FABRIC

Fast and secure unbounded cross-system encrypted data sharing in cloud computing

Wang, Lili; Lin, Ye; Yao, Ting; Xiong, Hu; Liang, Kaitai

DOI
10.1109/TDSC.2023.3240820

Publication date
2023

Document Version
Final published version

Published in
IEEE Transactions on Dependable and Secure Computing

FABRIC: Fast and Secure Unbounded Cross-System Encrypted Data Sharing in Cloud Computing

Lili Wang, Ye Lin, Ting Yao, Hu Xiong, *Senior Member, IEEE*, and Kaitai Liang

***Abstract*—Existing proxy re-encryption (PRE) schemes to secure cloud data sharing raise challenges such as supporting the heterogeneous system efficiently and achieving the unbounded feature. To address this problem, we proposed a fast and secure unbounded cross-domain proxy re-encryption scheme, named FABRIC, which enables the delegator to authorize the semi-trusted cloud server to convert one ciphertext of an identity-based encryption (IBE) scheme to another ciphertext of an attribute-based encryption (ABE) scheme. As the first scheme to achieve the feature mentioned above, FABRIC not only enjoys constant computation overhead in the encryption, decryption, and re-encryption phases when the quantity of attributes increases, but is also unbounded such that the new attributes or roles could be adopted into the system anytime. Furthermore, FABRIC achieves adaptive security under the decisional linear assumption (DLIN). Eventually, detailed theoretical and experimental analysis proved that FABRIC enjoys excellent performance in efficiency and practicality in the cloud computing scenario.**

***Index Terms*—Cloud computing, cross-domain data sharing, encrypted data sharing, privacy, security.**

I. Introduction

As an encouraging computing paradigm, cloud computing could reliably provide massive storage space at low cost [1], [2]. By using cloud services provided by cloud service providers (CSP), such as Amazon [3], Microsoft [4], and Apple [5], cloud users could conveniently store, process, and share data anytime and anywhere without any restrictions [6]. For example, in a hospital, the smart medical devices could collect the health data of patients and upload it to the cloud server. By using mobile devices to access patients' electronic health data stored in the cloud, even doctors on the trip could make timely diagnoses. However, since the cloud services are usually maintained by untrusted CSPs [7], [8], users' concern about shared private data being accessed by unauthorized users or malicious CSPs has become the major obstacle to the popularization of cloud computing [9], [10], [11]. Therefore, the public key encryption schemes are adopted to protect the shared data in cloud computing [12], [13]. Specifically, before uploading these data to the cloud server, the data owner utilizes the data user's public key to encrypt the shared data [14], [15]. In this way, only the legitimate data user who holds the corresponding secret key could decrypt the ciphertext. However, with the quantity of data users growing, the data owner needs to utilize the public key of every data user to encrypt and upload data separately, which brings heavy computation and communication burden to the data owner [16].

By achieving the secure access delegation, the primitive of proxy re-encryption (PRE) seems to be one practicable approach to the above-mentioned problems [17], [18], [19], [20]. Concretely, PRE enables a semi-trusted proxy to transform the ciphertext encrypted by the public key of delegator to the one encrypted by the public key of delegatee. During this process, any information about the plaintext or secret keys would not be obtained by the proxy. With the support of PRE, the data owner can encrypt data with his own public key and upload it to the cloud server. After being authorized by the data owner, the cloud server is able to transform ciphertext for each data user, who could in turn utilize his secret key to decrypt the re-encrypted ciphertext. Subsequently, various PRE schemes are proposed to meet the demands in diverse scenarios [21]. For instance, the identity-based PRE (IB-PRE) schemes [22], [23], [24] are presented with the considerable efficiency and freeness of public key certificates, thereby being in the good graces of those users with resource-constrained devices (e.g., mobile devices). The attribute-based PRE (AB-PRE) schemes [25], [26], [27] light the path for flexible secure data sharing, due to the advantages in achieving fine-grained access control. Nevertheless, it is worth noting that these schemes are designed for the single cryptosystem and not suitable for the heterogeneous scenario. By way of illustration, the patient equipped with the smart medical devices may adopt the identity-based cryptosystem (IBC) to encrypt the data collected from him/herself, while the decryption ability for the physicians could be determined by their corresponding

Manuscript received 11 June 2022; revised 17 January 2023; accepted 20 January 2023. Date of publication 31 January 2023; date of current version 13 November 2023. This work was supported in part by the National Natural Science Foundation of China under Grant U1936101, in part by the Open Fund of Advanced Cryptography and System Security Key Laboratory of Sichuan Province under Grant SKLACSS-202102, in part by the European Union's Horizon Research and Innovation Programme under Grant 952697 (ASSURED), 101021727 (IRIS) and 101070052 (TANGO). *(Corresponding author: Hu Xiong.)*

Lili Wang, Ye Lin, and Ting Yao are with the School of Information and Software Engineering, Network and Data Security Key Laboratory of Sichuan Province, University of Electronic Science and Technology of China, Chengdu 610054, China (e-mail: wanglili.uestc@gmail.com; ylin@alu.uestc.edu.cn; yaoting.uestc@gmail.com).

Hu Xiong is with the School of Information and Software Engineering, Network and Data Security Key Laboratory of Sichuan Province, University of Electronic Science and Technology of China, Chengdu 610054, China, and also with Advanced Cryptography and System Security Key Laboratory of Sichuan Province, Chengdu 610025, China (e-mail: xionghu.uestc@gmail.com).

Kaitai Liang is with the Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, 2628 Delft, CD, The Netherlands (e-mail: Kaitai.Liang@tudelft.nl).

This article has supplementary downloadable material available at https://doi.org/10.1109/TDSC.2023.3240820, provided by the authors.

Digital Object Identifier 10.1109/TDSC.2023.3240820

attributes to ensure the flexible and secure one-to-many data sharing. Naturally, the IBC and the attribute-based cryptosystem (ABC) are respectively adopted by the patient and the physicians. Due to the differences between the two encryption systems, it is difficult for the physician to obtain the patient's healthcare data from the ciphertext generated with the patient's identity, which makes it impossible for users who adopt the IBC to capture confidentiality and scalability at the same time. Furthermore, it also brings inconvenience for physicians to share the patient data collected by smart devices for timely diagnoses. Moreover, when an entity wants to adopt ABC for flexible data sharing, it will bring colossal computation and communication overhead to the resource-constrained devices in the system. Obviously, it is difficult for the existing solutions to convert the ciphertext in the high-efficiency IBC to the ciphertext in the flexible ABC, which is also the paramount obstacle to balancing the efficiency of encryption and the flexibility of secure data sharing.

However, in traditional ABE schemes, the size of the public parameters and the computation overhead of the setup algorithm is closely related to the number of attributes or the size of policies [28]. Once the number of attributes or the size of policies increases, both the communication and computation overheads of the setup algorithm will increase accordingly, which inevitably brings heavy burden. To solve this problem, the unbounded feature is first proposed by Lewko and Waters [29]. By using the bilinear entropy expansion lemma, the polynomial amount of entropy can be generated by using constant-size public parameters. Then, the entropy can be utilized to transform a bounded ABE scheme into an unbounded one, which means the setup algorithm is independent of the attribute number or the policy size. In this way, the computation and communication cost of the setup algorithm in the unbounded ABE mechanism can remain constant. Inspired by this method, a series of works are proposed subsequently [30], [31], [32], [33], [34]. So far, the cross-system featured with unbounded property has not been treated yet in the literature.

Driven by the challenges mentioned above, we presented FABRIC—a fast and secure unbounded cross-domain proxy re-encryption scheme. By carefully designing the proxy re-encryption key to integrate IBE with ABE, we can achieve the secure access delegation for the cross-domain data sharing. Specifically, in FABRIC, data is encrypted with the identity of the delegator (i.e., the public key of the delegator). An authorized proxy, who holds the re-encryption key, could transform this ciphertext to the re-encrypted ciphertext corresponding to the access policy. In this way, only in case the attribute set in the delegatee's secret key matches the access control policy in the re-encrypted ciphertext, the ciphertext could be successfully decrypted by the delegatee. Moreover, FABRIC adopts the monotone spanning program (MSP) as the access structure, and achieves unbounded feature based on the bilinear entropy extension lemma. Therefore, FABRIC is more expressive – because there is no limitation of the number of attributes or the size of policies, but also more efficient – due to the constant communication and computation overheads of the setup algorithm. Through the dual system encryption technology, FABRIC can be adaptively secure under the decisional linear assumption (DLIN). Concretely, we briefly summarize our contributions as follows:

1) We presented a fast and secure unbounded cross-domain proxy re-encryption scheme, namely FABRIC, which allows the proxy to transform the ciphertext encrypted with the delegator's identity to the ciphertext embedded with an access policy. Specifically, by using re-encryption key generated under the access policy to re-encrypt the ciphertext, FABRIC bridges the gap between IBE and ABE. Moreover, with the increase of the quantity of attributes, the re-encryption and decryption computation cost can still keep constant.
2) FABRIC achieves the unbounded feature by making the system setup algorithm and the size of public parameters independent of the length of attributes. Concretely, FABRIC uses the constant-size public parameters to generate any polynomial amount of entropy. In this way, the ABE setup algorithm should put no restriction on the length of the attributes or the size of the policies that will be used in the ciphertexts and keys. Hence, it is possible for FABRIC to be dynamically adjusted for suiting future application scenarios.
3) We formally proved that FABRIC is adaptively secure with the help of dual system encryption methodology [31]. Moreover, the experiment and the concrete theoretical analysis are conducted, which indicates that FABRIC enjoys excellent performance in efficiency and practicality in real-life scenarios.

A. Related Works

Blaze et al. [17] first introduced the notion of proxy re-encryption (PRE). In this primitive, an authorized semi-trust proxy is able to transform the ciphertext encrypted by the public key of the delegator to the ciphertext under the public key of the delegatee. Later, Ateniese et al. [18] not only formalized the definition of PRE, but also constructed a pairing-based PRE scheme and applied it to the distributed file system. Shao et al. [19] presented a pairing-free PRE scheme featured with CCA security and collusion-resistance. Ateniese et al. [20] proposed a key-private PRE scheme, in which the proxy cannot learn the identity of the delegator and delegatee from the ciphertext or re-encryption key. However, the above-mentioned schemes all rely on the public key infrastructure (PKI), which brings heavy computation and storage computation costs in certificate management.

Inspired by the identity-based cryptosystem (IBC), Green and Ateniese [35] introduced the first identity-based PRE (IB-PRE) scheme based on the bilinear map. Different from the traditional public key cryptography, this primitive enables users to use their identity as the public key. And a trusted entity called private key generator (PKG) is in charge of creating the corresponding secret key, thereby eliminating the burden of public key certificates. Subsequently, Chu and Tzeng [36] presented an IB-PRE scheme featured with short secret keys and ciphertexts, which achieves CPA security in the standard model. Ren et al. [37] proposed a hierarchical IB-PRE scheme featured with CCA security. Liang et al. [38] constructed a revocable IB-PRE and applied it into

TABLE I
FREQUENTLY USED NOTATIONS

Notation	Description
k	the security parameter
$\mathcal{H}, \mathcal{F}$	the hash function
$Param$	the public parameters
MSK	the master secret key
S	the attribute set
M	the access matrix M
$(\mathrm{M})_i$	the i-th row of the access matrix M
γ_i	the coefficient in i-th row
π	the mapping from the row of M to the attribute
(M, π)	the MSP access policy structure
$\top$	the matrix transpose method
m	the vector quantity
SK_s	the secret key in ABE
ISK_{id}	the secret key in IBE
msg	the message
id	the user's identity
$RK_{id \to S}$	the re-encryption key
$C_{id \to S}$	the re-encryption ciphertext

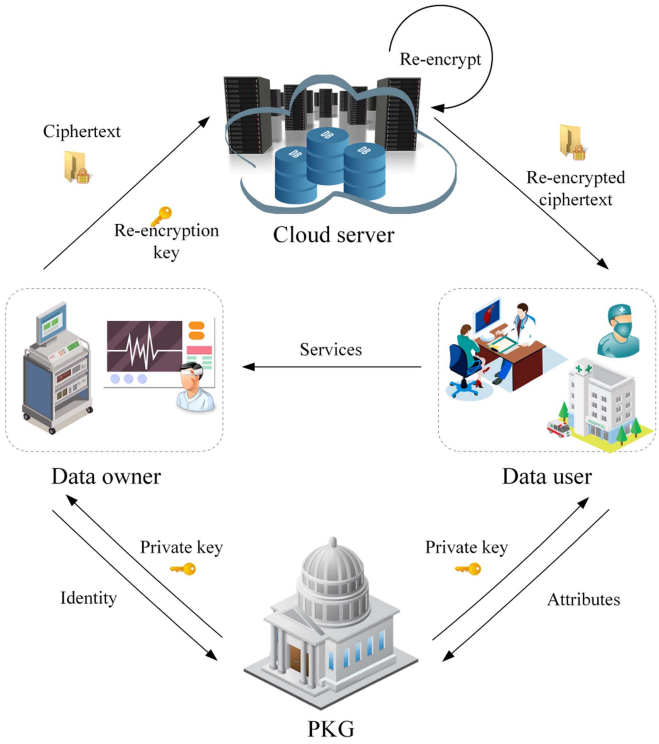

Fig. 1. System model.

cloud computing to build a secure data sharing system. To convert the data encrypted for a single user into a ciphertext that can be decrypted by multiple users simultaneously, Deng et al. [24] introduced an IB-PRE scheme, which enables the ciphertext transformation from the IBE scheme to the identity-based broadcast encryption (IBBE) scheme (denoted as IBET in the following). Specifically, IBBE allows a set of identities to be used to encrypt data during the encryption phase, and the ciphertext can be decrypted by the corresponding private key of any identity in the set of identities. However, IBC cannot achieve flexible and fine-grained access control to meet the one-to-many communication demands in real-life scenarios [39], [40].

Combining the advantage of attribute-based encryption (ABE), Liang et al. [41] presented the attribute-based PRE (AB-PRE) scheme, which allows the delegator to authorize a proxy to transform the ciphertext encrypted with an access policy into a ciphertext encrypted with another access policy. Only when the receiver holds the secret key with attributes which match the access policy formulated in the ciphertext, could the receiver decrypt ciphertext successfully. After that, Yu et al. [42] constructed an AB-PRE scheme for secure data sharing, which achieves attribute revocation. Liang et al. [26] presented a ciphertext-policy AB-PRE scheme for cloud data sharing, which not only supports more expressive access policies but also achieves adaptive security in the random oracle model (denoted as CPAS in the following). Ge et al. [27] introduced a CCA-secure key-policy AB-PRE scheme, which is proved to be adaptively secure in the standard model.

It is noteworthy that the above-mentioned schemes are all designed for the single cryptosystem. Due to the heterogeneous nature of cloud computing, different entities may adopt various cryptosystems [43], [44]. To achieve secure cross-domain data sharing, Mizuno and Doi [45] constructed a hybrid AB-PRE scheme(denoted as CAB in the following), where the ciphertext encrypted with an access policy could be transformed to the ciphertext of an IBE scheme. Nevertheless, there is the limitation on the size of policies or attributes formulated in secret keys or ciphertexts, which not only brings additional computation and communication burden but also limits the expressiveness of the access policy. Jiang et al. [46] presented a cross-domain encryption switching mechanism that can transform a PKI-based ciphertext into an IBE one, and vice versa. This mechanism requires a full-trusted third party to provide a transformation key for each conversion, and the ABE has not been considered in this work. Liu et al. [47] proposed an efficient privacy-preserving outsourced calculation framework with multiple keys (EPOM) to support integer operations on the data encrypted under different public keys. However, it does not provide a secure access delegation mechanism. That is to say, EPOM does not allow a semi-trusted proxy server to translate the ciphertext into another ciphertext. Lately, based on the linear secret sharing scheme (LSSS), Deng et al. [48] proposed an unbounded cross-domain AB-PRE (denoted as LUC in the following), which transforms the ciphertext of ABE to the ciphertext of IBE. To the best of our knowledge, our work is the first to transform the ciphertext of IBE to the ciphertext of ABE, but also is constructed based on the asymmetric pairing and achieves adaptive security.

II. SYSTEM FRAMEWORK AND MODEL

A. System Model

There are four entities involved in FABRIC as shown in Fig. 1:

- *Private Key Generator (PKG):* As the trusted entity in the system, PKG is not only responsible for initializing system parameters, but also for generating private keys for each user.

- *Data Owner:* The data owner takes the identity as his public key, while the corresponding private key is cteated by PKG. The data owner could utilize his own private key and the access policy to compute the re-encryption key, and sends it to the cloud server. Concretely, the data would be encrypted locally with the identity of the data owner, and the ciphertext would be sent and kept in the cloud server for sharing.
- *Cloud Server:* The cloud server provides massive storage space and efficient computing capabilities for ciphertext storage and sharing. On receiving the ciphertext and re-encryption key from data owner, the cloud server could transform the ciphertext for data user.
- *Data User:* The data user can obtain valuable information by studying shared data, thereby providing service to the data owners. The secret key held by the data user is embedded with an attribute set. Only the attributes matches the access policy in re-encrypted ciphertext, could the data user complete decryption operation.

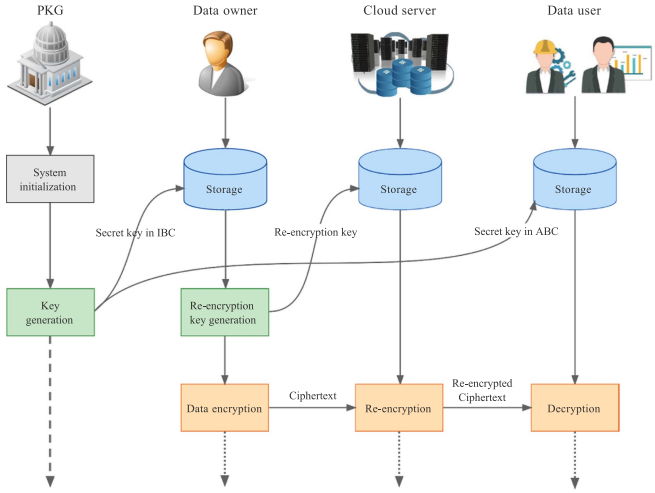

Fig. 2. The flowchart of cross-system encrypted data sharing.

B. Syntax of FABRIC

The FABRIC system involves an ABE system, an IBE system, and necessary algorithms for re-encryption. Concretely, the private key generation algorithm is designed for IBE system ($\mathsf{KeyGen}_{\mathsf{IBE}}$) and ABE system ($\mathsf{KeyGen}_{\mathsf{ABE}}$) respectively. Smart devices featured with limited resources will invoke the $\mathsf{Encrypt}$ algorithm in the IBE system to generate the ciphertext on the message to be encrypted under its own identity. To realize the expressive fine-grained access control for the ciphertext in IBE system, the idea of proxy re-encryption is introduced to bridge the gap between the ABE system and the IBE system. With regard to this, the $\mathsf{ReKeyGen}$ algorithm is proposed to generate a re-encryption key associated with the private key in the IBE system and an access policy in the ABE system. Meanwhile, the ReEnc algorithm is designed to convert one IBE ciphertext to another ABE ciphertext with the support of the re-encryption key generated by the $\mathsf{ReKeyGen}$ algorithm. In this way, the $\mathsf{Decrypt}_{\mathsf{ABE}}$ algorithm is also put forward to enable the ABE user to decrypt a re-encrypted ciphertext in case the attribute of this user matches the access policy embedded in this ciphertext. There are eight algorithms involved in FABRIC as bellow. For convenience, the system parameters $Param$ in all the following algorithm inputs are omitted. To better describe the proposed scheme, the frequently used notations involved in the proposed scheme are provided in Table I.

- $\mathsf{Setup}(1^{\mathsf{k}})$: After receiving the security parameter k, this algorithm returns the master secret key MSK as well as the system parameters $Param$.
- $\mathsf{KeyGen}_{\mathsf{IBE}}(MSK, id)$: After receiving MSK and the identity id of user, this algorithm returns user id's private key ISK_{id}.
- $\mathsf{KeyGen}_{\mathsf{ABE}}(MSK, S)$: After receiving MSK and user's attribute set S, this algorithm returns corresponding private key SK_S.
- $\mathsf{ReKeyGen}(ISK_{id}, (\mathbf{M}, \pi))$: On input user id's secret key ISK_{id} and a policy structure $(\mathbf{M}, \pi)$, this algorithm returns re-encryption key $RK_{id\to S}$.
- $\mathsf{Encrypt}(\mathsf{msg}, \mathsf{id})$: Taking user's identity id and the message msg as input, then return ciphertext C_{id}.
- $\mathsf{ReEnc}(C_{id}, RK_{id\to S})$: This algorithm takes the ciphertext C_{id} as well as re-encryption key $RK_{id\to S}$ as input, then returns re-encrypted ciphertext $C_{id\to S}$.
- $\mathsf{Decrypt}_{\mathsf{IBE}}(\mathsf{C_{id}}, \mathsf{ISK_{id}})$: After receiving the ciphertext C_{id} and user id's secret key ISK_{id}, this algorithm returns plaintext msg.
- $\mathsf{Decrypt}_{\mathsf{ABE}}(C_{id\to S}, SK_S)$: On input re-encrypted ciphertext $C_{id\to S}$ and the private key SK_S, this algorithm returns plaintext msg.

C. System Workflow

This section would briefly introduce the workflow of FABRIC to demonstrate how to achieve cross-system encrypted data sharing, and the corresponding flowchart is shown in Fig. 2. First, PKG runs the $\mathsf{Setup}(1^{\mathsf{k}})$ algorithm to initialize the system. Then, PKG runs the $\mathsf{KeyGen}_{\mathsf{IBE}}(MSK, id)$ algorithm with the identity to generate the secret key for the data owner in IBC. As for the data user in ABC, PKG takes the attribute set as input to run the $\mathsf{KeyGen}_{\mathsf{ABE}}(MSK, S)$ algorithm to generate the corresponding secret key. Before sharing encrypted data crossing different systems, the data owner should run the $\mathsf{ReKeyGen}(ISK_{id}, (\mathbf{M}, \pi))$ algorithm with a policy structure to generate the re-encryption key, and send it to the cloud server. To share the encrypted data with the data user, the data owner runs the $\mathsf{Encrypt}(\mathsf{msg}, \mathsf{id})$ algorithm with his own identity to encrypt the message, and sends the ciphertext to the cloud server. After receiving the ciphertext, the cloud server runs $\mathsf{ReEnc}(C_{id}, RK_{id\to S})$ algorithm with the pre-upload re-encryption key to re-encrypt the ciphertext, and distributes the re-encrypted ciphertext to authorized data users. In this way, data users could use their own secret key

to run the $\mathsf{Decrypt}_{\mathsf{ABE}}(C_{id\to S}, SK_S)$ algorithm to decrypt the re-encrypted ciphertext.

D. Threat Model and Security Goals

FABRIC confronts two types of active attacks. The first one is that the cloud servers may reveal the data uploaded by the data owner. Similar to [15], [49], the cloud servers in our proposed scheme are assumed to be "semi-trusted". It means that cloud servers will honestly perform the scheme, but they would intend to obtain as much private information as possible from the outsourced data. The other is that the unauthorized entries may attempt to access the shared data by impersonating the authorized data user. Different entries involved in our system are assumed to be curious and dishonest, and they may collude with each other to gain unauthorized access. Concretely, distinct data users featured with different attributes may collude to combine their private keys to access unauthorized data. Furthermore, the proxy server may also collude with the delegatee to decrypt the unauthorized transformed ciphertext. Considering these realistic attacks, the FABRIC is desirable to capture the following security goals.

1) Data confidentiality: If the data is encrypted before uploading to the cloud server, only the authorized user who holds the corresponding private key could decrypt it. Cloud servers and attackers should not be able to decrypt the ciphertext.
2) Controllable authorization: Data users should be considered authorized and able to decrypt the re-encrypted ciphertext only if the attribute set matches the access policy in the re-encrypted ciphertext. Unauthorized users and the proxy server should not be able to recover the message from the re-encrypted ciphertext.

E. Security Model

The security model of FABRIC could be formally defined via IND-CPA game, and $\mathcal{A}$, $\mathcal{C}$ represent the adversary and challenger, respectively.

Setup: $\mathcal{C}$ runs $\mathsf{FABRIC.Setup}(1^{\mathsf{k}}) \to (Param, MSK)$ and transfers $Param$ to $\mathcal{A}$.

Phase 1: $\mathcal{A}$ could query three types of request, and the challenger $\mathcal{C}$ can response as below:

- $\mathsf{KeyGen}_{\mathsf{IBE}}$: $\mathcal{A}$ queries with an identity id, $\mathcal{C}$ runs $\mathsf{FABRIC.KeyGen}_{\mathsf{IBE}}(MSK, id) \to ISK_{id}$ and transfers it to $\mathcal{A}$.
- $\mathsf{KeyGen}_{\mathsf{ABE}}$: $\mathcal{A}$ queries with an attribute set S, $\mathcal{C}$ runs $\mathsf{KeyGen}_{\mathsf{ABE}}(MSK, S) \to SK_S$ and returns it to $\mathcal{A}$.
- $\mathsf{ReKeyGen}$: $\mathcal{A}$ queries with an identity id and a policy structure $(\mathbf{M}, \pi)$, $\mathcal{C}$ first performs secret key query of user id on behalf of $\mathcal{A}$. After that, $\mathcal{C}$ runs the $\mathsf{FABRIC.ReKeyGen}(ISK_{id}, (\mathbf{M}, \pi)) \to RK_{id\to S}$ and returns it to $\mathcal{A}$.

Challenge: $\mathcal{A}$ first chooses an identity id^* with the following restrains:

- Case 1. The secret key and re-encryption key of identity id^* are not queried during *Phase 1*.
- Case 2. The secret key of user id^* is queried during *Phase 1*. At the same time, for the re-encryption keys of identity id^* and policy structures $(\mathbf{M}^*, \pi^*)$ that have been queried in *Phase 1*, there is no attribute set S which has been queried in *Phase 1*, that satisfies the policy structure $(\mathbf{M}^*, \pi^*)$.

For Case 1, the adversary $\mathcal{A}$ chooses two message msg_0, msg_1, and sends messages with the identity id^* to challenger $\mathcal{C}$. After that, $\mathcal{C}$ samples $v \in \{0, 1\}$ and runs $\mathsf{FABRIC.Encrypt}(\mathsf{msg_v}, \mathsf{id}^*)$ to obtain ciphertext $C^*_{id^*}$. Eventually, $\mathcal{C}$ returns $C^*_{id^*}$ to $\mathcal{A}$.

For Case 2, $\mathcal{A}$ chooses two message msg_0, msg_1, and sends messages with the identity id^* and the policy structure $(\mathbf{M}^*, \pi^*)$ to challenger $\mathcal{C}$. Then, the challenger $\mathcal{C}$ performs the same operation as Case 1 to obtain the ciphertext $C^*_{id^*}$. After that, $\mathcal{C}$ runs $\mathsf{FABRIC.ReEnc}(C^*_{id^*}, RK_{id^*\to S^*})$ to obtain the re-encrypted ciphertext $C^*_{id^*\to S^*}$. Finally, $\mathcal{C}$ returns $C^*_{id^*\to S^*}$ to $\mathcal{A}$.

Phase 2: Except for the limitations of the *Challenge* phase, $\mathcal{C}$ repeats as the same as *Phase 1*.

Guess: Eventually, $\mathcal{A}$ makes the guess v' of v. If $v' = v$, $\mathcal{A}$ wins this game.

III. Building Blocks

A. Access Structures

Definition 1. Let $\mathcal{U}$ and $\mathbb{A} \in 2^{\mathcal{U}} \setminus \{0\}$ denote the attribute universe and the access structure, respectively. And only the sets in $\mathbb{A}$ are the authorized sets. Furthermore, only if $E \in \mathbb{A}$, $E \subseteq F$, and $F \in \mathbb{A}$ holds, $\mathbb{A}$ is monotone where $\forall E, F \subseteq \mathcal{U}$.

Definition 2. Let $\mathcal{U}$ denote the attribute universe, and the monotonic span program (MSP) is composed of a mapping $\pi : \{1, \ldots, m\} \to \mathcal{U}$ and a matrix $\mathbf{M} \in \mathbb{Z}_p^{m\times n}$. Lewko et al. [50] presented an effective way to generate the MSP, where the element in $\mathbf{M}$ is either 0, 1 or -1. Suppose there is an attribute set S and $I = \{i | i \in (1, \ldots, m), \pi(i) \in S\}$. If S matches $(\mathbf{M}, \pi)$, we can compute the coefficients $\{\gamma_i\}_{i\in I}$ as

$$\sum_{i\in I} \gamma_i (\mathbf{M})_i = (1, 0, \ldots, 0).$$

It is noteworthy that $(\mathbf{M})_i$ denotes $\mathbf{M}$'s ith row. Moreover, the coefficients are either 0 or 1 according to [50].

B. Asymmetric Prime-Order Bilinear Group

Definition 3. Let $\mathcal{G}(k)$ denote the asymmetric pairing group generator. When receiving the security parameter k, the generator returns $(p, g, \mathbb{G}, h, \mathbb{H}, \mathbb{G}_T, e)$. It is noteworthy that g, h denote the generators of groups $\mathbb{G}, \mathbb{H}$, and p represents the prime order of $\mathbb{G}, \mathbb{H}, \mathbb{G}_T$. If the properties below hold,$e : \mathbb{G} \times \mathbb{H} \to \mathbb{G}_T$ can be regarded as a bilinear map:

- *Bilinear:* $\forall x, y \in \mathbb{Z}_p$, $e(g^x, h^y) = e(g, h)^{xy}$.
- *Non-Degenerate:* For any $g_1 \in \mathbb{G}$ and $h_1 \in \mathbb{H}$, there always holds $e(g_1, h_1) \neq 1$.

Moreover, we use the form of $[a]_1$, $[b]_2$, $[c]_T$ to represent $g^a, h^b, e(g, h)^c$, respectively. Then, set $\mathbf{d} = (d_1, d_2, \ldots, d_n)$, and use the form of $[\mathbf{d}]_1$ to represent $(g^{d_1}, g^{d_2}, \ldots, g^{d_n})$. In the same way, $[\mathbf{M}]_1$ for the matrix $\mathbf{M}$ is defined, and $[\mathbf{E}^\top \mathbf{F}]_T$ represents $e([\mathbf{E}]_1, [\mathbf{F}]_2)$.

IV. The Construction of FABRIC

The main obstacle to achieving secure cross-domain sharing is that users in IBE and ABE systems hold different public/secret key pairs, and users in the ABE system cannot decrypt the ciphertext in the IBE system. To solve this problem, FABRIC includes IBE and ABE schemes that share public parameters, bur also introduces the ReKeyGen algorithm and ReEnc algorithm. The ReKeyGen algorithm embeds the access policy into the re-encryption key, and the ReEnc algorithm can convert the IBE ciphertext to the ABE ciphertext with the help of re-encryption key, thus bridging IBE and ABE. The concrete construction of FABRIC is introduced below:

- *System Initialization*: PKG runs the Setup algorithm with a security parameter k to generate the public parameters $Param$ and the master secret key MSK, and the detail is described in Algorithm 1.

Algorithm 1: Setup(k).

Input: the security parameter k;
Output: the public parameters $Param = (e, g, h, H_1, H_2, G_1, G_2, T_1, T_2)$ and the master secret key $MSK = (g^{d_1}, g^{d_2}, g^{d_3}, a_1, a_2, b_1, b_2)$;

1: Compute $(p, g, \mathbb{G}, h, \mathbb{H}, \mathbb{G}_T, e) \leftarrow \mathcal{G}(k)$;
2: Define $\mathcal{H} : \{0,1\}^* \rightarrow \mathbb{G}$ and $\mathcal{F} : \mathbb{G}_T \rightarrow \mathbb{H}^3$ as hash functions;
3: Choose $d_1, d_2, d_3 \in_R \mathbb{Z}_p$ and $a_1, a_2, b_1, b_2 \in_R \mathbb{Z}_p^*$;
4: Compute $H_1 = h^{a_1}, H_2 = h^{a_2}, G_1 = g^{a_1}, G_2 = g^{a_2}, T_1 = e(g,h)^{d_1 a_1 + d_3}, T_2 = e(g,h)^{d_2 a_2 + d_3}$;
5: **return** $(Param = (e, g, h, H_1, H_2, G_1, G_2, T_1, T_2), MSK = (g^{d_1}, g^{d_2}, g^{d_3}, a_1, a_2, b_1, b_2))$;

- *Secret Key Generation*: As mentioned in Section II-A, there are two types of users in the system, which belong to IBC and ABC respectively. For the users in IBC, PKG takes their identity $id \in \{0,1\}^*$ to execute the $\mathsf{KeyGen}_{\mathsf{IBE}}$ algorithm defined in Algorithm 2, and the corresponding secret key ISK_{id} is returned. For the users in ABC, PKG takes the attribute set S to execute the $\mathsf{KeyGen}_{\mathsf{ABE}}$ algorithm defined in Algorithm 3, and the corresponding secret key SK_S is returned.
- *Re-Encryption Key Generation*: As shown in Algorithm 4, the ReKeyGen algorithm enables the user in IBC to generate the re-encryption key with his own secret key and a designated policy structure. For a policy structure $(\mathbf{M}, \pi)$, let m and n denote the rows and columns of $\mathbf{M}$, and use $(\mathbf{M})_{i,j}$ to represent the (i,j)th element of $\mathbf{M}$.
- *Encryption*: In FABRIC, the plaintext is encrypted with the user's public key, which is also his identity. The detail of Encrypt algorithm is shown in Algorithm 5.
- *Re-encryption*: By running the ReEnc algorithm with the re-encryption key, the proxy could transform the ciphertext in IBC into the re-encrypted ciphertext in ABC, and the detail is shown in Algorithm 6.

Algorithm 2: $\mathsf{KeyGen}_{\mathsf{IBE}}(Param, MSK, id)$.

Input: the public parameters $Param$, the master secret key MSK, and the identity $id \in \{0,1\}^*$ of a user;
Output: user id's secret key $ISK_{id} = (isk_0, isk_1)$;

1: Choose $r_1, r_2, \sigma_{id} \in_R \mathbb{Z}_p$;
2: Compute $isk_0 = (isk_{0,1}, isk_{0,2}, isk_{0,3}) = (h^{b_1 r_1}, h^{b_2 r_2}, h^{r_1 + r_2})$;
3: **for** $t = 1, 2$ **do**
4: Compute $isk_{1,t} = \mathcal{H}(t,1,id,0)^{\frac{b_1 r_1}{a_t}} \cdot \mathcal{H}(t,2,id,0)^{\frac{b_2 r_2}{a_t}} \cdot \mathcal{H}(t,3,id,0)^{\frac{r_1 + r_2}{a_t}} \cdot g^{d_t} \cdot g^{\frac{\sigma_{id}}{a_t}}$;
5: **end for**
6: Compute $isk_{1,3} = g^{d_3} \cdot g^{-\sigma_{id}}$;
7: Set $isk_1 = (isk_{1,1}, isk_{1,2}, isk_{1,3})$;
8: **return** $ISK_{id} = (isk_0, isk_1)$;

Algorithm 3: $\mathsf{KeyGen}_{\mathsf{ABE}}(Param, MSK, S)$.

Input: the public parameters $Param$, the master secret key MSK, and the attribute set S;
Output: the secret key $SK_S = (sk_0, \{sk_y\}_{y \in S}, sk')$;

1: Choose $r_1', r_2', \sigma' \in_R \mathbb{Z}_p$;
2: Compute $sk_0 = (sk_{0,1}, sk_{0,2}, sk_{0,3}) = (h^{b_1 r_1'}, h^{b_2 r_2'}, h^{r_1' + r_2'})$;
3: **for** each y in S **do**
4: Choose $\sigma_y \in_R \mathbb{Z}_p$;
5: **for** $t = 1, 2$ **do**
6: Compute $sk_{y,t} = \mathcal{H}(t,1,y)^{\frac{b_1 r_1'}{a_t}} \cdot \mathcal{H}(t,2,y)^{\frac{b_2 r_2'}{a_t}} \cdot \mathcal{H}(t,3,y)^{\frac{r_1' + r_2'}{a_t}} \cdot g^{\frac{\sigma_y}{a_t}}$;
7: **end for**
8: Set $sk_y = (sk_{y,1}, sk_{y,2}, g^{-\sigma_y})$;
9: **end for**
10: **for** $t = 1, 2$ **do**
11: Compute $sk_t' = \mathcal{H}(t,1,1,0)^{\frac{b_1 r_1'}{a_t}} \cdot \mathcal{H}(t,2,1,0)^{\frac{b_2 r_2'}{a_t}} \cdot \mathcal{H}(t,3,1,0)^{\frac{r_1' + r_2'}{a_t}} \cdot g^{d_t} \cdot g^{\frac{\sigma'}{a_t}}$;
12: **end for**
13: Set $sk' = (sk_1', sk_2', g^{d_3} \cdot g^{-\sigma'})$;
14: **return** $SK_S = (sk_0, \{sk_y\}_{y \in S}, sk')$;

- *Decryption*: It is worth noting that there are ciphertext and re-encrypted ciphertext which belong to different cryptosystems. For users in IBC, $\mathsf{Decrypt}_{\mathsf{IBE}}$ algorithm enables them to decrypt the ciphertext with their secret key ISK_{id}, which is demonstrated in Algorithm 7. While users in ABC need to first determine whether their attribute sets match the access policy in the re-encrypted ciphertext. When the attribute set in SK_S matches $MSP(\mathbf{M}, \pi)$, the constants $\{\gamma_i\}_{i \in I}$ which defined in Section III-A could be computed. In this way, users could run the $\mathsf{Decrypt}_{\mathsf{ABE}}$ algorithm to decrypt the re-encrypted ciphertext, which is given in Algorithm 8.

Algorithm 4: ReKeyGen $(Param, ISK_{id}, (\mathbf{M}, \pi))$.

Input: the public parameters $Param$, user id's secret key $ISK_{id} = (isk_0, isk_1)$ and a policy structure $(\mathbf{M}, \pi)$;
Output: the re-encryption key $RK_{id\to S} = (rk_0, rk_1, rk_2, rk_3, rk_4)$;

Choose $s'_1, s'_2, k_1, k_2, k_3 \in_R \mathbb{Z}_p$;
Compute $rk_0 = (rk_{0,1}, rk_{0,2}, rk_{0,3}) = (H_1^{s'_1}, H_2^{s'_2}, h^{s'_1+s'_2})$;
for For $i = 1, \ldots, m$ and $\ell = 1, 2, 3$ **do**
  Compute $rk_{1,i,\ell} = \mathcal{H}(1, \ell, \pi(i))^{s'_1} \cdot \mathcal{H}(2, \ell, \pi(i))^{s'_2} \cdot \prod_{j=1}^{n} [\mathcal{H}(1, \ell, j, 0)^{s'_1} \cdot \mathcal{H}(2, \ell, j, 0)^{s'_2}]^{(\mathbf{M})_{i,j}}$;
end for
Set $rk_1 = \{rk_{1,i,\ell}\}_{i\in[0,m]\wedge\ell\in[1,3]}$;
Compute $\boldsymbol{\eta} = (\eta_1, \eta_2, \eta_3) = \mathcal{F}(T_1^{s'_1} \cdot T_2^{s'_2})$, which is the hash result of $\mathcal{F}$;
Compute $rk_2 = (rk_{2,1}, rk_{2,2}, rk_{2,3}) = (\eta_1 \cdot h^{k_1}, \eta_2 \cdot h^{k_2}, \eta_3 \cdot h^{k_3})$;
Compute $rk_3 = (rk_{3,1}, rk_{3,2}, rk_{3,3}) = (isk_{1,1} \cdot g^{k_1}, isk_{1,2} \cdot g^{k_2}, isk_{1,3} \cdot g^{k_3})$;
Set $rk_4 = isk_0$;
return $RK_{id\to S} = (rk_0, rk_1, rk_2, rk_3, rk_4)$;

Algorithm 5: Encrypt(Param, msg, id).

Input: the public parameters $Param$, a message $msg \in \mathbb{G}_T$, and the user's identity $id \in \{0, 1\}^*$;
Output: the ciphertext $C_{id} = (c_0, c_1, c_2, c_3)$;

Choose $s_1, s_2 \in_R \mathbb{Z}_p$;
Compute $c_0 = msg \cdot T_1^{s_1} \cdot T_2^{s_2}$;
Compute $c_1 = (c_{1,1}, c_{1,2}, c_{1,3}) = (H_1^{s_1}, H_2^{s_2}, h^{s_1+s_2})$;
Compute $c_2 = (c_{2,1}, c_{2,2}, c_{2,3}) = (G_1^{s_1}, G_2^{s_2}, g^{s_1+s_2})$;
for $\ell = 1, 2, 3$ **do**
  Compute $c_{3,\ell} = \mathcal{H}(1, \ell, id, 0)^{s_1} \cdot \mathcal{H}(2, \ell, id, 0)^{s_2}$;
end for
Set $c_3 = (c_{3,1}, c_{3,2}, c_{3,3})$;
return$C_{id} = (c_0, c_1, c_2, c_3)$;

Algorithm 6: ReEnc $(Param, C_{id}, RK_{id\to S})$.

Input: the public parameters $Param$, the ciphertext $C_{id} = (c_0, c_1, c_2, c_3)$, and the re-encryption key $RK_{id\to S} = (rk_0, rk_1, rk_2, rk_3, rk_4)$;
Output: the re-encrypted ciphertext $C_{id\to S} = (c'_0, c'_1, c'_2, c'_3, c'_4)$;

Compute $RC = \prod_{i=1}^{3} e(c_{1,i}, rk_{3,i}) / \prod_{i=1}^{3} e(c_{3,i}, rk_{4,i})$;
Compute $c'_0 = c_0 / RC$;
Set $c'_1 = c_2$, $c'_2 = rk_0$, $c'_3 = rk_1$, $c'_4 = rk_2$
return $C_{id\to S} = (c'_0, c'_1, c'_2, c'_3, c'_4)$;

Algorithm 7: $\mathsf{Decrypt}_{\mathsf{IBE}}(\mathsf{Param}, \mathsf{C_{id}}, \mathsf{ISK_{id}})$.

Input: the public parameters $Param$, the ciphertext $C_{id} = (c_0, c_1, c_2, c_3)$, and the user id's secret key $ISK_{id} = (isk_0, isk_1)$;
Output: the message msg;

Compute $msg = c_0 \cdot \prod_{i=1}^{3} e(c_{3,i}, isk_{0,i}) / \prod_{i=1}^{3} e(c_{1,i}, isk_{1,i})$;
return msg;

Algorithm 8: $\mathsf{Decrypt}_{\mathsf{ABE}}(Param, C_{id\to S}, SK_S)$.

Input: the public parameters $Param$, the re-encrypted ciphertext $C_{id\to S} = (c'_0, c'_1, c'_2, c'_3, c'_4)$, and the user's secret key $SK_S = (sk_0, \{sk_y\}_{y\in S}, sk')$;
Output: the message msg;

Compute $D_1 = e(\prod_{i\in I}(c'_{3,i,1})^{\gamma_i}, sk_{0,1}) \cdot e(\prod_{i\in I}(c'_{3,i,2})^{\gamma_i}, sk_{0,2}) \cdot e(\prod_{i\in I}(c'_{3,i,3})^{\gamma_i}, sk_{0,3})$;
Compute $D_2 = e(sk'_1 \cdot \prod_{i\in I}(sk_{\pi(i),1})^{\gamma_i}, c'_{2,0}) \cdot e(sk'_2 \cdot \prod_{i\in I}(sk_{\pi(i),2})^{\gamma_i}, c'_{2,1}) \cdot e(sk'_3 \cdot \prod_{i\in I}(sk_{\pi(i),3})^{\gamma_i}, c'_{2,2})$;
Compute $\boldsymbol{\eta} = (\eta_1, \eta_2, \eta_3) = \mathcal{F}(D_2/D_1)$;
Compute $msg = c'_0 \cdot \prod_{i=1}^{3} e(c'_{1,i}, c'_{4,i}/\eta_i)$;
return the message msg;

V. Security Analysis

It is worth noting that a series of games, which demonstrate the interaction between $\mathcal{C}$ and $\mathcal{A}$, constitute the security proof of FABRIC. To achieve the semantic security, the adversary $\mathcal{A}$ should distinguish the random string between the challenge ciphertext generated by the challenger $\mathcal{C}$ in the security proof. In the simulation of the capability for the adversary $\mathcal{A}$, the challenger $\mathcal{C}$ needs to provide the ciphertext and private key according to $\mathcal{A}$'s request under certain restrictions. In a dual encryption system, private keys and ciphertexts can take one of two forms: normal and semi-functional. A normal ciphertext can be decrypted by both normal and semi-functional keys, while a semi-functional ciphertext can be only decrypted by the normal keys. Specifically, the semi-functional keys and ciphertexts are only used in the proof of security instead of the real system. The methodology involved in dual system encryption [31] would replace the keys and ciphertexts with semi-functional keys and semi-functional ciphertexts over a sequence of security games. The first game is the real security game, featured with normal keys and ciphertext. In the second game, the ciphertext is replaced with the semi-functional one and keys remain normal. In subsequent games, the key requested by the attacker is changed one by one to be semi-functional. In the last game, the ciphertext will be replaced with a random message. By proving the indistinguishability between these games step by step, the challenge ciphertext is proved to be indistinguishable from a random message which at last achieves semantic security. However, since FABRIC is a PRE scheme, the indistinguishability of ciphertext and re-encrypted ciphertext needs to be proved respectively. Therefore, in the Appendix, which can be found on the Computer Society Digital Library at,[1] we construct the security game $Game_2$ - $Game_5$ and $Game_6$ - $Game_{10}$ to prove the security of ciphertext and re-encrypted ciphertext, respectively. By proving

[1] http://doi.ieeecomputersociety.org/10.1109/TDSC.2023.3240820

the indistinguishability of these security games, the security of FABRIC is proved. Finally, since the adversary cannot obtain any information about the plaintext from the ciphertext and re-encrypted ciphertext, FABRIC could be against the adversary and its attacks described in Section II-D, thus satisfying the proposed security goals.

A. Security Game With Random Oracles

We first define the algorithm $\mathsf{Init}(p)$. When receiving the prime p, this algorithm chooses $j_1, j_2 \in_R \mathbb{Z}_p^*$ and returns

$$\mathbf{J} = \begin{bmatrix} j_1 & 0 \\ 0 & j_2 \\ 1 & 1 \end{bmatrix} \tilde{\mathbf{j}}^{\perp} = \begin{bmatrix} j_1^{-1} \\ j_2^{-1} \\ -1 \end{bmatrix}.$$

Then, the security game between $\mathcal{A}$ and $\mathcal{C}$ with random oracles could be demonstrated as below:

$\mathsf{Setup}\ (k)$: When receiving the security parameter k, $\mathcal{C}$ runs $\mathcal{G}(k)$ to get $G = (p, g, \mathbb{G}, h, \mathbb{H}, \mathbb{G}_T, e)$. $\mathcal{C}$ samples $d_1, d_2, d_3 \in_R \mathbb{Z}_p$ and uses $\mathbf{d}$ to represent the column vector $(d_1, d_2, d_3)^\top$. Run $\mathsf{Init}(p)$ to obtain $(\mathbf{X}, \mathbf{x}^\perp)$ and $(\mathbf{Y}, \mathbf{y}^\perp)$. Then, $\mathcal{C}$ sends $Param = (G, [\mathbf{X}]_1, [\mathbf{X}]_2, [\mathbf{d}^\top \mathbf{X}]_T)$ to $\mathcal{A}$ and stores $MSK = (\mathbf{X}, \mathbf{Y}, [\mathbf{d}]_1)$.

Furthermore, $\mathcal{C}$ maintains three lists: $\mathcal{L}_1, \mathcal{L}_2$, and $\mathcal{L}_3$. $\mathcal{L}_1$ keeps the items of $(id, \mathbf{R}_{id})$, $(y, \mathbf{W}_y)$ and $(j, \mathbf{U}_j)$, where id represents identity, y represents attribute, and j is integer. And $\mathbf{R}_{id}$, $\mathbf{W}_y$ and $\mathbf{U}_j$ denote 3×3 matrices of $\mathbb{Z}_p$. $\mathcal{L}_2$ keeps the items of (q, r), where $r \in \mathbb{G}$ and q represents $(t, l, id, 0)$, (t, l, y), $(t, l, j, 0)$ (for $l \in \{1, 2, 3\}$ and $t \in \{1, 2\}$). The list $\mathcal{L}_3$ keeps the items of (id, ISK_{id}), (S, SK_S), and $((id, (\mathbf{M}, \pi)), RK_{id \to S})$.

Hash Queries: $\mathcal{C}$ simulates the random oracle as below:

- $(t, l, id, 0)$: $\mathcal{C}$ first searches $\mathcal{L}_2$ for $((t, l, id, 0), r)$. If item exists, $\mathcal{C}$ responds $\mathcal{A}$ with r. Otherwise, $\mathcal{C}$ searches $\mathcal{L}_1$ for $(id, \mathbf{R}_{id})$. If item exists, $\mathcal{C}$ computes $r = [(\mathbf{R}_{id}^\top \mathbf{X})_{l,t}]_1$. Then add $((t, l, id, 0), r)$ to $\mathcal{L}_2$ and return it to $\mathcal{A}$. Otherwise, $\mathcal{C}$ chooses $\mathbf{R}_{id} \in \mathbb{Z}_p^{3\times 3}$ and adds $(id, \mathbf{R}_{id})$ to $\mathcal{L}_1$. Similarly, r is computed as $[(\mathbf{R}_{id}^\top \mathbf{X})_{l,t}]_1$ and sent to $\mathcal{A}$. And $((t, l, id, 0), r)$ is added to $\mathcal{L}_2$.
- (t, l, y): $\mathcal{C}$ first searches $\mathcal{L}_2$ for $((t, l, y), r)$. If item exists, $\mathcal{C}$ responds $\mathcal{A}$ with r. Otherwise, $\mathcal{C}$ searches $\mathcal{L}_1$ for $(y, \mathbf{W}_y)$. If item exists, $\mathcal{C}$ computes $r = [(\mathbf{W}_y^\top \mathbf{X})_{l,t}]_1$. Then add $((t, l, y), r)$ to $\mathcal{L}_2$ and return it to $\mathcal{A}$. Otherwise, $\mathcal{C}$ chooses $\mathbf{W}_y \in \mathbb{Z}_p^{3\times 3}$ and add $(y, \mathbf{W}_y)$ to $\mathcal{L}_1$. Similarly, r is computed as $[(\mathbf{W}_y^\top \mathbf{X})_{l,t}]_1$ and sent to $\mathcal{A}$. And $((t, l, y), r)$ is added to $\mathcal{L}_2$,
- $(t, l, j, 0)$: $\mathcal{C}$ first searches $\mathcal{L}_2$ for $((t, l, j, 0), r)$. If item exists, $\mathcal{C}$ responds $\mathcal{A}$ with r. Otherwise, $\mathcal{C}$ searches $\mathcal{L}_1$ for $(j, \mathbf{U}_j)$. If item exists, $\mathcal{C}$ computes $r = [(\mathbf{U}_j^\top \mathbf{X})_{l,t}]_1$. Then add $((t, l, j, 0), r)$ to $\mathcal{L}_2$ and return it to $\mathcal{A}$. Otherwise, $\mathcal{C}$ chooses $\mathbf{U}_j \in \mathbb{Z}_p^{3\times 3}$ and add $(j, \mathbf{U}_j)$ to $\mathcal{L}_1$.
  Similarly, r is computed as $[(\mathbf{U}_j^\top \mathbf{X})_{l,t}]_1$ and sent to $\mathcal{A}$. And $((t, l, j, 0), r)$ is added to $\mathcal{L}_2$.
- $(T_1^{s_1} \cdot T_2^{s_2})$: $\mathcal{C}$ first searches $\mathcal{L}_2$ for $((T_1^{s_1} \cdot T_2^{s_2}), \boldsymbol{\eta})$. If item exists, $\mathcal{C}$ responds $\mathcal{A}$ with $\boldsymbol{\eta}$. Otherwise, $\mathcal{C}$ chooses $\boldsymbol{\eta} \in_R \mathbb{H}^3$ and adds $((T_1^{s_1} \cdot T_2^{s_2}), \boldsymbol{\eta})$ to $\mathcal{L}_2$. Finally, $\boldsymbol{\eta}$ is sent to $\mathcal{A}$.
- Anything else : $\mathcal{C}$ first searches $\mathcal{L}_2$ for (q, r). If item exists, $\mathcal{C}$ responds $\mathcal{A}$ with r. Otherwise, $\mathcal{C}$ chooses $r' \in_R \mathbb{G}$. Then, add (q, r') to $\mathcal{L}_2$ and return r' to $\mathcal{A}$.

Phase 1: $\mathcal{C}$ could response $\mathcal{A}$ as follows:

- $\mathsf{KeyGen_{IBE}}$: $\mathcal{A}$ queries with the identity id, $\mathcal{C}$ first searches $\mathcal{L}_3$ for (id, ISK_{id}). If item exists, $\mathcal{C}$ responds $\mathcal{A}$ with ISK_{id}. Otherwise, $\mathcal{C}$ retrieves $\mathbf{R}_{id}$ from list $\mathcal{L}_1$. Then $\mathcal{C}$ randomly samples $\mathbf{r}_{id} \in \mathbb{Z}_p^2$ and $\sigma_{id} \in \mathbb{Z}_p$, and computes

$$isk_0 = [\mathbf{Y}\mathbf{r}_{id}]_2$$

$$isk_1 = [\mathbf{d} + \mathbf{R}_{id}\mathbf{Y}\mathbf{r}_{id} + \sigma_{id}\mathbf{x}^\perp]_1.$$

  Let $ISK_{id} = (isk_0, isk_1)$ be the secret key of user id. Finally, $\mathcal{C}$ inserts (id, ISK_{id}) into the list $\mathcal{L}_3$ and returns it to $\mathcal{A}$.
- $\mathsf{KeyGen_{ABE}}$: $\mathcal{A}$ queries with an attribute set S, $\mathcal{C}$ first searches $\mathcal{L}_3$ for (S, SK_S). If item exists, $\mathcal{C}$ responds $\mathcal{A}$ with SK_S. Otherwise, $\mathcal{C}$ retrieves $\mathbf{U}_1$ and $\{\mathbf{W}_y\}_{y\in S}$ from $\mathcal{L}_3$. $\mathcal{C}$ samples $\mathbf{r}_s \in_R \mathbb{Z}_p^2$, $\sigma' \in \mathbb{Z}_p$, and $\{\sigma_y\}_{y\in S} \in \mathbb{Z}_p$. Then computes

$$sk_0 = [\mathbf{Y}\mathbf{r}_s]_2, \quad sk_y = [\mathbf{W}_y\mathbf{Y}\mathbf{r}_s + \sigma_y\mathbf{x}^\perp]_1$$

$$sk' = [\mathbf{d} + \mathbf{U}_1\mathbf{Y}\mathbf{r}_s + \sigma'\mathbf{x}^\perp]_1,$$

  for all $y \in S$. Let $SK_S = (sk_0, \{sk_y\}_{y\in S}, sk')$ be the secret key associated with the attribute set S. Finally, $\mathcal{C}$ adds (S, SK_S) to $\mathcal{L}_3$ and returns it to $\mathcal{A}$.
- $\mathsf{ReKeyGen}$: $\mathcal{A}$ queries with a policy structure $(\mathbf{M}, \pi)$ and an identity id, $\mathcal{C}$ first searches $\mathcal{L}_3$ for $((id, (\mathbf{M}, \pi)), RK_{id\to S})$. If item exists, $\mathcal{C}$ responds $\mathcal{A}$ with $RK_{id\to S}$. Otherwise, suppose $\mathcal{A}$ has queried the secret key of user id. If not, $\mathcal{C}$ could perform query on behalf of $\mathcal{A}$. After that, $\mathcal{C}$ run the $\mathsf{ReKeyGen}()$ algorithm to generate $RK_{id\to S}$. Finally, $\mathcal{C}$ inserts $((id, (\mathbf{M}, \pi)), RK_{id\to S})$ into the list $\mathcal{L}_3$ and returns $RK_{id\to S}$ to $\mathcal{A}$.

Challenge: $\mathcal{A}$ first chooses an identity id^* with the restrains below:

- Case 1. The secret key and re-encryption key of identity id^* are not queried by the adversary $\mathcal{A}$ during *Phase 1*.
- Case 2. The secret key of identity id^* is queried during *Phase 1*. At the same time, for the re-encryption keys of identity id^* and policy structures $(\mathbf{M}^*, \pi^*)$ that have been queried in *Phase 1*, there is no attribute set S which has been queried in *Phase 1*, that satisfies the policy structure $(\mathbf{M}^*, \pi^*)$.

For Case 1, the adversary $\mathcal{A}$ chooses two message $msg_0, msg_1 \in \mathbb{G}_T$, and sends messages with the identity id^* to challenger $\mathcal{C}$. After that, $\mathcal{C}$ retrieves $\mathbf{R}_{id}$ from list $\mathcal{L}_1$ and samples $v \in \{0, 1\}$, $s_1, s_2 \in \mathbb{Z}_p$. Let $\mathbf{s} = (s_1, s_2)$, and compute

$$c_0^* = msg_v \cdot [\mathbf{d}^\top \mathbf{X}\mathbf{s}]_T$$

$$c_1^* = [\mathbf{X}\mathbf{s}]_2, \quad c_2^* = [\mathbf{X}\mathbf{s}]_1, \quad c_3^* = [\mathbf{R}_{id}^\top \mathbf{X}\mathbf{s}]_1.$$

Finally, $\mathcal{C}$ responds $\mathcal{A}$ with the ciphertext $C_{id^*}^* = (c_0^*, c_1^*, c_2^*, c_3^*)$.

For Case 2, $\mathcal{A}$ chooses two message $msg_0, msg_1 \in \mathbb{G}_T$, and sends messages with the identity id^* and the policy structure $(\mathbf{M}^*, \pi^*)$ to $\mathcal{C}$. After that, for all $i = 1, \ldots, m, j = 1, \ldots, n, l, t,$

$\mathcal{C}$ obtains $[(\mathbf{M}_{\pi(i)}^{\top}\mathbf{X})_{l,t}]_1$ and $[(\mathbf{U}_j^{\top}\mathbf{X})_{l,t}]_1$ from $\mathcal{L}_1$, and chooses $v \in \{0,1\}$, $s_1, s_2, s_1', s_2', k_1, k_2 \in \mathbb{Z}_p$. Let $\mathbf{s} = (s_1, s_2)$, $\mathbf{s}' = (s_1', s_2')$, and $\mathbf{k} = (k_1, k_2, k_3)$. After that, $\mathcal{C}$ computes $(T_1^{s_1'} \cdot T_2^{s_2'}) = [\mathbf{d}^{\top}\mathbf{X}\mathbf{s}']_T$ and perform hash query on behalf of $\mathcal{A}$. Then, retrieve $\boldsymbol{\eta}$ from list $\mathcal{L}_2$ and compute

$$c_0'^{*} = msg_v / [\mathbf{k}^{\top}\mathbf{X}\mathbf{s}]_T$$
$$c_1'^{*} = [\mathbf{X}\mathbf{s}]_1, \quad c_2'^{*} = [\mathbf{X}\mathbf{s}']_2,$$
$$c_{3,i}'^{*} = [\sum_{j=1}^{n}(\mathbf{M})_{i,j}\mathbf{U}_j^{\top}\mathbf{X}\mathbf{s}' + \mathbf{W}_{\pi(i)}^{\top}\mathbf{X}\mathbf{s}']_1$$
$$c_4'^{*} = \boldsymbol{\eta}[\mathbf{k}]_2.$$

Let $c_3'^{*} = \{c_{3,i}'^{*}\}_{i\in[0,m]}$. Finally, the challenger $\mathcal{C}$ returns $C_{id^{*}\to S^{*}}^{*} = (c_0'^{*}, c_1'^{*}, c_2'^{*}, c_3'^{*}, c_4'^{*})$ to adversary $\mathcal{A}$.

Phase 2: Except for the limitations of the *Challenge* phase, $\mathcal{C}$ repeats as the same as *Phase 1*.

Guess: Eventually, $\mathcal{A}$ makes the guess v' of v. If $v' = v$, $\mathcal{A}$ wins this game.

B. Security Strength

Theorem 1 (Data confidentiality). Data confidentiality in FABRIC means that data contents can achieve confidential to any curious cloud servers and malicious attackers, only the authorized user who holds the corresponding private key could decrypt it.

Proof. The security of our protocol is defined by a series of security games between the challenger $\mathcal{C}$ and the adversary $\mathcal{A}$. As for the data confidentiality, the goal of the security proof is to establish the relationship between the semantic security of the proposed FABRIC and the underlying mathematic assumption. To capture the attacks on the original ciphertext and the re-encryption ciphertext, the capability of the adversary is simulated by the oracles in the security proof. In this way, the confidentiality of the FABRIC could be formally proved under the given mathematic assumption. Readers can refer to [50] and [51] for more details. This theorem is directly derived from Theorem 2, and the proof of Theorem 2 is given in the Appendix, available in the online supplemental material.

Theorem 2. The widely accepted DLIN assumption will fail with non-negligible advantage in case that the security of our scheme is broken by an adversary $\mathcal{A}$.

Theorem 3 (Controllable authorization). FABRIC can offer the controllable authorization property, which guarantees that only data users with the attribute set matches the access policy can be authorized and able to decrypt the re-encrypted ciphertext. Any data users collude with each other or the proxy server colludes with the delegatee is unable to access private information.

Proof. The property of controllable authorization is implicitly considered in the security model of data confidentiality. Suppose that there exists a curious proxy server collude with the delegate to obtain private information from the original ciphertexts, re-encrypted ciphertexts and re-encrypted keys. As demonstrated in the Lemma 1, the KeyGen oracle and ReKeyGen oracle have already been considered to capture the attacks mounted by the collusion between the proxy server and the unauthorized users. That is to say, this malicious collusion is still unable to access private information. Thus, FABRIC is featured with the controllable authorization property.

C. Security Proof

Theorem 4. If the above lemmas hold, the FABRIC scheme achieves adaptively secure under DLIN assumption.

Proof. It is noteworthy that the definition of DLIN assumption and the indistinguishability of games are both provided concretely in Appendix, available in the online supplemental material. The $Game_0$–$Game_5$ and $Game_0$–$Game_{10}$ are demonstrated to be indistinguishable respectively, which means that the advantages of adversary $\mathcal{A}$ to distinguish whether ciphertext and re-encrypted ciphertext are random messages are the same as that of $Game_5$ and $Game_{10}$. Obviously, the advantage of $\mathcal{A}$ in $Game_5$ or $Game_{10}$ is negligible, so the advantage of $\mathcal{A}$ in $Game_0$ is negligible as well.

VI. Experiment and Evaluation

We conduct the comparisons between existing works with the proposed scheme with respect to properties and efficiency, thereby demonstrating the practicality and feasibility of FABRIC.

A. Properties Comparison

Table II summarizes and compares the properties of FABRIC and other related works [24], [26], [45], [48], where the "✓" indicates "satisfy" and the × indicates "not satisfy". It is noteworthy that IBET [24] enables the ciphertext transformation from the IBE scheme to the identity-based broadcast encryption (IBBE) scheme. Specifically, IBBE allows a set of identities to be used to encrypt data during the encryption phase, and the ciphertext can be decrypted by the corresponding private key of any identity in the set of identities. Obviously, compared to ABE, IBBE cannot achieve flexible access control. Furthermore, CAB [45] and LUC [48] achieve the conversion from the ciphertext of ABE to the ciphertext of IBE. While CPAS [26] is an AB-PRE scheme constructed based on LSSS. Since the computation cost of encryption and decryption in [26], [48] is linear with the quantity of attributes, users suffer from heavy computation overhead. Compared with ABE schemes [26], [45], only FABRIC and LUC [48] are unbounded. It can be observed that, schemes [24], [45], [48] merely achieve selective security which is too impractical to be applied in real-world scenarios, and only FABRIC and CPAS [26] achieve adaptively security. Furthermore, CAB [45] is constructed based on the AND gate, which limits the expressiveness of the access policy. LUC [48] and CPAS [26] utilize LSSS to formulate access policy, while FABRIC is based on MSP. Compared with LSSS, MSP gets more flexible access control. Furthermore, FABRIC is proved to be adaptive security, which shows it has better resistance to various attack scenarios. More importantly, different from other schemes [24], [26], [45], [48], only FABRIC is constructed

TABLE II
THE COMPARISON OF PROPERTIES

Scheme	Type	ABE Policy	Unbounded	Security	Asymmetric Pairing	Access Structure
IBET [24]	IBE → IBBE	–	–	Selectively	×	–
CAB [45]	ABE → IBE	CP-ABE	×	Selectively	×	AND Gate
LUC [48]	ABE → IBE	CP-ABE	✓	Selectively	×	LSSS
CPAS [26]	ABE → ABE	CP-ABE	×	Adaptively	×	LSSS
FABRIC	IBE → ABE	CP-ABE	✓	Adaptively	✓	MSP

TABLE III
THE COMPARISON OF COMPUTATION COSTS

Scheme	Setup	Encryption	Re-Encryption	Decryption-I	Decryption-II
IBET [24]	$(m+2)T'_G + T'_p$	$2T'_G + T'_{GT}$	T'_p	T'_p	$T'_{GT} + 3T'_p$
CAB [45]	$(3n+1)T'_G + T'_{GT}$	$(n+1)T'_G + T'_{GT}$	$2T'_G + T'_{GT} + (i+1)T'_p$	$(i+1)T'_p$	$2T'_p$
LUC [48]	$T'_{GT} + T'_p$	$(5n+2)T'_G + T'_{GT}$	$iT'_{GT} + (3i+1)T'_p$	$iT'_{GT} + (3i+1)T'_p$	$3T'_p$
CPAS [26]	$(n+4)T'_G + T'_{GT} + T'_p$	$(3n+5)T'_G + T'_{GT} + T'_p$	$(6i+9)T'_G + 2T'_{GT} + 2T'_p$	$(i+1)T'_{GT} + (2i+2)T'_p$	$(3i+3)T'_{GT} + (6i+6)T'_p$
FABRIC	$2T_G + 2T_H + 2T_{GT} + 2T_p$	$9T_G + 3T_H + 2T_{GT}$	$6T_p$	$6T_p$	$9T_p$

TABLE IV
THE COMPARISON OF COMMUNICATION COSTS

Scheme	Public Parameters	Ciphertext	Re-Encrypted Ciphertext	Re-encryption Key	Secret Key in IBC	Secret Key in ABC
IBET [24]	$(m+4)\lvert\mathbb{G}'\rvert + \lvert\mathbb{G}'_T\rvert$	$2\lvert\mathbb{G}'\rvert + \lvert\mathbb{G}'_T\rvert$	$4\lvert\mathbb{G}'\rvert + \lvert\mathbb{G}'_T\rvert$	$4\lvert\mathbb{G}'\rvert$	$\lvert\mathbb{G}'\rvert$	–
CAB [45]	$(3n+3)\lvert\mathbb{G}'\rvert + \lvert\mathbb{G}'_T\rvert$	$(n+1)\lvert\mathbb{G}'\rvert + \lvert\mathbb{G}'_T\rvert$	$\lvert\mathbb{G}'\rvert + 2\lvert\mathbb{G}'_T\rvert$	$(2n+3)\lvert\mathbb{G}'\rvert + \lvert\mathbb{G}'_T\rvert$	$2\lvert\mathbb{G}'\rvert$	$(2n+1)\lvert\mathbb{G}'\rvert$
LUC [48]	$6\lvert\mathbb{G}'\rvert + \lvert\mathbb{G}'_T\rvert$	$(3n+2)\lvert\mathbb{G}'\rvert + \lvert\mathbb{G}'_T\rvert$	$4\lvert\mathbb{G}'\rvert + \lvert\mathbb{G}'_T\rvert$	$(2n+5)\lvert\mathbb{G}'\rvert$	$2\lvert\mathbb{G}'\rvert$	$(2n+1)\lvert\mathbb{G}'\rvert$
CPAS [26]	$(n+6)\lvert\mathbb{G}'\rvert + \lvert\mathbb{G}'_T\rvert$	$(2n+4)\lvert\mathbb{G}'\rvert + \lvert\mathbb{G}'_T\rvert$	$(4n+7)\lvert\mathbb{G}'\rvert + 2\lvert\mathbb{G}'_T\rvert$	$(3n+7)\lvert\mathbb{G}'\rvert + \lvert\mathbb{G}'_T\rvert$	–	$(n+2)\lvert\mathbb{G}'\rvert$
FABRIC	$3\lvert\mathbb{G}\rvert + 3\lvert\mathbb{H}\rvert + 2\lvert\mathbb{G}_T\rvert$	$6\lvert\mathbb{G}\rvert + 3\lvert\mathbb{H}\rvert + \lvert\mathbb{G}_T\rvert$	$3(n+1)\lvert\mathbb{G}\rvert + 6\lvert\mathbb{H}\rvert + \lvert\mathbb{G}_T\rvert$	$3(n+1)\lvert\mathbb{G}\rvert + 9\lvert\mathbb{H}\rvert$	$3\lvert\mathbb{G}\rvert + 3\lvert\mathbb{H}\rvert$	$3(n+1)\lvert\mathbb{G}\rvert + 3\lvert\mathbb{H}\rvert$

based on the asymmetric pairing group. From this aspect of view, FABRIC obviously has higher availability in real-life scenarios.

B. *Efficiency Analysis*

Tables III and IV summarize the theoretical computation and communication costs of different schemes, respectively. However, it is noteworthy that related schemes [24], [26], [45], [48] are constructed based on the symmetric pairing group, while FABRIC is constructed based on the asymmetric pairing group. Although the schemes constructed separately based on the symmetric group and the asymmetric group cannot be directly compared, theoretical analysis can still show the efficiency of these schemes. As shown in Tables III and IV, only FABRIC and IBET [24], which encrypt the message with the identity, not only enjoy the constant computation overhead during the encryption and decryption-I phase, but also achieve constant ciphertext length. In contrast, the computation overhead and ciphertext length of other schemes [26], [45], [48] grow linearly as the quantity of attributes increases. Moreover, different from ABE-related schemes [26], [45], [48], in the re-encryption phase, only FABRIC achieves constant computation cost. Because the re-encrypted ciphertext of schemes [24], [45], [48] is the ciphertext of IBE, these schemes [24], [45], [48] can enjoy constant computation cost in the decryption-II phase and re-encrypted ciphertext length. While the computation overhead and ciphertext length of CPAS [26], whose re-encrypted ciphertext is the ABE ciphertext, are both positively correlated with the number of attributes. Fortunately, although FABRIC takes ABE ciphertext as the re-encrypted ciphertext, it still enjoy the constant computation cost in all phases.

Furthermore, it can be observed from Table IV that the sizes of secret keys in IBC are constant, while the sizes of secret keys in ABC are proportional to the sizes of the attribute sets. To thoroughly make clear the differences of these schemes in computation and communication efficiency, all these schemes are implemented with the pairing-based cryptography (PBC) library [52]. Specifically, the experimental simulations are not executed on the same platform. The IBE system is arranged to operate in a resource-constrained mobile platform with a processor of HUAWEI Kirin 990 as well as 8 GB of memory. The remaining experimental scenarios run on a platform equipped with an Intel Core i5-8400@2.80 GHz Processor as well as 16 GB of memory. We use the parameters setting from "d224.param" in the PBC library to initialize the asymmetric pairing group, and the parameters in "a1.param" are used to initialize the symmetric pairing group. Then, Fig. 3 demonstrates the computation costs in encryption, re-encryption, decryption-I, and decryption-II phase of all these schemes, respectively. Fig. 4 shows the communication costs of the public parameters, the ciphertext, the re-encrypted ciphertext, and the re-encryption key. According to the results demonstrated in Fig. 3(a) and (c), during the encryption and decryption-I phase, the computation overhead of schemes [26], [45], [48] increase with the number of attributes, while only FABRIC and IBET [24] enjoy constant computation overhead. It is worth noting that, FABRIC requires

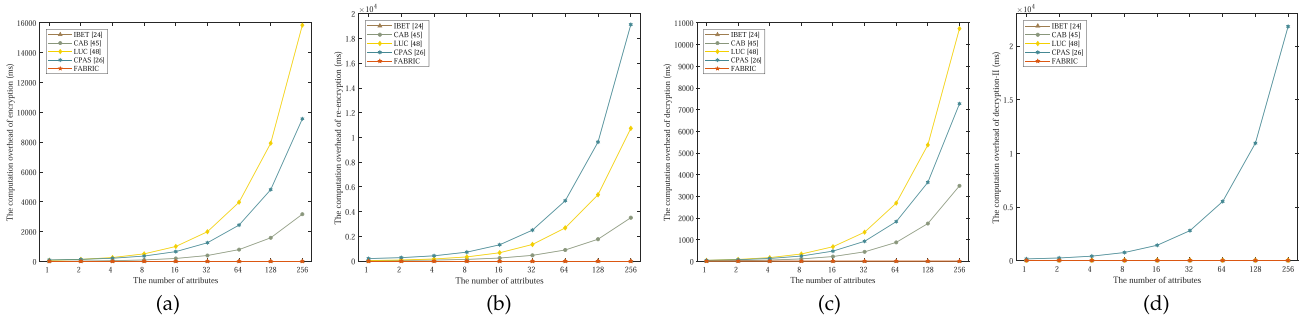

Fig. 3. The computation overhead. (a) The time-cost of encryption (b) The time-cost of re-encryption (c) The time-cost of decryption-I (d) The time-cost of decryption-II.

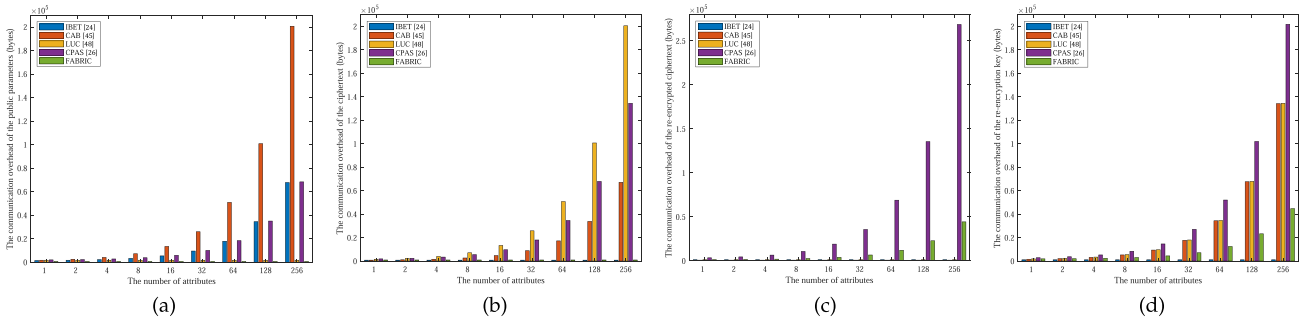

Fig. 4. The communication overhead. (a) The size of public parameters (b) The size of ciphertext (c) The size of re-encrypted ciphertext (d) The size of re-encryption key.

the lowest computation overhead in that phase. Even in an environment with limited resources, FABRIC still demonstrates its advantage of encryption overhead from IBC systems. Furthermore, all other schemes [26], [45], [48] and FABRIC achieve constant computation overhead during the decryption-II phase. Moreover, the computation overhead of FABRIC is lower than IBET [24] and LUC [48], but slightly higher than CAB [45]. The experimental results show that the efficiency of IBC system is retained during encryption. Also, the flexibility of ABC system is demonstrated during the decryption process. Overall, Fig. 3 shows the advantage of FABRIC in computation overhead of the cross-system scenario with unbalanced resources, as it can effectively and swiftly transform IBC ciphertext into flexible ABC ciphertext, connecting both systems efficiently. As for the communication overhead, it can be observed from Fig. 4(a) that only FABRIC and LUC [48] enjoy the constant size of the public parameters, which is significantly shorter than IBET [24], CAB [45], and CPAS [26]. Moreover, the ciphertext length of FABRIC is a constant value, which is slightly longer than IBET [24] but much shorter than other schemes [26], [45], [48]. While the re-encrypted ciphertext length of FABRIC increases with the increase of attributes, which is longer than schemes [24], [45], [48]. However, FABRIC's re-encrypted ciphertext is indispensable for flexible data sharing in the cloud environment, thus it can be accepted by those data users with unlimited resources. Furthermore, as a scheme that also takes ABE ciphertext as the re-encrypted ciphertext, the re-encrypted ciphertext length of FABRIC is much shorter than CPAS [26]. Briefly speaking, FABRIC enjoys excellent performance in efficiency, security, and functionality, which makes it practical and effective in real-life scenarios.

VII. Conclusion

The existing encryption schemes have the failure of simultaneously achieving the efficiency of encryption and flexible data sharing. To conquer this problem, we proposed FABRIC – a fast and secure unbounded cross-domain proxy re-encryption scheme. FABRIC not only eliminates the need to pre-define the length of attributes during the system initialization, but also keeps the computation overhead in the encryption, re-encryption, and decryption phase constant as the number of attributes increases. Moreover, FABRIC is formally proved to be adaptively secure under the DLIN assumption. Finally, the theoretical analysis as well as the experiment disclose that FABRIC enjoys excellent efficiency and practicality for secure data sharing in real-life scenarios. The secure access delegation from one ABE ciphertext to another ABE ciphertext with different access policies is regarded as our future work. Another future work could be the constructions of cross-domain proxy re-encryption schemes from arithmetic span programs with more expressive access policy.

ACKNOWLEDGMENTS

The authors would like to thank anonymous reviewers for their valuable comments and constructive remarks.

REFERENCES

[1] M. Armbrust et al., "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, pp. 50–58, 2010.
[2] K. Xue, N. Gai, J. Hong, D. S. Wei, P. Hong, and N. Yu, "Efficient and secure attribute-based access control with identical sub-policies frequently used in cloud storage," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 1, pp. 635–646, Jan./Feb. 2022.
[3] Amazon, "Amazon simple storage service (Amazon s3)," 2021, Accessed: Jun. 01, 2021. [Online]. Available: https://aws.amazon.com/s3/
[4] Microsoft, "Azure storage service," 2021, Accessed: Jun. 01, 2021. [Online]. Available: https://azure.microsoft.com/
[5] Apple, "iCloud," 2021, Accessed: Jun. 01, 2021. [Online]. Available: https://www.icloud.com/
[6] K. Chard, K. Bubendorfer, S. Caton, and O. F. Rana, "Social cloud computing: A vision for socially motivated resource sharing," *IEEE Trans. Serv. Comput.*, vol. 5, no. 4, pp. 551–563, Fourth Quarter 2012.
[7] M. Ali, S. U. Khan, and A. V. Vasilakos, "Security in cloud computing: Opportunities and challenges," *Inf. Sci.*, vol. 305, pp. 357–383, 2015.
[8] P. Muncaster, "Over a third of firms have suffered a cloud attack," 2019, Accessed: Jun. 01, 2021. [Online]. Available: https://www.infosecurity-magazine.com/news/over-third-firms-have-suffered/
[9] G. Anthes, "Security in the cloud," *Commun. ACM*, vol. 53, no. 11, pp. 16–18, 2010.
[10] K. Popović and Ž. Hocenski, "Cloud computing security issues and challenges," in *Proc. 33rd Int. Conv. Mipro*, 2010, pp. 344–349.
[11] J. Hong, K. Xue, N. Gai, D. S. Wei, and P. Hong, "Service outsourcing in F2C architecture with attribute-based anonymous access control and bounded service number," *IEEE Trans. Dependable Secure Comput.*, vol. 17, no. 5, pp. 1051–1062, Sep./Oct. 2020.
[12] K. Ren, C. Wang, and Q. Wang, "Security challenges for the public cloud," *IEEE Internet Comput.*, vol. 16, no. 1, pp. 69–73, Jan./Feb. 2012.
[13] J. Hong et al., "TAFC: Time and attribute factors combined access control for time-sensitive data in public cloud," *IEEE Trans. Serv. Comput.*, vol. 13, no. 1, pp. 158–171, Jan./Feb. 2020.
[14] K. Xue, W. Chen, W. Li, J. Hong, and P. Hong, "Combining data owner-side and cloud-side access control for encrypted cloud storage," *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 8, pp. 2062–2074, Aug. 2018.
[15] Y. Xue, K. Xue, N. Gai, J. Hong, D. S. Wei, and P. Hong, "An attribute-based controlled collaborative access control scheme for public cloud storage," *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 11, pp. 2927–2942, Nov. 2019.
[16] W. Li, K. Xue, Y. Xue, and J. Hong, "TMACS: A robust and verifiable threshold multi-authority access control system in public cloud storage," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 5, pp. 1484–1496, May 2016.
[17] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," in *Proc. Int. Conf. Theory Appl. Cryptogr. Techn.*, 1998, pp. 127–144.
[18] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," *ACM Trans. Inf. Syst. Secur.*, vol. 9, no. 1, pp. 1–30, 2006.
[19] J. Shao and Z. Cao, "CCA-secure proxy re-encryption without pairings," in *Proc. Int. Workshop Public Key Cryptography*, 2009, pp. 357–376.
[20] G. Ateniese, K. Benson, and S. Hohenberger, "Key-private proxy re-encryption," in *Proc. Cryptographers' Track RSA Conf.*, 2009, pp. 279–294.
[21] Z. Qin, H. Xiong, S. Wu, and J. Batamuliza, "A survey of proxy re-encryption for secure data sharing in cloud computing," *IEEE Trans. Serv. Comput.*, to be published, doi: 10.1109/TSC.2016.2551238.
[22] P. Xu, T. Jiao, Q. Wu, W. Wang, and H. Jin, "Conditional identity-based broadcast proxy re-encryption and its application to cloud Email," *IEEE Trans. Comput.*, vol. 65, no. 1, pp. 66–79, Jan. 2016.
[23] Y. Zhou, H. Deng, Q. Wu, B. Qin, J. Liu, and Y. Ding, "Identity-based proxy re-encryption version 2: Making mobile access easy in cloud," *Future Gener. Comput. Syst.*, vol. 62, pp. 128–139, 2016.
[24] H. Deng et al., "Identity-based encryption transformation for flexible sharing of encrypted data in public cloud," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 3168–3180, Apr. 2020. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9056799
[25] K. Liang, L. Fang, D. S. Wong, and W. Susilo, "A ciphertext-policy attribute-based proxy re-encryption scheme for data sharing in public clouds," *Concurrency Comput. Pract. Exp.*, vol. 27, no. 8, pp. 2004–2027, 2015.
[26] K. Liang et al., "A secure and efficient ciphertext-policy attribute-based proxy re-encryption for cloud data sharing," *Future Gener. Comput. Syst.*, vol. 52, pp. 95–108, 2015.
[27] C. Ge, W. Susilo, L. Fang, J. Wang, and Y. Shi, "A CCA-secure key-policy attribute-based proxy re-encryption in the adaptive corruption model for dropbox data sharing system," *Des. Codes Cryptogr.*, vol. 86, no. 11, pp. 2587–2603, 2018.
[28] J. Chen, J. Gong, L. Kowalczyk, and H. Wee, "Unbounded ABE via bilinear entropy expansion, revisited," in *Proc. Annu. Int. Conf. Theory Appl. Cryptographic Techn.*, 2018, pp. 503–534.
[29] A. Lewko and B. Waters, "Unbounded hibe and attribute-based encryption," in *Proc. Annu. Int. Conf. Theory Appl. Cryptographic Techn.*, 2011, pp. 547–567.
[30] T. Okamoto and K. Takashima, "Fully secure unbounded inner-product and attribute-based encryption," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, 2012, pp. 349–366.
[31] N. Attrapadung, "Dual system encryption via doubly selective security: Framework, fully secure functional encryption for regular languages, and more," in *Proc. Annu. Int. Conf. Theory Appl. Cryptographic Techn.*, 2014, pp. 557–577.
[32] J. Chen, R. Gay, and H. Wee, "Improved dual system abe in prime-order groups via predicate encodings," in *Proc. Annu. Int. Conf. Theory Appl. Cryptographic Techn.*, 2015, pp. 595–624.
[33] N. Attrapadung, "Dual system encryption framework in prime-order groups via computational pair encodings," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, 2016, pp. 591–623.
[34] S. Agrawal and M. Chase, "FAME: Fast attribute-based message encryption," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2017, pp. 665–682.
[35] M. Green and G. Ateniese, "Identity-based proxy re-encryption," in *Proc. Int. Conf. Appl. Cryptogr. Netw. Secur.*, 2007, pp. 288–306.
[36] C.-K. Chu and W.-G. Tzeng, "Identity-based proxy re-encryption without random oracles," in *Proc. Int. Conf. Inf. Secur.*, 2007, pp. 189–202.
[37] Y. Ren, D. Gu, S. Wang, and X. Zhang, "Hierarchical identity-based proxy re-encryption without random oracles," *Int. J. Found. Comput. Sci.*, vol. 21, no. 06, pp. 1049–1063, 2010.
[38] K. Liang, J. K. Liu, D. S. Wong, and W. Susilo, "An efficient cloud-based revocable identity-based proxy re-encryption scheme for public clouds data sharing," in *Proc. Eur. Symp. Res. Comput. Secur.*, 2014, pp. 257–272.
[39] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proc. Annu. Int. Conf. Theory Appl. Cryptographic Techn.*, 2005, pp. 457–473.
[40] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. 13th ACM Conf. Comput. Commun. Secur.*, 2006, pp. 89–98.
[41] X. Liang, Z. Cao, H. Lin, and J. Shao, "Attribute based proxy re-encryption with delegating capabilities," in *Proc. 4th Int. Symp. Inf., Comput. Commun. Secur.*, 2009, pp. 276–286.
[42] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute based data sharing with attribute revocation," in *Proc. 5th ACM Symp. Inf., Comput. Commun. Secur.*, 2010, pp. 261–270.
[43] K. Zheng, Q. Zheng, P. Chatzimisios, W. Xiang, and Y. Zhou, "Heterogeneous vehicular networking: A survey on architecture, challenges, and solutions," *IEEE Commun. Surv. Tuts.*, vol. 17, no. 4, pp. 2377–2396, Fourth Quarter 2015.
[44] T. Qiu, N. Chen, K. Li, M. Atiquzzaman, and W. Zhao, "How can heterogeneous Internet of Things build our future: A survey," *IEEE Commun. Surv. Tuts.*, vol. 20, no. 3, pp. 2011–2027, Third Quarter 2018.
[45] T. Mizuno and H. Doi, "Hybrid proxy re-encryption scheme for attribute-based encryption," in *Proc. Int. Conf. Inf. Secur. Cryptol.*, 2009, pp. 288–302.
[46] P. Jiang, J. Ning, K. Liang, C. Dong, J. Chen, and Z. Cao, "Encryption switching service: Securely switch your encrypted data to another format," *IEEE Trans. Serv. Comput.*, vol. 14, no. 5, pp. 1357–1369, Sep./Oct. 2021.
[47] X. Liu, R. H. Deng, K.-K. R. Choo, and J. Weng, "An efficient privacy-preserving outsourced calculation toolkit with multiple keys," *IEEE Trans. Inf. Forensics Secur. Tut.*, vol. 11, no. 11, pp. 2401–2414, May 2016.
[48] H. Deng, Z. Qin, Q. Wu, Z. Guan, and Y. Zhou, "Flexible attribute-based proxy re-encryption for efficient data sharing," *Inf. Sci.*, vol. 511, pp. 94–113, 2020.
[49] K. He, J. Chen, Q. Zhou, R. Du, and Y. Xiang, "Secure dynamic searchable symmetric encryption with constant client storage cost," *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 1538–1549, 2021.

[50] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in *Proc. Annu. Int. Conf. Theory Appl. Cryptographic Techn.*, 2010, pp. 62–91.
[51] N. Attrapadung, "Dual system encryption via doubly selective security: Framework, fully secure functional encryption for regular languages, and more," in *Proc. Annu. Int. Conf. Theory Appl. Cryptographic Techn.*, 2014, pp. 557–577.
[52] B. Lynn, "The pairing-based cryptography (PBC) library," 2006. [Online]. Available: http://crypto.stanford.edu/pbc

Lili Wang received the BS degree from the Chengdu University of Technology, in 2019. He is currently working toward the MS degree with the School of Information and Software Engineering, University of Electronic Science and Technology of China. His research interest includes proxy re-encryption public key cryptography.

Ye Lin received the BS degree from the University of Electronic Science and Technology of China, in 2022. He is currently working toward the MS degree with the School of Information and Software Engineering, University of Electronic Science and Technology of China. His research interest includes proxy re-encryption public key cryptography.

Ting Yao received the BS degree from the Jiangxi University of Science and Technology, in 2020. She is currently working toward the MS degree with the School of Information and Software Engineering, University of Electronic Science and Technology of China. Her research interest includes proxy re-signature public key cryptography.

Hu Xiong (Senior Member, IEEE) received the PhD degree from the School of Computer Science and Engineering, University of Electronic Science and Technology of China (UESTC), in 2009. He is currently a full professor with the School of Information and Software Engineering, UESTC. His research interests include applied cryptography and cypberspace security.

Kaitai Liang received the PhD degree in computer science from the Department of Computer Science, City University of Hong Kong, Hong Kong, in 2014. He is currently an assistant professor with the Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Delft, The Netherlands. His current research interests include applied cryptography and information security, in particular, encryption, blockchain, postquantum crypto, privacy enhancing technology, and security in cloud computing.