, it computes

$$ek_{1,i} = g^\alpha \mathcal{H}_1(att_i)^r, \quad ek_2 = g^r.$$

to generate the encryption key $ek = (\mathcal{S}, \{ek_{1,i}\}_{i \in [k]}, ek_2)$. $\mathcal{B}$ then chooses $c_0 = m_b \cdot T$ and $c_1 = C$. $\mathcal{B}$ then simulates the rest of ciphertext components based on the encryption algorithm. Finally, $\mathcal{B}$ returns the ciphertext c to $\mathcal{A}$.

Guess. $\mathcal{A}$ submits a bit b' to $\mathcal{B}$. If $b = b'$, $\mathcal{B}$ outputs 1 denoted $T = e(g,g)^{abc}$; otherwise, it outputs 0.

The core component of our simulation is that $e(A,B) = e(g,g)^{ab}$, $c_0 = m_b \cdot T$ and $c_1 = C = g^c$. If $T = e(g,g)^{abc}$, the simulation is identical to the real scheme. If T is a random term, $\mathcal{A}$ has no advantage to break the game. It is worth to notice that the encryption algorithm has re-randomized the encryption key ek, $\mathcal{A}$ has no advantage to break the game from guessing encryption key.

There is no abort during our simulation. Hence, the advantage of $\mathcal{A}$ breaks the decisional BDH assumption is ϵ, where ϵ is the advantage of $\mathcal{A}$ breaking our proposed scheme. □

A. Security Proof for Theorem 2

Proof. Suppose there exists a polynomial-time adversary $\mathcal{A}$ that can break our scheme in EU-CMA model with a non-negligible advantage. Then we can build a simulator $\mathcal{B}$ that can break the computational BDH assumption with a non-negligible advantage. Specially, $\mathcal{B}$ receives the tuple $(p, \mathbb{G}, \mathbb{G}_T, e, g, A = g^a, B = g^b, C = g^c)$ to output $e(g,g)^{abc}$ interacting with $\mathcal{A}$.

Setup. $\mathcal{B}$ randomly picks $\beta \in \mathbb{Z}_p$ and two collision-resistant hash functions $\mathcal{H}_2 : \Omega_{\mathsf{rcv}} \rightarrow \mathbb{G}$ and $\mathcal{H}_3 : \{0,1\}^* \rightarrow \mathbb{G}$ and returns the master public key $mpk = (p, \mathbb{G}, \mathbb{G}_T, e, B = g^b, e(A,B), e(B,B)^\beta)$ to $\mathcal{A}$. We have the group generator $B = g^b$, and two terms $\alpha = a/b$ and β because of

$$e(A,B) = e(g^a, g^b) = e(g,g)^{ab} = e(g,g)^{b^2 \cdot a/b} = e(B,B)^{a/b}.$$

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

$\mathcal{O}_{\mathcal{S}}(att)$: Let $\mathcal{L} = \{att, r, r'\}$ be a hash list. $\mathcal{A}$ queries the random oracle on the attribute att. If att has been queried before, it returns $B^{r_i} g^{r'_i}$; otherwise, $\mathcal{B}$ picks $r = 0$ and $r' \in \mathbb{Z}_p$ if $att \models \mathbb{R}$, picks $r, r' \in \mathbb{Z}_p$ if $att \not\models \mathbb{R}$. Then, $\mathcal{B}$ updates a hash list $\mathcal{L} \leftarrow \mathcal{L} \cup \{att, r, r'\}$. $\mathcal{B}$ returns $B^{r_i} g^{r'_i}$ to $\mathcal{A}$.

$\mathcal{O}_{\mathsf{EKGen}}(\mathcal{S})$: $\mathcal{A}$ queries the encryption key generation oracle on the set of the sender's attributes $\mathcal{S} = (att_1, att_2, \ldots, att_k)$. $\mathcal{B}$ randomly picks $r \in \mathbb{Z}_p$ and for $i \in [k]$, it computes:

$$ek_{1,i} = B^\alpha \cdot \mathcal{H}_1(att_i)^r, \quad ek_2 = B^r.$$

$\mathcal{B}$ returns the encryption key $ek = (\mathcal{S}, \{ek_{1,i}\}_{i \in [k]}, ek_2)$ to $\mathcal{A}$.

$\mathcal{O}_{\mathsf{DKGen}}(\mathbb{R})$: $\mathcal{A}$ queries the encryption key generation oracle on the set of the receiver's policy $\mathbb{R} = (\mathbb{N}, \pi)$. Due to β is known, $\mathcal{B}$ run the decryption key generation algorithm and returns $ek = (\mathcal{S}, \{ek_{1,i}\}_{i \in [k]}, ek_2)$ to $\mathcal{A}$.

$\mathcal{O}_{\mathsf{Enc}}(ek, \mathbb{R}, \mathbb{S}, m)$: $\mathcal{A}$ queries the encryption oracle on the set of the encryption key ek, a set of receiver's attribute $\mathcal{R}$ and a set of sender's attribute and a message m. Due to all inputs are known, $\mathcal{B}$ run the encryption algorithm and returns $c = ((\mathcal{S}, \mathcal{R}'), c_0, c_1, \{c_{2,i}\}_{i \in [l]}, c_3, c_4, \{c_{5,i'}\}_{i' \in [k']})$ to $\mathcal{A}$.

Guess. $\mathcal{A}$ submits a ciphertext c to $\mathcal{B}$, where $c = ((\mathcal{S}, \mathcal{R}'), c_0, c_1, \{c_{2,i}\}_{i \in [l]}, c_3, c_4, \{c_{5,i'}\}_{i' \in [k']})$. Parse the access structure of the sender $\mathbb{S} = (\mathbb{M}, \rho)$, where $\mathbb{M} \in \mathbb{Z}_p^{\ell_{\mathbb{M}} \times n_{\mathbb{M}}}$ is a matrix and $\rho : [\ell_{\mathbb{M}}] \rightarrow \Omega_{\mathsf{snd}}$ is a mapping function. $\mathcal{B}$ randomly picks $\vec{x} = (c, x_2, \ldots, x_{n_{\mathbb{M}}})^\perp \in \mathbb{Z}_p^{n_{\mathbb{M}} \times 1}$ and computes $\vec{\kappa} = (\kappa_1, \kappa_2, \ldots, \kappa_{\ell_{\mathbb{M}}}) = \mathbb{M}\vec{x}$. Let I be the set $s.t.$ $I = \{i | i \in [\ell_{\mathbb{M}}], \rho(i) = \mathcal{S}\}$. It takes terms $\{\omega_i\}_{i \in I}$ $s.t.$ $\sum_{i \in I} \omega \mathbb{M}_i = (1, 0, \ldots, 0)$ and computes:

$$\prod_{i \in I} \left(\frac{e(c_{5,i}, g^b)}{e(\mathcal{H}_1(att_{\mathsf{snd},i}), c_3) \cdot e(\mathcal{H}_3(c_{1\text{-}4}), c_4)} \right)^{\kappa_i \omega_i} = e(g,g)^{abc}.$$

$\mathcal{B}$ returns $e(g,g)^{abc}$. Therefore, $\mathcal{B}$ can break computational BDH tuple.

There is no abort during our simulation. Hence, the advantage of $\mathcal{A}$ breaks the computational BDH assumption is ϵ, where ϵ is the advantage of $\mathcal{A}$ breaking our proposed scheme. □