SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks (2016)
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sec.1574

SPECIAL ISSUE PAPER

Online/offline unbounded multi-authority attribute-based encryption for data sharing in mobile cloud computing

Yinghui Zhang[1,2,3]*, Dong Zheng[1]*, Qi Li[4], Jin Li[5,6] and Hui Li[2]

[1] National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications, Xi'an 710121, China
[2] State Key Laboratory of Integrated Service Networks (ISN), Xidian University, Xi'an 710071, China
[3] State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China
[4] School of Computer Science and Technology, Nanjing University of Posts and Telecommunications, Nanjing 210023, China
[5] School of Computer Science, Guangzhou University, Guangzhou 510006, China
[6] School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China

ABSTRACT

In order to realize attribute-based data sharing in cloud computing, multi-authority attribute-based encryption (MA-ABE) is extremely attractive. However, most of the existing MA-ABE schemes cannot support a fully large attribute universe and are not suitable for resource-constrained mobile data owners in that the computation cost in secret key generation and encryption is extremely heavy. To tackle the earlier challenges, we propose an online/offline MA-ABE scheme, which realizes both the online/offline secret key generation and the online/offline encryption while supporting a fully large attribute universe. In the offline phase, one global-identity authority and multiple attribute authorities do the majority of the work to issue attribute secret keys before knowing users' global identity and attributes. The data owner can perform most of the encryption computation tasks before knowing the actual message and access structure. Furthermore, the online phase can rapidly assemble the final decryption key and ciphertexts when related specifications become known. Particularly, global-identity authority and attribute authorities need not to cooperate in the whole process. Our online/offline MA-ABE scheme allows the access policies encoded in linear secret sharing schemes. The formal selective security proof and extensive performance analysis indicate that our scheme is very suitable for data sharing in mobile cloud computing. Copyright © 2016 John Wiley & Sons, Ltd.

KEYWORDS
data sharing; attribute-based encryption; online/offline key; online/offline encryption; multi-authority; unbounded universe

***Correspondence**
Yinghui Zhang; Dong Zheng, West Chang'an Avenue, Chang'an District, Xi'an, Shaanxi 710121, China.
E-mail: yhzhaang@163.com

1. INTRODUCTION

Nowadays, with significant improvements of the Internet environment, an increasing number of people outsource their data to third-party cloud platforms for good experiences or cost savings. For another, data security concerns still hinder some individuals and organizations from deploying cloud computing platforms, which are not fully trusted by users in realize life. Accordingly, before outsourcing their private files to public clouds, data owners have to encrypt their files to ensure data confidentiality. Whereas, the cloud data of the ciphertext form make data sharing difficult to a large extent. Especially, it is very challenging for resource-limited users to realize fine-grained data sharing in mobile cloud computing.

To address the challenge earlier, Sahai and Waters [1] proposed a fuzzy identity-based encryption scheme, which is an attribute-based encryption (ABE) scheme supporting threshold key policies. As an attractive cryptographic primitive, ABE enables fine-grained data sharing in cloud computing. For the sake of expressiveness, Goyal *et al.* [2] proposed two kinds of ABE notions: key-policy ABE (KP-ABE) and ciphertext-policy ABE (CP-ABE). Also, they proposed a concrete KP-ABE construction. In a KP-ABE system, messages are encrypted with descriptive attributes, and users' decryption keys are generated based on a specific access structure, which is related to attributes and used to specify the user's decryption ability. The case of CP-ABE is on the contrary. In a CP-ABE system, each user can apply for a decryption key by submitting his or

her attributes to related authorities. During the encryption phase, the data owner first specifies an access structure and then encrypts the message with respect to the access structure. In both cases, a successful decryption can be performed only if the attributes satisfy the access structures. We note that CP-ABE is more suitable for realizing attribute-based data sharing in that it allows data owners themselves to specify access structures. In recent years, CP-ABE schemes have found many important applications for outsourced data security in cloud computing.

However, many existing ABE schemes only support a single-attribute authority, which individually manages all the attributes in the system. To realize distributed privilege authorization, Chase [3] proposed the first multi-authority ABE (MA-ABE) scheme, where each user has attributes issued by different attribute authorities. In a MA-ABE system, there are two important performance issues to be addressed for practical applications. For one thing, a desirable MA-ABE scheme should support a fully large attribute universe. That is, the attribute universe in the system can be an exponential scale of the security parameter. At the same time, the attributes used in encryption should not be limited for any reasons. For another, most of the existing MA-ABE schemes suffer severe efficiency drawbacks because the computation cost in key generation, encryption and decryption often increases with attribute-related parameters. Hence, in existing MA-ABE schemes, the workload of attribute authorities is extremely heavy, and these schemes are not suitable for mobile users with limited resources. In this paper, we tackle the earlier challenges simultaneously.

1.1. Our contribution

Contributions of this paper can be summarized as follows:

- Aiming to realize practical attribute-based data sharing in cloud computing, we propose the notion of online/offline multi-authority CP-ABE (OO-MA-CP-ABE) and present an online/offline multi-authority attribute-based data sharing system (OO-MA-ABDS). The key component is an OO-MA-CP-ABE scheme supporting a fully large attribute universe, in which one global-identity authority (GA) and multiple attribute authorities (AAs) are involved to decentralize the privilege authorization.
- In the proposed system, the computation required for the generation of user global-identity secret keys, the generation of user attribute secret keys and the encryption of messages are split into an offline phase and an online phase. In the offline phase, GA and AAs do the majority of the work to issue attribute secret keys before knowing users' global identity and attributes. The data owner can perform most of the encryption computation tasks before knowing the actual message and the access structure. Furthermore, the online phase can rapidly assemble the final decryption key and ciphertexts when related specifications become known.
- The technique of online/offline digital signature (OOS) is used by AAs to efficiently generate a signature on users' attribute secret keys. GA further generates users' global-identity secret keys, and hence, the decryption key for users only when the online signature is valid. Theoretical analysis and performance comparisons indicate that the proposed OO-MA-ABDS system is extremely suitable for resource-constrained users in mobile cloud computing.

1.2. Related work

In this section, we summarize the related works on ABE and online/offline cryptography.

1.2.1. Attribute-based encryption.

Because the introduction of ABE in implementing fine-grained access control systems [1], plenty of researches have been performing on flexible ABE schemes. In [2], Goyal *et al.* [2] introduced two complementary notions of ABE called KP-ABE and CP-ABE. They presented a construction of KP-ABE by generating the private key according to the monotonic access structures. However, CP-ABE is more attractive than KP-ABE in attribute-based data sharing in practice in that it enables data owners to specify an access structure over attributes and use it to encrypt files based on the corresponding public attributes. The first CP-ABE scheme was proposed by Bethencourt *et al.* [4], which is proven secure in the generic group model. To improve the security proof, Cheung and Newport [5] proposed another CP-ABE construction and proved its security in the standard model. The construction supports the access structures of AND gate on different attributes.

In order to further protect users' attribute privacy, anonymous ABE has been studied [6,7]. However, most of the existing anonymous ABE schemes suffer a severe efficiency drawback because of the direct decryption method, where users have to perform many computation tasks to check whether his or her attributes match the hidden access policy in ciphertexts or not. In order to tackle this problem, Zhang *et al.* [7] introduced a novel technique called match-then-decrypt into anonymous ABE where a matching phase is added before the decryption phase to improve the decryption efficiency. It is noted that the revocation issue is essential and difficult in ABE systems, because users may change their attributes frequently in practice and each attribute is conceivably shared by multiple users. Yu *et al.* [8] proposed a CP-ABE scheme supporting indirect revocation. Directly revocable CP-ABE and KP-ABE schemes are considered by Zhang *et al.* [9,10] and Shi *et al.* [11], respectively. For communication overhead savings, ABE with constant-size ciphertexts [12–14] are necessary. Attribute-based access control systems based on ABE were proposed in [15,16] for secure cloud storage. There are also many works proposed to make further improvements on ABE, such as ABE with user accountability [17–19] and

DOI: 10.1002/sec

expressive ABE [20]. Although having various attractive features, most of the earlier CP-ABE schemes only support a single attribute authority, which is not desirable in that users' attributes often are issued by different authorities in practice.

In order to fill the earlier gap, Chase [3] proposed several MA-ABE schemes, where each user can apply for secret keys from different attribute authorities. Since then, many researches have been performing on MA-ABE [21–26]. In MA-ABE system, an important issue of supporting large attribute universe has to be considered. In [22], Lewko *et al.* taken this issue into account and classified the ABE into two flavors: the small attribute universe and the large attribute universe. In ABE systems supporting the small attribute universe, the system public parameter size often depends on the amount of attributes in the system; and hence, the scale of the attribute universe is polynomially bounded in security parameters. In the case of large universe, the attribute universe scale can be an exponential level. However, some large universe ABE constructions [23,27], which are called semi-large ABE, have a limitation that the attributes used in encryption cannot be chosen arbitrarily. To eliminate this restriction, Lewko *et al.* [22] proposed the first unbounded KP-ABE scheme. The scheme can support a fully large attribute universe in composite order groups. Furthermore, Rouselakis *et al.* [28] proposed both KP-ABE and CP-ABE in groups of prime orders, where the attribute universe is unbounded. It is noted that the schemes [22,28] only allow a single authority. Recently, a MA-KP-ABE scheme [25] and a MA-CP-ABE [26] were constructed, and both schemes allow unbounded attribute universe.

Besides the large universe issue, efficiency concerns are also important in practical MA-ABDS. In fact, in most of the existing ABE schemes, the computation cost is very high and increases with the attribute-related parameters. In MA-ABE, the result is even more serious. ABE suitable for mobile cloud computing was proposed by Zhang *et al.* [14], which features constant computation cost and constant-size ciphertexts. The scheme has been used to realize attribute-based data sharing in mobile computing in [16]. To reduce the computation cost of ABE decryption at the user side, Green *et al.* [29] proposed an ABE scheme, which allows users to outsource most of the computation tasks in decryption to cloud servers. In the outsourced ABE [30], the authors considered the validity of computation results from cloud servers. Outsourced ABE schemes in [31,32] can support outsourced encryption and decryption simultaneously. Recently, Li *et al.* [33] further considered the outsourcing of key generation computation besides the outsourced encryption and decryption. For computation cost savings in basic cryptographic operations, Chen *et al.* [34,35] realized secure outsourcing of modular exponentiations. Especially, online/offline ABE schemes have recently been presented in [27,36]. However, all these schemes cannot support multiple AAs.

1.2.2. Online/offline cryptography.

The idea of online/offline was initiated by Even *et al.* [37] for digital signatures. Later, Shamir *et al.* [38] proposed a paradigm called hash-sign-switch based on Chameleon hashing functions to design online/offline signature schemes. An online/offline signature scheme consists of two phases, and it can efficiently enable handover authentication in wireless networks [39]. Before the message to be signed is known, the first offline phase is performed. The second online phase is performed once the message is known, and it is supposed to be very fast. In the online/offline signature schemes based on the hash-sign-switch paradigm [38], one security flaw is the key exposure problem of Chameleon hashing. To solve this problem, a special double-trapdoor hash family was proposed by Chen *et al.* [40,41], and they applied the hash-sign-switch paradigm to propose a much more efficient generic online/offline signature scheme.

The technique of online/offline encryption was introduced by Guo *et al.* [42], where they proposed an identity-based online/offline encryption (IBOOE) scheme. Note that IBOOE has been used to realize secure and efficient handover authentication in wireless networks [43]. In [42], the encryption process is split into two phases: the offline phase and the online phase. The offline phase does the vast majority of the work to encrypt a message, and it does not require the knowledge of the message to be encrypted and the receiver's identity. This division of computational tasks makes encryption affordable by mobile devices with limited computation power in that most of the works can be executed offline. A more efficient IBOOE scheme was proposed by Liu *et al.* [44]. Very recently, an improved IBOOE scheme has been proposed by Lai *et al*. [45]. They proposed an efficient transformation to obtain an online/offline encryption scheme from a traditional identity-based encryption scheme. Especially, Hohenberger *et al.* [36] proposed several online/offline ABE schemes. The first fully secure online/offline predicate encryption and ABE schemes have recently been presented by Datta *et al.* [27], in which only the online/offline encryption is considered.

1.3. Organization

The remaining of this paper is organized as follows. Some preliminaries are reviewed in Section 2. We then present the definition and security model of OO-MA-ABE in Section 3. The architecture of the proposed online/offline multi-authority attribute-based data sharing system and its concrete construction together with security results are presented in Section 4. Performance comparisons are made in Section 5. Finally, we conclude this paper in Section 6.

2. PRELIMINARIES

In this section, we first give some notations used throughout the paper and then briefly review some cryptographic backgrounds, access structures, and the notion of OOS.

2.1. Notations

In order to facilitate the understanding, we explain some notations used throughout the paper in Table I.

2.2. Cryptographic backgrounds

Definition 1 (Bilinear pairings)**.** *Let $\mathbb{G}$, $\mathbb{G}_T$ be cyclic multiplicative groups of prime order p. Let $g \in_R \mathbb{G}$ be a generator. We call $\hat{e}$ a bilinear pairing if $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a map with the following properties:*

(1) Bilinear: $\hat{e}(g^a, g^b) = \hat{e}(g, g)^{ab}$ for all $a, b \in \mathbb{Z}_p$.
(2) Non-degenerate: There exists $g_1, g_2 \in \mathbb{G}$ such that $\hat{e}(g_1, g_2) \neq 1$.
(3) Computable: There is an efficient algorithm to compute $\hat{e}(g_1, g_2)$ for all $g_1, g_2 \in \mathbb{G}$.

Definition 2 (q-type problem [28])**.** *This problem involves a challenger and an adversary. The challenger first chooses parameters of bilinear pairings p, $\mathbb{G}$, $\mathbb{G}_T$, and $\hat{e}$. Then it picks $g \in_R \mathbb{G}$ and sends the following terms to the adversary.*

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

where a, s, and $b_i (i \in [q])$ are randomly chosen from $\mathbb{Z}_p$. The challenger also flips a random coin $b \in \{0,1\}$ and sends T to the adversary, where $T = e(g,g)^{a^{q+1}s}$ if $b = 0$ and otherwise $T \in \mathbb{G}_T$. At last, the adversary outputs a guess bit $b' \in \{0,1\}$.

The advantage of the adversary in the earlier game with security parameter λ is defined as $\mathsf{Adv}_{q\text{-type}}(\lambda) = \left|\Pr[\, b' = b\,] - 1/2\right|$. The q-type assumption holds in $\mathbb{G}$ if no probabilistic polynomial time (PPT) algorithm has a non-negligible probability in the earlier game.

2.3. Access structures

Definition 3 (Access structures [46])**.** *Let $\mathcal{U}$ be a set of parties. A collection $\mathbb{A} \subseteq 2^{\mathcal{U}}$ is monotone if $\forall B \in \mathbb{A}$ and $C \in 2^{\mathcal{U}}$: if $B \subseteq C$ then $C \in \mathbb{A}$. An access structure (respectively, monotone access structure) on $\mathcal{U}$ is a collection (respectively, monotone collection) $\mathbb{A}$ of non-empty subsets of $\mathcal{U}$, that is, $\mathbb{A} \subseteq 2^{\mathcal{U}} \setminus \{\emptyset\}$. The sets in $\mathbb{A}$ are called the authorized sets; otherwise, the sets are called the unauthorized sets.*

In attribute-based encryption systems, the roles of the parties are determined by the attributes in the attribute universe $\mathcal{U}$. Therefore, the access structure $\mathbb{A}$ will contain the authorized sets of attributes.

Definition 4 (Linear secret sharing schemes [46])**.** *Let $\mathcal{U}$ be the attribute universe and $\mathbb{A}$ an access structure on $\mathcal{U}$. An LSSS can be used to represent an access structure $\mathbb{A} = (M, \rho)$, where M is an $\ell \times n$ matrix which is called the share-generating matrix and ρ maps a row of M into an attribute. An LSSS consists of two algorithms:*

- ***Share***$((M, \rho), s)$*: This algorithm is used to share a secret value s based on attributes. Considering a vector $\vec{v} = (s, y_2, \ldots, y_n)^T$, where $s \in \mathbb{Z}_p$ is the secret to be shared and $y_2, \ldots, y_n \in_R \mathbb{Z}_p$, then $\lambda_i = \vec{M}_i \cdot \vec{v}$ is a share of the secret s which belongs to the attribute $\rho(i)$.*

Table I. Meanings of symbols.

Symbol	Description
$[k]$	The integer set $\{1, 2, \cdots, k\}$.
$[k_1, k_2]$	The set $\{k_1, k_1+1, \cdots, k_2\}$ containing consecutive integers.
$\|S\|$	The cardinality of the set S.
$s \in_R S$	The variable s is chosen uniformly at random from S.
GA/AA_k	The global-identity authority / the k-th attribute authority.
U	The system attribute universe.
U_k	The attribute domain managed by AA_k
GP	The global system parameter.
GPK/GMK	The global-identity authority public key / the corresponding master secret key of GA.
$\mathsf{APK}_k/\mathsf{AMK}_k$	The attribute authority public key / the corresponding master secret key of AA_k.
$\mathsf{APK}_{\mathbb{A}}$	The attribute authority public key involved in the access structure $\mathbb{A}$.
$CT_{\text{off}}/CT_{\mathbb{A}}$	The offline ciphertext / the online ciphertext under $\mathbb{A}$.
$S_{\text{GID},k}/S_{\text{GID}}$	The attribute set of user GID issued by AA_k / the attribute set of user GID.
$\mathsf{uask}_{\text{off}}/\mathsf{uask}_{\text{on},S}$	The offline *user-attribute* secret key / the online *user-attribute* secret key of attribute set S.
$\mathsf{ugsk}_{\text{off}}$	The offline *user-global-identity* secret key from GA.
$\mathsf{ugsk}_{\text{GID}}/\mathsf{uask}_{S_{\text{GID}}}$	The user GID's final *user-global-identity* secret key / the final *user-attribute* secret key from GA in online phase.

DOI: 10.1002/sec

- ***Reconstruction***$(\lambda_1, \ldots, \lambda_\ell, (M, \rho))$*: This algorithm is used to reconstruct s from secret shares. Let* $S \in \mathbb{A}$ *be any authorized set and* $I = \{i | \rho(i) \in S\} \subseteq \{1, 2, \ldots, \ell\}$*. Then there exists coefficients* $\{\omega_i\}_{i \in I}$ *such that* $\sum_{i \in I} \omega_i \vec{M}_i = (1, 0, \ldots, 0)$*, thus we have* $\sum_{i \in I} \omega_i \lambda_i = s$*.*

2.4. Online/offline digital signature

An OOS scheme $\Sigma_{\mathbf{sign}}$ comprises five algorithms as follows:

- **SigSetup**$(1^\lambda) \rightarrow (\mathsf{SP})$: The signature setup algorithm is run by a user. It outputs the signature parameters SP by taking a security parameter λ as inputs. Note that SP is published by the user so that the other entities can obtain it.
- **SigKeyGen**$(\mathsf{SP}) \rightarrow (\mathsf{SK}, \mathsf{VK})$: This algorithm can be performed by any user based on SP to generate a matching signing and verification keys $(\mathsf{SK}, \mathsf{VK})$.
- **OffSign**$(\mathsf{SP}, \mathsf{VK}, \mathsf{SK}) \rightarrow \Sigma_{\text{off}}$: Before knowing the message to be signed, a signer takes SP, VK, and SK as inputs and runs this algorithm to generate an offline signature Σ_{off}.
- **OnSign**$(\mathsf{SP}, m, \mathsf{SK}, \Sigma_{\text{off}}) \rightarrow \Sigma_{\text{on}}$: When a message m is specified to be signed, the signer takes SP, m, SK, and Σ_{off} as inputs and runs this algorithm to rapidly assemble the final online signature Σ_{on} of m. It is noted that m is included in Σ_{on}.
- **Verify**$(\mathsf{SP}, \mathsf{VK}, \Sigma_{\text{on}}) \rightarrow (\mathsf{true}$ or $\mathsf{false})$: Upon receiving a signature Σ_{on}, the verifier checks its validity based on SP and VK. If valid, it outputs true, otherwise is returned false.

3. DEFINITION AND SECURITY MODEL

In this section, we give the definition and formalized security model of online/offline multi-authority CP-ABE.

3.1. Definition of online/offline multi-authority ciphertext-policy attribute-based encryption

- **GlobalSetup**$(1^\lambda) \rightarrow (\mathsf{GP})$: The global system setup algorithm is run by GA. It outputs the global system parameter GP by taking a security parameter λ as inputs. Note that GP is published by GA so that the other entities can obtain it.
- **GASetup**$(\mathsf{GP}) \rightarrow (\mathsf{GPK}, \mathsf{GMK})$: GA takes GP as inputs and runs this algorithm to generate the GA public key GPK and the corresponding master secret key GMK.
- **AASetup**$(\mathsf{GP}, k, U_k) \rightarrow (\mathsf{APK}_k, \mathsf{AMK}_k)$: Each AA_k takes GP, its index $k \in [K]$ and attribute universe U_k as inputs and runs this algorithm to generate the attribute authority public key APK_k and the corresponding master secret key AMK_k.
- **Encrypt$_{\text{off}}$**$(\mathsf{GP}, \mathsf{GPK}, \mathsf{APK}) \rightarrow CT_{\text{off}}$: Mobile data (MD) takes GP, GPK and APK as inputs and runs this algorithm to generate an offline ciphertext CT_{off}, where $\mathsf{APK} = \bigcup_{k \in [K]} \mathsf{APK}_k$.
- **Encrypt$_{\text{on}}$**$(\mathsf{GP}, \mathsf{APK}_{\mathbb{A}}, m, \mathbb{A}, CT_{\text{off}}) \rightarrow CT_{\mathbb{A}}$: In order to encrypt a message m under a specified access policy $\mathbb{A}$, MD takes GP, the set $\mathsf{APK}_{\mathbb{A}}$ of attribute authority public keys involved in $\mathbb{A}$, m, $\mathbb{A}$, and an offline ciphertext CT_{off} as inputs, and runs this algorithm to generate the final ciphertext $CT_{\mathbb{A}}$.
- **AAKeyGen$_{\text{off}}$**$(\mathsf{GP}, \mathsf{AMK}_k) \rightarrow \mathsf{uask}_{\text{off}}$: AA_k takes GP and AMK_k as inputs and runs this algorithm to generate an offline *user-attribute* secret key $\mathsf{uask}_{\text{off}}$.
- **AAKeyGen$_{\text{on}}$**$(\mathsf{GP}, \mathsf{GID}, \mathsf{APK}_k, S_{\text{GID},k}, \mathsf{uask}_{\text{off}}) \rightarrow \mathsf{uask}_{\text{on},S_{\text{GID},k}}$: Whenever a user GID applies for a secret key for attribute set $S_{\text{GID},k}$ from AA_k, AA_k takes GP, GID, APK_k, $S_{\text{GID},k}$, and $\mathsf{uask}_{\text{off}}$ as inputs, and runs this algorithm to generate a partial online *user-attribute* secret key $\mathsf{uask}_{\text{on},S_{\text{GID},k}}$. It is noted that the user GID's online attribute secret key is $\mathsf{uask}_{\text{on},S_{\text{GID}}} = \bigcup_{k \in [K]} \mathsf{uask}_{\text{on},S_{\text{GID},k}}$, where $S_{\text{GID}} = \bigcup_{k \in [K]} S_{\text{GID},k}$.
- **GAKeyGen$_{\text{off}}$**$(\mathsf{GP}, \mathsf{GMK}) \rightarrow \mathsf{ugsk}_{\text{off}}$: GA takes GP and GMK as inputs and runs this algorithm to generate an offline *user-global-identity* secret key $\mathsf{ugsk}_{\text{off}}$.
- **GAKeyGen$_{\text{on}}$**$(\mathsf{GP}, \mathsf{uask}_{\text{on},S_{\text{GID}}}, \mathsf{ugsk}_{\text{off}}) \rightarrow SK_{S_{\text{GID}}}$: Whenever a user GID applies for a decryption key from GA, GA takes GP, $\mathsf{uask}_{\text{on},S_{\text{GID}}}$ and $\mathsf{ugsk}_{\text{off}}$ as inputs, and runs this algorithm to generate the user GID's final *user-global-identity* secret key $\mathsf{ugsk}_{\text{GID}}$ and the final *user-attribute* secret key $\mathsf{uask}_{S_{\text{GID}}}$. Then the decryption key is $SK_{S_{\text{GID}}} = (\mathsf{ugsk}_{\text{GID}}, \mathsf{uask}_{S_{\text{GID}}})$.
- **Decrypt**$(\mathsf{GP}, CT_{\mathbb{A}}, SK_{S_{\text{GID}}}) \rightarrow m$ or $\perp$: DC takes GP, a ciphertext $CT_{\mathbb{A}}$ of a message m under $\mathbb{A}$, and a decryption key $SK_{S_{\text{GID}}}$ associated with S_{GID} as inputs, and runs this algorithm to output the message m if S_{GID} is an authorized set of $\mathbb{A}$. Otherwise, the symbol $\perp$ is returned.

3.2. Formalized security model

The security model for OO-MA-CP-ABE is defined by the game as follows, which is run between a challenger $\mathcal{B}$ and an adversary $\mathcal{A}$.

(1) **Init:** The adversary $\mathcal{A}$ commits to a challenge access structure $\mathbb{A}^*$ and sends it to the challenger $\mathcal{B}$.
(2) **Setup:** The challenger $\mathcal{B}$ chooses a sufficiently large security parameter λ and does

- Run **GlobalSetup**(1^λ) to obtain GP.
- Run **GASetup**(GP) to get $(\mathsf{GPK}, \mathsf{GMK})$
- For $k \in [K]$, run **AASetup**(GP, k, U_k) to obtain $(\mathsf{APK}_k, \mathsf{AMK}_k)$.
- Run **SigSetup**(1^λ) to get SP.

Then $\mathcal{B}$ gives GP, SP, GPK, and $\{\mathsf{APK}_k\}_{k\in[K]}$ to $\mathcal{A}$. Besides, $\mathcal{A}$ specifies a corrupted set $\mathbb{K}_c \subset \mathbb{K}$ of AAs, and $\{\mathsf{AMK}_k\}_{k\in[|\mathbb{K}_c|]}$ are returned to $\mathcal{A}$.

(3) **Phase 1:** The adversary $\mathcal{A}$ issues a polynomially bounded number of queries to the following oracles with a restriction that S_{GID} does not satisfy $\mathbb{A}^*$.

- **AAKeyGen Oracle** $\mathcal{O}_{\mathrm{AAK}}$: The adversary $\mathcal{A}$ submits a GID and an attribute list S_{GID}. For $k \in \mathbb{K}\backslash\mathbb{K}_c$, $\mathcal{B}$ returns $\mathsf{uask}_{\mathrm{on},S_{\mathrm{GID}}}$ and Σ_{on} to $\mathcal{A}$.
- **GAKeyGen Oracle** $\mathcal{O}_{\mathrm{GAK}}$: Upon receiving $\mathsf{uask}_{\mathrm{on},S_{\mathrm{GID}}}$ and Σ_{on} from $\mathcal{A}$, $\mathcal{B}$ checks its validity based on **Verify**. Note that $\mathcal{B}$ returns $\mathrm{SK}_{S_{\mathrm{GID}}} = (\mathsf{ugsk}_{\mathrm{GID}}, \mathsf{uask}_{S_{\mathrm{GID}}})$ only if $\mathsf{uask}_{\mathrm{on},S_{\mathrm{GID}}}$ is valid.

(4) **Challenge:** Once $\mathcal{A}$ decides that **Phase 1** is over, it outputs two messages m_0 and m_1 of the same length on which it wishes to be challenged under $\mathbb{A}^*$. The challenger $\mathcal{B}$ flips a random coin $b \in \{0,1\}$, computes $CT_{\mathbb{A}*} = \mathbf{Encrypt_{on}}(\mathsf{GP}, \mathsf{APK}_{\mathbb{A}}, m, \mathbb{A}, CT_{\mathrm{off}})$ and sends $CT_{\mathbb{A}*}$ to $\mathcal{A}$, where $CT_{\mathrm{off}} = \mathbf{Encrypt_{off}}(\mathsf{GP}, \mathsf{GPK}, \mathsf{APK})$.

(5) **Phase 2:** The same as **Phase 1**.

(6) **Guess:** The adversary $\mathcal{A}$ outputs a guess bit $b' \in \{0,1\}$ for b and wins the game if $b' = b$. The advantage of $\mathcal{A}$ in the earlier game with security parameter λ is defined as follows:

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{OO\text{-}MA\text{-}CP\text{-}ABE}}(\lambda) = \left|\Pr[\, b' = b\,] - 1/2\right|.$$

4. ONLINE/OFFLINE MULTI-AUTHORITY ATTRIBUTE-BASED ENCRYPTION FOR DATA SHARING IN MOBILE CLOUD COMPUTING

In this section, we first propose the system architecture of online/offline multi-authority attribute-based data sharing, then give the concrete system and security results.

4.1. System architecture

As shown in Figure 1, the system architecture of OO-MA-ABDS system consists of one GA, multiple AAs, cloud service provider (CSP), MD owner and data consumer (DC).

Subsequently, we describe the system architecture of OO-MA-ABDS system in detail.

(1) GA runs **GlobalSetup** and **SigSetup** to generate global system parameters.
(2) GA runs **GASetup** to join the system.
(3) AA runs **AASetup** and **SigKeyGen** to join the system.
(4) MD runs $\mathbf{Encrypt_{off}}$ to generate offline ciphertexts CT_{off} and make preparation for file outsourcing.
(5) MD runs $\mathbf{Encrypt_{on}}$ to generate online ciphertexts $CT_{\mathbb{A}}$ and outsource $CT_{\mathbb{A}}$ to CSP.
(6) AA runs $\mathbf{AAKeyGen_{off}}$ and **OffSign** to generate $\mathsf{uask}_{\mathrm{off}}$ and Σ_{off}, respectively.
(7) Upon receiving GID and $S_{\mathrm{GID},k}$ from MD, AA returns $\mathsf{uask}_{\mathrm{on},S_{\mathrm{GID}}}$ and Σ_{on} by respectively running $\mathbf{AAKeyGen_{on}}$ and **OnSign**.
(8) GA runs $\mathbf{GAKeyGen_{off}}$ to generate $\mathsf{ugsk}_{\mathrm{off}}$.
(9) Upon receiving Σ_{on} and $\mathsf{uask}_{\mathrm{on},S_{\mathrm{GID}}}$ from MD, GA runs **Verify** to check the validity of Σ_{on}. If and only if Σ_{on} is valid, GA runs $\mathbf{GAKeyGen_{on}}$ to return $SK_{S_{\mathrm{GID}}}$.
(10) DC downloads $CT_{\mathbb{A}}$ from CSP, and runs **Decrypt** to get a plaintext messages if $S_{\mathrm{GID},k}$ matches $\mathbb{A}$.

It is noted that the algorithms have not to be performed in the earlier sequence.

4.2. The proposed online/offline multi-authority attribute-based data sharing system

(1) **Global initialization phase.** In the system initialization phase, GA chooses a security parameter λ and describes a tuple $(\mathbb{G}, \mathbb{G}_T, p, \hat{e})$, where $\mathbb{G}$ and $\mathbb{G}_T$ are two cyclic multiplicative groups of large prime order p and $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a bilinear map. Let g be a generator of $\mathbb{G}$. Also, GA specifies an online/offline signature scheme $\Sigma_{\mathbf{sign}} = (\mathbf{SigSetup}, \mathbf{SigKeyGen}, \mathbf{OffSign}, \mathbf{OnSign}, \mathbf{Verify})$. Then GA generates global system parameters based on the following procedures:

- GA runs the algorithm $\mathbf{GlobalSetup}(1^\lambda)$: GA selects $h, u, v, \omega \in_R \mathbb{G}$ and sets global system parameters as $\mathsf{GP} = (g, h, u, v, \omega)$.
- GA runs the algorithm $\mathbf{SigSetup}(1^\lambda)$ of $\Sigma_{\mathbf{sign}}$ to obtain a signature parameter SP.

(2) **Global-identity authority initialization phase.** In the GA initialization phase, GA performs the **GASetup** algorithm with GP as inputs

- $\mathbf{GASetup}(\mathsf{GP})$: GA selects an exponent $\alpha \in_R \mathbb{Z}_p$ and computes $\mathsf{GPK} = \hat{e}(g,g)^{\alpha}$. Then it publishes GPK and keeps $\mathsf{GMK} = \alpha$ secret.

(3) **Attribute authorities initialization phase.** In the AA initialization phase, for each $k \in [K]$, AA_k takes as inputs GP, k, U_k, and does the following procedures:

- AA_k performs the algorithm $\mathbf{AASetup}(\mathsf{GP}, k, U_k)$: AA_k selects an exponent $\alpha_k \in_R \mathbb{Z}_p$ and computes $\mathsf{APK}_k = (u^{\alpha_k}, h^{\alpha_k})$ and $\mathsf{AMK}_k = \alpha_k$.

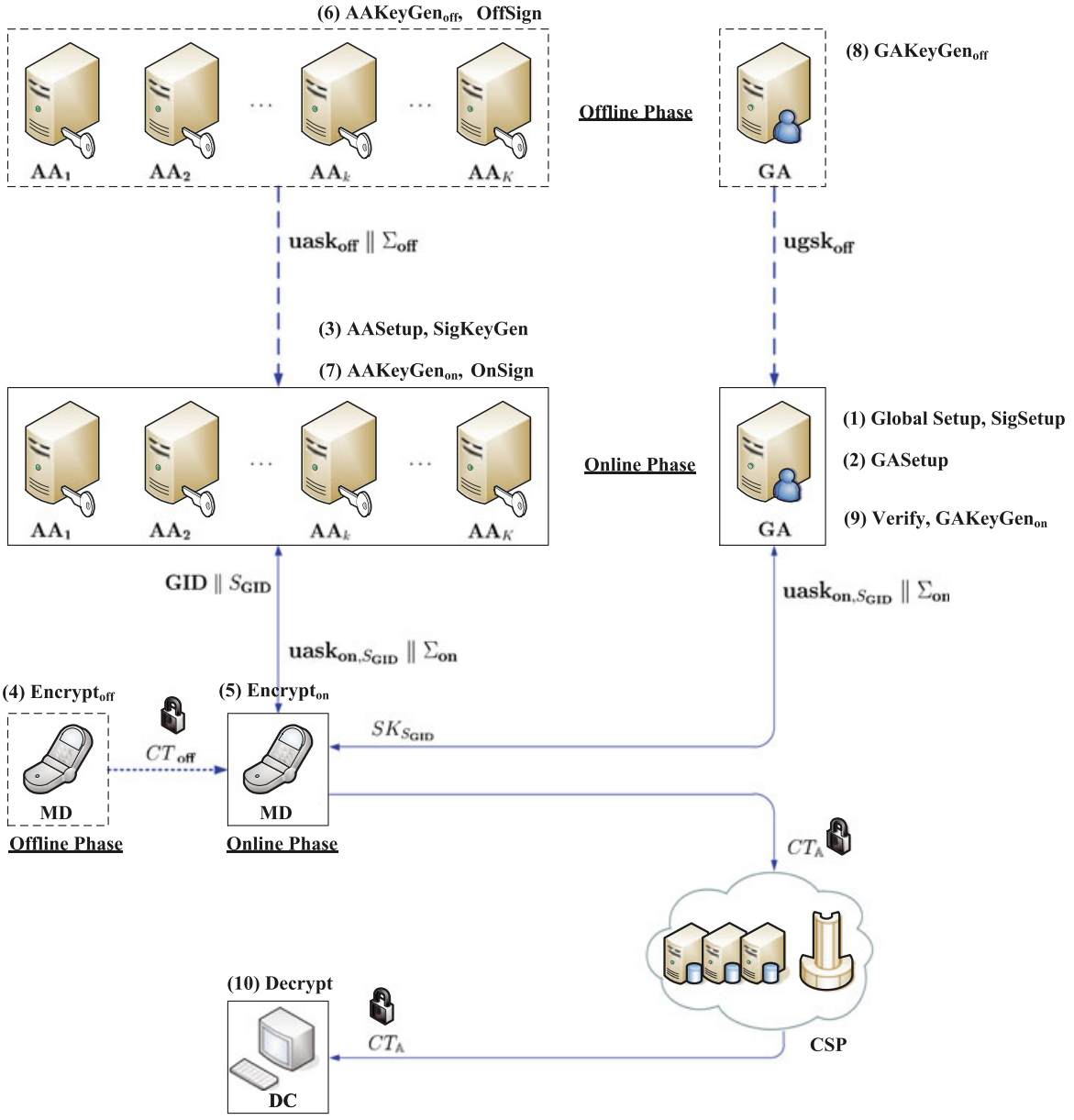


Figure 1. System architecture of online/offline multi-authority attribute-based data sharing system.

- AA_k runs the algorithm **SigKeyGen**(SP) of $\Sigma_{\mathbf{sign}}$ to obtain a signing-verification key pair (ASK_k, AVK_k).
- AA_k publishes APK_k, AVK_k and secretly keeps AMK_k and ASK_k.

(4) **Offline outsourcing preparation phase.** In the offline preparation phase of file outsourcing, MD takes GP, GPK, and $APK = \bigcup_{k\in[K]} APK_k$ as inputs and generates immediate offline ciphertexts CT_{off} based on the **Encrypt$_{off}$** algorithm as follows:

- **Encrypt$_{off}$**(GP, GPK, APK): Firstly, MD chooses $s \in_R \mathbb{Z}_p$, computes $K_m = GPK^s$ and $C'_0 = g^s$. Then MD sets $IT_{main} = (s, K_m, C'_0)$ as a main part of offline ciphertexts. For $j \in [J]$, MD picks $\lambda'_j, x_j, t_j \in_R \mathbb{Z}_p$, computes $C'_{j,1} = \left((u')^{x_j} h'\right)^{-t_j}$, $C'_{j,2} = g^{t_j}$ and $C'_{j,3} = \omega^{\lambda'_j} v^{t_j}$, where u' and h' are the attribute authority public key of some AA, and J is used by MD to determine the size of the offline ciphertext pool. Note that $APK_k = \left(u^{\alpha_k}, h^{\alpha_k}\right)$ corresponds to AA_k. Furthermore, MD sets $IT_{att,j} = \left(\lambda'_j, x_j, t_j, C'_{j,1}, C'_{j,2}, C'_{j,3}\right)$ as an attribute part of offline ciphertexts. Finally, MD sets $CT_{off} = (IT_{main}, IT_{att})$, where $IT_{att} = \{IT_{att,j}\}_{j\in[J]}$. We

note that CT_{off} constitutes an immediate offline ciphertext pool and it can be updated by MD if necessary.

(5) **Online file outsourcing phase.** Before outsourcing a file $m \in \mathbb{G}_T$ to CSP, MD can specify an access policy $\mathbb{A}$, encrypt m with respect to $\mathbb{A}$, and then upload the ciphertext to CSP. Therefore, in the online file outsourcing phase, MD chooses related offline modules from the CT_{off} pool. Besides, MD takes GP, the set $\mathsf{APK}_{\mathbb{A}}$ of attribute authority public keys involved in $\mathbb{A}$, m, and $\mathbb{A}$ as inputs, and runs the algorithm **Encrypt$_{\text{on}}$** to generate the final ciphertext $CT_{\mathbb{A}}$. Note that $\mathbb{A} = (M, \rho)$ is encoded in an LSSS policy, where $M \in \mathbb{Z}_p^{\ell \times n}$ and $\rho : [\ell] \to \mathbb{Z}_p$.

- **Encrypt$_{\text{on}}$**$(\mathsf{GP}, \mathsf{APK}_{\mathbb{A}}, m, \mathbb{A}, CT_{\text{off}})$: MD chooses any one offline main module $\mathsf{IT}_{\text{main}} = (s, K_m, C_0') = (s, \hat{e}(g,g)^{\alpha s}, g^s)$. Then MD chooses $y_2, \cdots, y_n \in_R \mathbb{Z}_p$, sets $\vec{y} = (s, y_2, \cdots, y_n)^{\mathrm{T}}$ and computes the share vector $\vec{\lambda} = (\lambda_1, \lambda_2, \cdots, \lambda_\ell)^{\mathrm{T}} = M\vec{y}$. In addition, for $j \in [\ell]$, suppose $\rho(j)$ corresponds to an attribute controlled by AA_k, MD chooses some offline attribute modules $\mathsf{IT}_{\text{att},j}$ from the CT_{off} pool, where $\mathsf{IT}_{\text{att},j} = \left(\lambda_j', x_j, t_j, C_{j,1}', C_{j,2}', C_{j,3}'\right)$ with $C_{j,1}' = \left((u^{\alpha_k})^{x_j} h^{\alpha_k}\right)^{-t_j}$, $C_{j,2}' = g^{t_j}$ and $C_{j,3}' = \omega^{\lambda_j'} v^{t_j}$. MD sets $C = m \cdot K_m$, $C_0 = C_0'$, $C_{j,1} = C_{j,1}'$, $C_{j,2} = C_{j,2}'$, $C_{j,3} = C_{j,3}'$, $C_{j,4} = \lambda_j - \lambda_j'$ and $C_{j,5} = -t_j \cdot (\rho(j) - x_j)$. Finally, the ciphertext of m under $\mathbb{A}$ is $CT_{\mathbb{A}} = (\mathbb{A}, C, C_0, \{C_{j,1}, C_{j,2}, C_{j,3}, C_{j,4}, C_{j,5}\}_{j \in [\ell]})$, which is outsourced to CSP by MD.

(6) **Offline user AAKeyGen phase.** In the offline phase of user's attribute secret key generation, to generate an offline *user-attribute* secret key $\mathsf{uask}_{\text{off}}$, each AA_k runs **AAKeyGen$_{\text{off}}$** algorithm with GP and AMK_k as inputs, and runs **OffSign** with SP, ASK_k and AVK_k as inputs in the following.

- **AAKeyGen$_{\text{off}}$**$(\mathsf{GP}, \mathsf{AMK}_k)$: At first, each AA_k chooses $r_i, \tilde{x}_i \in_R \mathbb{Z}_p$ for $i \in [|U_k|]$, computes $K_{i,1}' = g^{\frac{r_i}{\alpha_k}}$ and $K_{i,2}' = (u^{\tilde{x}_i} h)^{r_i}$, then sets $\mathsf{uask}_{\text{off}} = \left\{r_i, \tilde{x}_i, K_{i,1}', K_{i,2}'\right\}_{i \in [|U_k|]}$.
- AA_k runs **OffSign**$(\mathsf{SP}, \mathsf{AVK}_k, \mathsf{ASK}_k)$ to get Σ_{off}.

(7) **Online user AAKeyGen phase.** In the online phase of user's attribute secret key generation, to generate an online *user-attribute* secret key $\mathsf{uask}_{\text{on},S_{\text{GID}}}$, each AA_k runs **AAKeyGen$_{\text{on}}$** algorithm with GP, GID, APK_k $S_{\text{GID},k}$, and $\mathsf{uask}_{\text{off}}$ as inputs, and runs **OnSign** with SP, ASK_k and Σ_{off} as inputs in the following, where $\mathsf{uask}_{\text{off}} = \{r_i, \tilde{x}_i, K_{i,1}', K_{i,2}'\}_{i \in [|U_k|]}$.

- **AAKeyGen$_{\text{on}}$**$(\mathsf{GP}, \mathsf{GID}, \mathsf{APK}_k, S_{\text{GID},k}, \mathsf{uask}_{\text{off}})$: For $i \in [|S_{\text{GID},k}|]$, AA_k computes $K_{i,1} = K_{i,1}'$, $K_{i,2} = K_{i,2}'$ and $\tilde{K}_{i,2} = r_i \cdot (\mathrm{att}_i - \tilde{x}_i)$, then sets $\mathsf{uask}_{\text{on},S_{\text{GID},k}} = \left\{K_{i,1}, K_{i,2}, \tilde{K}_{i,2}\right\}_{i \in [|S_{\text{GID},k}|]}$. It is noted that the user GID's online attribute secret key is $\mathsf{uask}_{\text{on},S_{\text{GID}}} = \bigcup_{k \in [K]} \mathsf{uask}_{\text{on},S_{\text{GID},k}}$.
- AA_k firstly sets $m_{\text{sig}} = \mathsf{GID} \parallel \mathsf{APK}_k \parallel S_{\text{GID},k} \parallel \mathsf{uask}_{\text{on},S_{\text{GID}}} \parallel \Sigma_{\text{off}}$, and then runs the online signing algorithm **OnSign**$(\mathsf{SP}, m_{\text{sig}}, \mathsf{ASK}_k, \Sigma_{\text{off}})$ to get signature Σ_{on}. It is default that the signature message m_{sig} is contained in Σ_{on}.

(8) **Offline user GAKeyGen phase.** In the offline phase of user's global identity secret key generation, to generate an offline *global-identity* secret key $\mathsf{ugsk}_{\text{off}}$, GA runs **GAKeyGen$_{\text{off}}$** algorithm with GP and GMK as inputs in the following:

- **GAKeyGen$_{\text{off}}$**$(\mathsf{GP}, \mathsf{GMK})$: GA chooses $r \in_R \mathbb{Z}_p$, computes $K_0' = g^{\alpha} \omega^r$, $K_3' = g^r$ and $D = v^{-r}$. Then GA sets $\mathsf{ugsk}_{\text{off}} = (K_0', K_3', D)$.

(9) **Online user GAKeyGen phase.** In the online phase of user's global identity secret key generation, to generate a decryption key $SK_{S_{\text{GID}}}$ for the user GID with attributes S_{GID}, GA runs **Verify** with SP, AVK_k, and Σ_{on} as inputs for each AA_k involved in S_{GID}, and runs **GAKeyGen$_{\text{on}}$** with GP, $\mathsf{uask}_{\text{on},S_{\text{GID}}}$ and $\mathsf{ugsk}_{\text{off}}$ as inputs as follows:

- If and only if **Verify**$(\mathsf{SP}, \mathsf{AVK}_k, \Sigma_{\text{on}}) = \mathsf{true}$, that is, Σ_{on} is a valid signature, GA proceeds.
- **GAKeyGen$_{\text{on}}$**$(\mathsf{GP}, \mathsf{uask}_{\text{on},S_{\text{GID}}}, \mathsf{ugsk}_{\text{off}})$: GA firstly chooses $\xi_i \in_R \mathbb{Z}_p$ for $i \in [|S_{\text{GID}}|]$, suppose $\mathrm{att}_i \in U_k$, sets $K_0 = K_0' = g^{\alpha} \omega^r$, $K_{i,1} = (K_{i,1})^{\xi_i} = g^{\frac{r_i \xi_i}{\alpha_k}}$, $K_{i,2} = (K_{i,2} \cdot u^{\tilde{K}_{i,2}})^{\xi_i}$. $D = (u^{\mathrm{att}_i} h)^{r_i \xi_i} \cdot v^{-r}$ and $K_3 = K_3' = g^r$, then sets the user GID's final *user-global-identity* secret key as $\mathsf{ugsk}_{\text{GID}} = (K_0, K_3)$ and the final *user-attribute* secret key is $\mathsf{uask}_{S_{\text{GID}}} = \{K_{i,1}, K_{i,2}\}_{i \in [|S_{\text{GID}}|]}$. Finally, the decryption key is $SK_{S_{\text{GID}}} = (\mathsf{GID}, S_{\text{GID}}, \mathsf{ugsk}_{\text{GID}}, \mathsf{uask}_{S_{\text{GID}}})$.

(10) **File access phase.** Data consumer downloads a ciphertext $CT_{\mathbb{A}}$ from CSP and runs the algorithm **Decrypt** with GP, $CT_{\mathbb{A}}$, and $SK_{S_{\text{GID}}}$ as inputs to recover the corresponding plaintext message.

- **Decrypt**$(\mathsf{GP}, CT_{\mathbb{A}}, SK_{S_{\text{GID}}})$: If S_{GID} is not an authorized set of $\mathbb{A}$, DC aborts, and it returns $\perp$. Otherwise, DC computes constants $\{\omega_j \in \mathbb{Z}_p\}_{j \in I}$ such that $\sum_{j \in I} \omega_j \vec{M}_j = (1, 0, \ldots, 0)$,

where $I = \{j|\rho(j) \in S_{\mathrm{GID}}\} \subseteq [\ell]$ and $\vec{M}_j$ denotes the j-th row of M. It is noted that these constants exist in that S_{GID} is authorized by $\mathbb{A}$. Then DC computes

$$B = \frac{\hat{e}\,(K_0, C_0)}{\prod_{j\in I} T_j^{\omega_j}} = \hat{e}(g,g)^{\alpha s}$$

where

$$T_j = \hat{e}\left(K_{i,1}, C_{j,1} \cdot (u^{\alpha_k})^{C_{j,5}}\right) \times \hat{e}\left(K_{i,2}, C_{j,2}\right) \hat{e}\left(K_3, C_{j,3} \cdot \omega^{C_{j,4}}\right)$$

and the index of the attribute $\rho(j)$ in S_{GID} is i and k indicates that $\rho(j)$ is issued by the k-th attribute authority AA_k. Finally, DC gets $M = C/B$.

4.3. Security results

The security of the proposed OO-MA-ABDS system is given by the Theorem 1 as follows:

Theorem 1. If the adopted OOS scheme is existentially unforgeable, then our OO-MA-ABDS system is secure in the standard model against the selective access structures and chosen messages attackers in the proposed security model under the q-type assumption in $\mathbb{G}$.

Proof. The proposed OO-MA-ABDS system is based on a potential OO-MA-CP-ABE scheme, which is denoted by Π. In the following, we will show that any PPT attacker $\mathcal{A}$ with a non-negligible advantage ϵ in the proposed security model against Π can be used to design a PPT simulator $\mathcal{B}$, which can break the q-type assumption with advantage ϵ. The simulator $\mathcal{B}$ plays the challenger and interacts with $\mathcal{A}$. The simulation proceeds as follows:

(1) **Init:** The challenger $\mathcal{B}$ obtains the given terms of the q-type assumption. In addition, the adversary $\mathcal{A}$ gives a challenge access structure $\mathbb{A}^* = (M^*, \rho^*)$ to $\mathcal{B}$. We note that the index of the $\ell \times n$ matrix M^* satisfies $\ell, n \le q$ and $\rho^* : [\ell] \to \mathbb{Z}_p$.

(2) **Setup:** The challenger $\mathcal{B}$ chooses a sufficiently large security parameter λ, and does

- **GlobalSetup**(1^λ): $\mathcal{B}$ sets $g = g$ and $\omega = g^a$. Then $\mathcal{B}$ chooses $h', u', v' \in_R \mathbb{Z}_p$, and returns to the parameters as follows based on the given terms in the assumption.

$$h = g^{h'} \cdot \prod_{(j,k)\in[\ell,n]} \left(g^{a^k/b_j^2}\right)^{-\rho^*(j)M^*_{j,k}}$$

$$u = g^{u'} \cdot \prod_{(j,k)\in[\ell,n]} \left(g^{a^k/b_j^2}\right)^{M^*_{j,k}},$$

$$v = g^{v'} \cdot \prod_{(j,k)\in[\ell,n]} \left(g^{a^k/b_j}\right)^{M^*_{j,k}}$$

Also, an existentially unforgeable OOS scheme $\Sigma_{\mathbf{oos}}$ is adopted. $\mathcal{B}$ sets $\mathsf{GP} = (g, h, u, v, \omega)$.

- **SigSetup**(1^λ): $\mathcal{B}$ generates an online/offline signature parameter SP.

- **GASetup**(GP): $\mathcal{B}$ picks $\alpha' \in_R \mathbb{Z}_p$ and implicitly sets $\mathsf{GMK} = \alpha = a^{q+1} + \alpha'$. Then $\mathsf{GPK} = e(g,g)^{\alpha} = e(g,g)^{\alpha'} \cdot e(g^a, g^{a^q})$ is given to $\mathcal{A}$.
- **AASetup**(GP, k, U_k): For each AA_k, $\mathcal{B}$ chooses $\alpha_k \in_R \mathbb{Z}_p$ and sets $\mathsf{AMK}_k = \alpha_k$ and $\mathsf{APK}_k = \left(u^{\alpha_k}, h^{\alpha_k}\right)$. AA_k runs the algorithm **SigKeyGen**(SP) of $\Sigma_{\mathbf{oos}}$ to obtain a signing-verification key pair $(\mathsf{ASK}_k, \mathsf{AVK}_k)$.

Finally, $\mathcal{B}$ gives GP, SP, GPK, and $\{\mathsf{APK}_k\}_{k\in[K]}$ to $\mathcal{A}$. Besides, $\mathcal{A}$ specifies a corrupted set $\mathbb{K}_c \subset \mathbb{K}$ of AAs, and $\{\mathsf{AMK}_k\}_{k\in[|\mathbb{K}_c|]}$ are returned to $\mathcal{A}$.

(3) **Phase: 1** $\mathcal{A}$ makes queries to the following oracles with a restriction that S_{GID} is not authorized by $\mathbb{A}^*$.

- **AAKeyGen Oracle** $\mathcal{O}_{\mathrm{AAK}}$: $\mathcal{A}$ submits a GID and an attribute list S_{GID}. For $\mathrm{AA}_k \in \mathbb{K}_c$, $\mathcal{A}$ generates $\mathsf{uask}_{\mathrm{on},S_{\mathrm{GID},k}}$, and Σ_{off} itself and sends them to $\mathcal{B}$. Note that $\mathcal{B}$ can also generates $\mathsf{uask}_{\mathrm{on},S_{\mathrm{GID},k}}$ and Σ_{off} itself. Subsequently, $\mathcal{B}$ generates $\mathsf{uask}_{\mathrm{on},S_{\mathrm{GID}}}$, and Σ_{on} for $\mathcal{A}$ based on attribute authority master secret keys.
- **GAKeyGen Oracle** $\mathcal{O}_{\mathrm{GAK}}$: Upon receiving GID, $\mathsf{uask}_{\mathrm{on},S_{\mathrm{GID}}}$, and Σ_{on} from $\mathcal{A}$, $\mathcal{B}$ checks its validity based on **Verify**$(\mathsf{SP}, \mathsf{AVK}_k, \Sigma_{\mathrm{on}})$. If each value is true, $\mathcal{B}$ returns $\mathrm{SK}_{S_{\mathrm{GID}}} = (\mathsf{ugsk}_{\mathrm{GID}}, \mathsf{uask}_{S_{\mathrm{GID}}})$ to $\mathcal{A}$ by performing the following procedures. Because S_{GID} is not authorized by $\mathbb{A}^* = (A^*, \rho)$, $\mathcal{B}$ can find a vector $\vec{\omega} = (\omega_1, \omega_2, \ldots, \omega_n)^\top \in \mathbb{Z}_p^n$ such that $\omega_1 = -1$ and $\langle M_j^*, \vec{\omega}\rangle = 0$ for all $j \in I = \{j|\rho^*(j) \in S_{\mathrm{GID}} \wedge j \in [\ell]\}$. Then $\mathcal{B}$ selects $r' \in \mathbb{Z}_p$ and implicitly sets

$$r = r' + \omega_1 a^q + \omega_2 a^{q-1} + \ldots + \omega_n a^{q+1-n} = r' + \sum_{i\in[n]} \omega_i a^{q+1-i}$$

Then $\mathcal{B}$ calculates the final *user-global-identity* secret key as $\mathsf{ugsk}_{\mathrm{GID}} = (K_0, K_3)$ as follows:

$$K_0 = g^{\alpha}\omega^r = g^{a^{q+1}+\alpha'} \cdot g^{ar'} \prod_{i\in[n]} g^{\omega_i a^{q+2-i}}$$

$$= g^{\alpha'}(g^{a})^{r'}\prod_{i=2}^{n}\left(g^{a^{q+2-i}}\right)^{\omega_i},$$

$$K_3 = g^{r} = g^{r'}\prod_{i\in[n]}\left(g^{a^{q+1-i}}\right)^{\omega_i}$$

In addition, aiming to answer the final *user-attribute* secret key $\mathsf{uask}_{S_{\mathrm{GID}}} = \{K_{i,1}, K_{i,2}\}_{i\in[|S_{\mathrm{GID}}|]}$, for each $\mathrm{att}_i \in S_{\mathrm{GID}}$, $\mathcal{B}$ chooses $\tilde{\xi}_i \in \mathbb{Z}_p$ and sets $\xi_i = r_i^{-1}\xi_i'$, where $r_i \in \mathbb{Z}_p$ is chosen by $\mathcal{B}$ in $\mathcal{O}_{\mathrm{AAK}}$ in terms of the proposed scheme and ξ_i' is implicitly computed as

$$\begin{aligned}\xi_i' &= \tilde{\xi}_i + r\cdot\sum_{\substack{j'\in[\ell],\\ \rho^*(j')\notin S_{\mathrm{GID}}}}\frac{b_{j'}}{\mathrm{att}_i-\rho^*(j')}\\ &= \tilde{\xi}_i + r'\cdot\sum_{\substack{j'\in[\ell],\\ \rho^*(j')\notin S_{\mathrm{GID}}}}\frac{b_{j'}}{\mathrm{att}_i-\rho^*(j')}\\ &\quad+\sum_{\substack{(j,j')\in[n,\ell],\\ \rho^*(j')\notin S_{\mathrm{GID}}}}\frac{\omega_j b_{j'}a^{q+1-j}}{\mathrm{att}_i-\rho^*(j')}\end{aligned}$$

Obviously, ξ_i' (and hence ξ_i) is well defined for attributes in S_{GID}. Then, for each $\mathrm{att}_i \in S_{\mathrm{GID}}$, suppose att_i is managed by AA_k, $\mathcal{B}$ calculates $K_{i,1}$ as follows:

$$\begin{aligned}K_{i,1} &= g^{r_i\xi_i/\alpha_k} = g^{\xi_i'/\alpha_k}\\ &= g^{\tilde{\xi}_i/\alpha_k}\cdot\prod_{\substack{j'\in[\ell],\\ \rho^*(j')\notin S_{\mathrm{GID}}}}\left(g^{b_{j'}}\right)^{r'/\alpha_k(\mathrm{att}_i-\rho^*(j'))}\\ &\quad\cdot\prod_{\substack{(j',j)\in[\ell,n],\\ \rho^*(j')\notin S_{\mathrm{GID}}}}\left(g^{b_{j'}a^{q+1-j}}\right)^{\omega_j/\alpha_k(\mathrm{att}_i-\rho^*(j'))}\end{aligned}$$

As for $K_{i,2}$, we know that the valid form is $(u^{\mathrm{att}_i}h)^{r_i\xi_i}\cdot v^{-r}$. In the following, we show that although $\mathcal{B}$ cannot directly compute v^{-r} because of an unknown multiplication factor, it still can generate a valid $K_{i,2}$ for each $\mathrm{att}_i \in S_{\mathrm{GID}}$, which is ensured by the choose of ξ_i. That is, a factor in $\left(u^{\mathrm{att}_i}h\right)^{r_i\xi_i}$ due to ξ_i and the unknown factor in v^{-r} cancel each other in multiplication. In fact,

$$\begin{aligned}\left(u^{\mathrm{att}_i}h\right)^{r_i\xi_i} &= \left(u^{\mathrm{att}_i}h\right)^{\xi_i'}\\ &= \left(u^{\mathrm{att}_i}h\right)^{\tilde{\xi}_i}\cdot\left(K_{i,1}^{\alpha_k}/g^{\tilde{\xi}_i}\right)^{u'\mathrm{att}_i+h'}\\ &\quad\cdot\prod_{\substack{(j',j,k)\in[\ell,\ell,n],\\ \rho^*(j')\notin S_{\mathrm{GID}}}} g^{\frac{r'(\mathrm{att}_i-\rho^*(j))M_{j,k}^*a^kb_{j'}}{(\mathrm{att}_i-\rho^*(j'))b_j^2}}\\ &\quad\cdot\prod_{\substack{(i',j',j,k)\in[n,\ell,\ell,n],\\ \rho^*(j')\notin S_{\mathrm{GID}}}} g^{\frac{\omega_{i'}(\mathrm{att}_i-\rho^*(j))M_{j,k}^*a^{q+1+k-i'}b_{j'}}{(\mathrm{att}_i-\rho^*(j'))b_j^2}}\\ &\triangleq \Omega_1\cdot\prod_{\substack{(i',j)\in[n,\ell],\\ \rho^*(j)\notin S_{\mathrm{GID}}}} g^{\frac{\omega_{i'}(\mathrm{att}_i-\rho^*(j))M_{j,i'}^*a^{q+1+i'-i'}b_j}{(\mathrm{att}_i-\rho^*(j))b_j^2}}\\ &= \Omega_1\cdot\prod_{\substack{j\in[\ell],\\ \rho^*(j)\notin S_{\mathrm{GID}}}} g^{\langle\vec{\omega},M_j^*\rangle a^{q+1}/b_j}\\ &\triangleq \Omega_1\cdot\Delta\end{aligned}$$

where $\Delta = \prod_{\substack{j\in[\ell],\\ \rho^*(j)\notin S_{\mathrm{GID}}}} g^{\langle\vec{\omega},M_j^*\rangle a^{q+1}/b_j}$ cannot be directly obtained by $\mathcal{B}$. Note that Ω_1 includes the remaining part of the product and it can be obtained by $\mathcal{B}$.

On the other hand, v^{-r} can be computed as

$$\begin{aligned}v^{-r} &= v^{-r'}\cdot\left(g^{v'}\prod_{(j,k)\in[\ell,n]} g^{M_{j,k}^*a^k/b_j}\right)^{-\sum_{i\in[n]}\omega_i a^{q+1-i}}\\ &= v^{-r'}\cdot\prod_{i\in[n]}\left(g^{a^{q+1-i}}\right)^{-v'\omega_i}\\ &\quad\cdot\prod_{(i,j,k)\in[n,\ell,n]}\left(g^{a^{q+1+k-i}/b_j}\right)^{-\omega_iM_{j,k}^*}\\ &\triangleq \Omega_2\cdot\prod_{(i,j)\in[n,\ell]} g^{-\omega_iM_{j,k}^*a^{q+1}/b_j}\\ &= \Omega_2\cdot\prod_{\substack{j\in[\ell],\\ \rho^*(j)\notin S_{\mathrm{GID}}}} g^{-\langle\vec{\omega},M_j^*\rangle a^{q+1}/b_j}\\ &= \Omega_2\cdot\Delta^{-1}\end{aligned}$$

where Ω_2 includes the remaining part of the product and it can be obtained by $\mathcal{B}$.

Therefore, $\mathcal{B}$ computes $K_{i,2} = \left(u^{\mathrm{att}_i}h\right)^{r_i\xi_i}\cdot v^{-r} = \Omega_1\cdot\Omega_2$, and sets $\mathsf{uask}_{S_{\mathrm{GID}}} = \{K_{i,1}, K_{i,2}\}_{i\in[|S_{\mathrm{GID}}|]}$. Finally, $\mathcal{B}$ returns $\mathrm{SK}_{S_{\mathrm{GID}}} = (\mathsf{ugsk}_{\mathrm{GID}}, \mathsf{uask}_{S_{\mathrm{GID}}})$ to the adversary $\mathcal{A}$.

(4) **Challenge:** $\mathcal{A}$ submits two messages m_0 and m_1 of the same length on which it wishes to be challenged under $\mathbb{A}^*$. $\mathcal{B}$ flips a random coin $b \in \{0,1\}$,

computes $CT_{\mathbb{A}*}$ in the following and sends it to $\mathcal{A}$. Firstly, $\mathcal{B}$ sets $C = m_b \cdot T \cdot e(g^s, g)^{\alpha'}, C_0 = g^s$. Then $\mathcal{B}$ chooses $\{\upsilon_i \in_R \mathbb{Z}_p\}_{i\in[2,n]}$ and sets $\vec{\upsilon} = \left(s, sa + \upsilon_2, sa^2 + \upsilon_3, \ldots, sa^{n-1} + \upsilon_n\right)^\top$. Then $\vec{\lambda}' = M^* \vec{\upsilon}$, for each $j \in [\ell]$, it follows that

$$\begin{aligned}\lambda'_j &= \sum_{i\in[n]} M^*_{j,i} sa^{i-1} + \sum_{i=2}^{n} M^*_{j,i}\upsilon_i \\ &= \sum_{i\in[n]} M^*_{j,i} sa^{i-1} + \tilde{\lambda}_j\end{aligned}$$

Note that each $\tilde{\lambda}_j = \sum_{i=2}^{n} M^*_{j,i}\upsilon_i$ is known to $\mathcal{B}$. For each row $j \in [\ell]$, suppose $\rho^*(x) \in U_k$, $\mathcal{B}$ implicitly sets $t_j = -sb_j$ and computes

$$\begin{aligned}C'_{j,1} &= \left(u^{\alpha_k \rho^*(j)} h^{\alpha_k}\right)^{-t_j} \\ &= (g^{sb_j})^{-\alpha_k(u'\rho^*(j)+h')} \\ &\quad \cdot \left(\prod_{(i,k)\in[\ell,n]} g^{(\rho^*(j)-\rho^*(i))M^*_{i,k}a^k/b_i^2}\right)^{-\alpha_k sb_j} \\ &= \left(g^{sb_j}\right)^{-\alpha_k(u'\rho^*(j)+h')} \\ &\quad \cdot \prod_{\substack{(i,k)\in[\ell,n],\\ i\neq j}} \left(g^{sa^k b_j/b_i^2}\right)^{\alpha_k M^*_{i,k}(\rho^*(i)-\rho^*(j))},\end{aligned}$$

$$C'_{j,2} = g^{t_j} = \left(g^{sb_j}\right)^{-1}$$

and

$$\begin{aligned}C'_{j,3} &= \omega^{\lambda'_j} v^{t_j} \\ &= \omega^{\tilde{\lambda}_j} \cdot \prod_{i\in[n]} g^{M^*_{j,i}sa^i} \cdot \left(g^{sb_j}\right)^{-v'} \\ &\quad \cdot \prod_{(i,k)\in[\ell,n]} g^{-M^*_{i,k}a^k sb_j/b_i} \\ &= \omega^{\tilde{\lambda}_j} \cdot \left(g^{sb_j}\right)^{-v'} \cdot \prod_{i\in[n]} g^{M^*_{j,i}sa^i} \\ &\quad \cdot \prod_{k\in[n]} g^{-M^*_{j,k}sa^k b_j/b_j} \cdot \prod_{\substack{(i,k)\in[\ell,n],\\ i\neq j}} g^{-M^*_{i,k}a^k sb_j/b_i} \\ &= \omega^{\tilde{\lambda}_j} \cdot \left(g^{sb_j}\right)^{-v'} \cdot \prod_{\substack{(i,k)\in[\ell,n],\\ i\neq j}} \left(g^{sa^k b_j/b_i}\right)^{-M^*_{i,k}}\end{aligned}$$

Furthermore, $\mathcal{B}$ selects $\beta_j, \gamma_j \in_R \mathbb{Z}_p$ for $j \in [\ell]$. Suppose $\mathrm{att}_j \in U_k$, $\mathcal{B}$ sets $C_{j,1} = C'_{j,1} \cdot (u^{\alpha_k})^{-\gamma_j}$, $C_{j,2} = C'_{j,2}$, $C_{j,3} = C'_{j,3} \cdot \omega^{-\beta_j} = C'_{j,3} \cdot (g^a)^{-\beta_j}$, $C_{j,4} = \beta_j$, $C_{j,5} = \gamma_j$. Finally, $\mathcal{B}$ gives the challenge ciphertext $CT_{\mathbb{A}} = \left(\mathbb{A}, C, C_0, \{C_{j,1}, C_{j,2}, C_{j,3}, C_{j,4}, C_{j,5}\}_{j\in[\ell]}\right)$ to the adversary $\mathcal{A}$.

(5) **Phase 2:** The same as **Phase 1**.

(6) **Guess:** $\mathcal{A}$ outputs a guess bit $b' \in \{0, 1\}$ of b. If and only if $b' = b$, $\mathcal{B}$ outputs 0, that is, it claims that $T = \hat{e}(g, g)^{sa^{q+1}}$. Therefore, if $\mathcal{A}$ breaks the proposed system with a non-negligible advantage ϵ, $\mathcal{B}$ obtains probability ϵ in breaking the q-type assumption in $\mathbb{G}$.

□

5. PERFORMANCE COMPARISONS

In this section, the proposed scheme is compared with existing typical schemes [3,21–23,26,27,36] from both the security and the efficiency aspects. We summarize the comparison results in Tables II and III. To be specific, in Table II, we make a comparison in accordance with the type of access structures, the expressiveness of access structures, the security level (selective security or full security), the bilinear group type, the type of attribute universe, the characteristic of multi-authority, the support of online/offline key generation (OO KeyGen), and the support of online/offline encryption (OO Encrypt). It is worth noting that the offline key generation mode can alleviate the computation workload of GA and AAs. The offline encryption mechanism eliminates most of the computation task of users, which is suitable for resource-constrained mobile users in cloud computing. For the sake of simplicity, in Table III, we denote the number of AAs by K, the attribute set of a user by S, the amount of rows in an access structure matrix by ℓ, and the row set of access structure matrix used in decryption by I. In addition, the symbol "×" (respectively, "√") indicates that the scheme cannot (respectively, can) realize the corresponding property. In Table III, **P**, **E**, and **M** are used to respectively represent a bilinear pairing operation, an exponentiation operation and a multiplication operation in bilinear groups. Arithmetic operations in $\mathbb{Z}_p$ are ignored in comparisons without impacting the results. In Table III, each scheme is compared in terms of the system setup cost, the (online) key generation cost, the (online) encryption cost, and the decryption cost.

As shown in Table II, all these schemes are expressive and allow LSSS ciphertext policies except the tree key policies in [3,21]. As for the security level, the schemes [3,21,23,26,27,36] and ours are proven secure in the standard model, and the scheme [22] uses the random oracle model. Note that only the schemes [22,23,27] are fully secure and the others achieve selective security. The schemes [22,23] are suitable for composite order groups. Especially, only the proposed scheme and the schemes [26,36] support fully large attribute universe, and the other schemes just support small or semi-large attribute universe. As for online/offline mode, only the schemes [27,36] and ours realize online/offline encryption mechanism, in

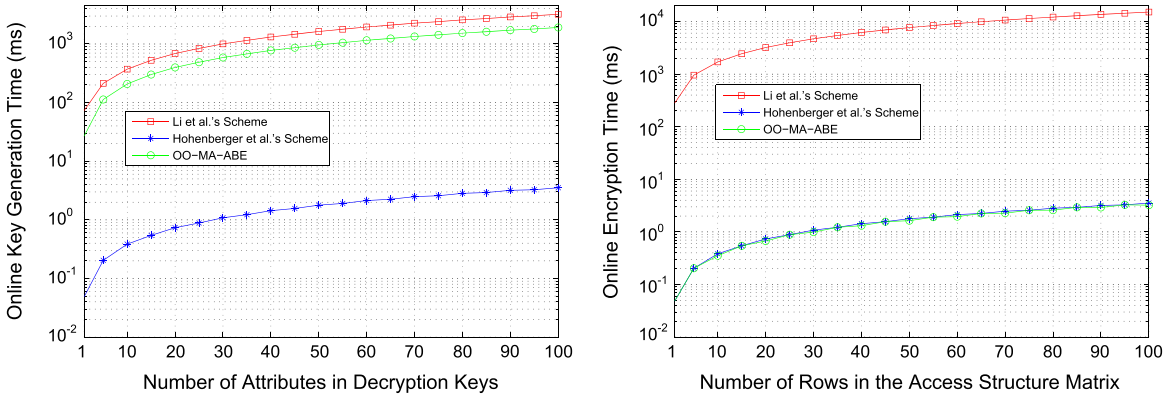
Table II. Performance comparisons between ABE cchemes.

Schemes	Policy	Expressiveness	Security	Group	Attribute universe	Multi-authority	OO KeyGen	OO Encrypt
[3,21] †	KP	Tree	Selective (S)	Prime	Semi-large	√	×	×
[22]	CP	LSSS	Full (R)	Composite	Small	√	×	×
[23]	CP	LSSS	Full (S)	Composite	Semi-large	√	×	×
[26]	CP	LSSS	Selective (S)	Prime	Fully large	√	×	×
[36]	CP §	LSSS	Selective (S)	Prime	Fully large	×	√ ‡	√
[27]	CP	LSSS	Full (S)	Prime	Semi-large	×	×	√
Ours	CP	LSSS	Selective (S)	Prime	Fully large	√	√	√

[†]The other schemes in [3,21] support threshold policies and small attribute universe. [‡]The CP-ABE scheme in [36] does not simultaneously realize online/offline key generation and encryption with provable security. [§]The other schemes in [36] support key polices.

Table III. Computation cost comparisons between fully large universe constructions.

Schemes	System setup	Key generation	Encryption	Decryption
[26]	$1\mathbf{P} + (2K+1)\mathbf{E}$	$(5\|S\|+4)\mathbf{E} + (1\|S\|+2)\mathbf{M}$	$(5\ell+2)\mathbf{E} + (2\ell+1)\mathbf{M}$	$(3\|I\|+1)\mathbf{P} + \|I\|\mathbf{E} + (3\|I\|+1)\mathbf{M}$
[36]	$1\mathbf{P} + 1\mathbf{E}$	$\|S\|\mathbf{M}$	$1\mathbf{M}$	$(3\|I\|+2)\mathbf{P} + (2\|I\|+1)\mathbf{E} + (4\|I\|+2)\mathbf{M}$
Ours	$1\mathbf{P} + (2K+1)\mathbf{E}$	$3\|S\|\mathbf{E} + \|S\|\mathbf{M}$	$1\mathbf{M}$	$(3\|I\|+1)\mathbf{P} + 3\|I\|\mathbf{E} + (5\|I\|+1)\mathbf{M}$



(a) The Online Key Generation Cost (b) The Online Encryption Cost

Figure 2. Online computation cost comparisons between fully large universe constructions.

which, however, only our scheme allows multiple AAs. The scheme [27] does not support online/offline key generation and the scheme [36] fails to realize offline key generation and offline encryption with provable security simultaneously.

Consider the support of fully large attribute universe, Table III just compares the schemes [26,36] and our scheme, where the related online key generation and online encryption computation cost is considered in the scheme [36] and ours. We can see from Table III that the proposed OO-MA-CP-ABE scheme and the scheme [36] require the same online encryption cost, which is much less than that of the scheme [26]. Our scheme can support multiple attribute authorities and the number of pairings in decryption phase is one less than that of the scheme [36].

In order to precisely evaluate the performance, we implement and compare the computation cost of the Li *et al*. scheme [26], the Hohenberger *et al*. scheme [36] with that of ours in Figure 2. Note that the vertical axis is log scale. In Figure 2(a), our simulation experiments are based on the Stanford Pairing-Based Cryptography Library (PBC) and a Linux machine with Intel Core 2 processors running at 2.40 GHz and 2G memory. In Figure 2(b), our simulation experiments are based on the Java Pairing-Based Cryptography Library and a Lenovo P780 smartphone with Android OS 4.2 operation system. In our experiments, type A pairings are adopted. We consider the worst case of access structures, which ensures that all the ciphertext components are involved in decryption. Specifically, we generate 100 distinct access structures in the form of $(A_1 \wedge A_2 \wedge \cdots \wedge A_k)$ with k increasing from 1 to 100, where each component A_i is not wildcard. In each case, a corresponding secret key that contains exact k attributes is generated. For each access structure, the experiment is repeated 100 times on the PC and 50 times on the smartphone, and the average values are used

as the final experimental results. Obviously, the experiment results indicate that the proposed OO-MA-CP-ABE scheme is very efficient considering its desirable features in Table II.

In general, the proposed OO-MA-CP-ABE scheme is the first online/offline multi-authority CP-ABE scheme. We argue that the proposed scheme is suitable for data sharing in mobile cloud data sharing.

6. CONCLUSIONS AND FUTURE WORK

Aiming at tackling the challenging issues of large universe and computation overheads in multi-authority attribute-based data sharing, we first introduce the notion and formalized security model of OO-MA-ABE, and then give a concrete OO-MA-ABDS system. The key component is an OO-MA-CP-ABE scheme supporting a fully large attribute universe, in which one GA and multiple AAs are involved to decentralize the privilege authorization. In particular, GA and AAs need not to cooperate in the whole process. The proposed OO-MA-CP-ABE scheme allows the access policies encoded in linear secret sharing schemes. Theoretical analysis and extensive performance comparisons indicate that the proposed data sharing scheme is suitable for mobile cloud computing.

It would be interesting to construct OO-MA-CP-ABE schemes supporting offline key generation, offline encryption and offline decryption simultaneously. Another possible goal for future research would be to find OO-MA-CP-ABE schemes proven secure under static assumptions.

ACKNOWLEDGEMENTS

This work is supported by National Natural Science Foundation of China (nos. 61402366, 61272037, 61502248, 61472472, and 61272457), Natural Science Basic Research Plan in Shaanxi Province (no. 2015JQ6236), and Scientific Research Program funded by Shaanxi Provincial Education Department (no. 15JK1686). Also, Yinghui Zhang is supported by New Star Team of Xi'an University of Posts and Telecommunications, Jin Li is sponsored by a project funded by the Priority Academic Program Development of Jiangsu Higer Education Institutions and the Jiangsu Collaborative Innovation Center on Atmospheric Environment and Equipment Technology, and Qi Li is sponsored by NUPTSF (no. NY215008).

REFERENCES

1. Sahai A, Waters B. Fuzzy identity-based encryption. In *Advances in Cryptology-EUROCRYPT'05 Lecture Notes in Computer Science*, Vol. 3494, Cramer R (ed). Springer: Berlin-Heidelberg, 2005; 557–557.
2. Goyal V, Pandey O, Sahai A, Waters B. Attribute-based encryption for fine-grained access control of encrypted data. *Proceedings of the 13th ACM Conference on Computer and Communications Security*, CCS'06, ACM: New York, 2006; 89–98.
3. Chase M. Multi-authority attribute based encryption. In *Theory of Cryptography, Lecture Notes in Computer Science*, Vol. 4392, Vadhan S (ed). Springer: Berlin-Heidelberg, 2007; 515–534.
4. Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute-based encryption. *IEEE Symposium on Security and Privacy, SP'07*, IEEE: Oakland, 2007; 321–334.
5. Cheung L, Newport C. Provably secure ciphertext policy abe. *Proceedings of the 14th ACM Conference on Computer and Communications Security*, CCS'07, ACM: New York, 2007; 456–465.
6. Nishide T, Yoneyama K, Ohta K. Abe with partially hidden encryptor-specified access structure. In *Proceedings of Applied Cryptography and Network Security. ACNS'08, Lecture Notes in Computer Science*, Vol. 5037, Bellovin S, Gennaro R, Keromytis A, Yung M (eds). Springer: Berlin-Heidelberg, 2008; 111–129.
7. Zhang Y, Chen X, Li J, Wong D S, Li H. Anonymous attribute-based encryption supporting efficient decryption test. In *Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security*, ACM: New York, 2013; 511–516.
8. Yu S, Wang C, Ren K, Lou W. Attribute-based data sharing with attribute revocation. *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, ASIACCS'10, ACM: New York, 2010; 261–270.
9. Zhang Y, Chen X, Li J, Li H, Li F. FDR-ABE: attribute-based encryption with flexible and direct revocation. *The 5th International Conference on Intelligent Networking and Collaborative Systems (INCos)*, IEEE: Oakland, 2013; 38–45.
10. Zhang Y, Chen X, Li J, Li H, Li F. Attribute-based data sharing with flexible and direct revocation in cloud computing. *KSII Transactions on Internet & Information Systems* 2014; **8**(11): 4028–4049.
11. Shi Y, Zheng Q, Liu J, Han Z. Directly revocable key-policy attribute-based encryption with verifiable ciphertext delegation. *Information Sciences* 2015; **295**: 221–231.
12. Herranz J, Laguillaumie F, Ràfols C. Constant size ciphertexts in threshold attribute-based encryption. In *Public Key Cryptography-PKC 2010, Lecture Notes in Computer Science*, Vol. 6056, Nguyen P, Pointcheval D (eds). Springer: Berlin-Heidelberg, 2010; 19–34.
13. Takashima K. Expressive attribute-based encryption with constant-size ciphertexts from the decisional lin-

ear assumption. In *Security and Cryptography for Networks*. Springer: Berlin-Heidelberg, 2014; 298–317.
14. Zhang Y, Zheng D, Chen X, Li J, Li H. Computationally efficient ciphertext-policy attribute-based encryption with constant-size ciphertexts. In *Provable Security*. Springer: Berlin-Heidelberg, 2014; 259–273.
15. Liu Q, Wang G, Wu J. Time-based proxy re-encryption scheme for secure data sharing in a cloud environment. *Information Sciences* 2014; **258**: 355–370.
16. Zhang Y, Zheng D, Chen X, Li J, Li H. Efficient attribute-based data sharing in mobile clouds. *Pervasive and Mobile Computing* 2016; **28**: 135–149.
17. Li J, Ren K, Zhu B, Wan Z. Privacy-aware attribute-based encryption with user accountability. In *Proceedings of the International Information Security Conference. ISC'09, Lecture Notes in Computer Science*, Vol. 5735, Samarati P, Yung M, Martinelli F, Ardagna C (eds). Springer: Berlin-Heidelberg, 2009; 347–362.
18. Liu Z, Cao Z, Wong DS. Blackbox traceable cp-abe: how to catch people leaking their keys by selling decryption devices on ebay. *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, ACM: New York, 2013; 475–486.
19. Xhafa F, Feng J, Zhang Y, Chen X, Li J. Privacy-aware attribute-based phr sharing with user accountability in cloud computing. *The Journal of Supercomputing* 2015; **71**(5): 1607–1619.
20. Balu A, Kuppusamy K. An expressive and provably secure ciphertext-policy attribute-based encryption. *Information Sciences* 2014; **276**: 354–362.
21. Chase M, Chow SS. Improving privacy and security in multi-authority attribute-based encryption. *Proceedings of the 16th ACM Conference on Computer and Communications Security*, CCS '09, ACM: New York, 2009; 121–130.
22. Lewko A, Waters B. Decentralizing attribute-based encryption. In *Advances in cryptology–EUROCRYPT 2011, Lecture Notes in Computer Science,* Vol. 6632. Springer: Berlin-Heidelberg, 2011; 568–588.
23. Liu Z, Cao Z, Huang Q, Wong DS, Yuen TH. Fully secure multi-authority ciphertext-policy attribute-based encryption without random oracles. In *Computer security–ESORICS 2011*. Springer: Berlin-Heidelberg, 2011; 278–297.
24. Li J, Huang Q, Chen X, Chow SSM, Wong DS, Xie D. Multi-authority ciphertext-policy attribute-based encryption with accountability. *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, ASIACCS'11, ACM: New York, 2011; 386–390.
25. Li Q, Ma J, Li R, Xiong J, Liu X. Large universe decentralized key-policy attribute-based encryption. *Security and Communication Networks* 2015; **8** (3): 501–509.
26. Li Q, Ma J, Li R, Xiong J, Liu X. Provably secure unbounded multi-authority ciphertext-policy attribute-based encryption. *Security and Communication Networks* 2015; **8**(18): 4098–4109.
27. Datta P, Dutta R, Mukhopadhyay S. Fully secure, online/offline predicate and attribute-based encryption. In *Information Security Practice and Experience*. Springer: Berlin-Heidelberg, 2015; 331–345.
28. Rouselakis Y, Waters B. Practical constructions and new proof methods for large universe attribute-based encryption. *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, ACM: New York, 2013; 463–474.
29. Green M, Hohenberger S, Waters B. Outsourcing the decryption of abe ciphertexts. *Proceedings of the 20th USENIX Conference on Security*, SEC'11, USENIX Association: Berkeley, CA, USA, 2011; 34–34.
30. Li J, Huang X, Li J, Chen X, Xiang Y. Securely outsourcing attribute-based encryption with checkability. *IEEE Transactions on Parallel and Distributed Systems* 2014; **25**(8): 2201–2210.
31. Zhou Z, Huang D. Efficient and secure data storage operations for mobile cloud computing. *Proceedings of the 8th International Conference on Network and Service Management*, ACM: New York, 2012; 37–45.
32. Li J, Jia C, Li J, Chen X. Outsourcing encryption of attribute-based encryption with mapreduce. *The 14-th International Conference on Information and Communications Security*, Springer: Berlin Heidelberg, 2012; 191–201.
33. Li J, Chen X, Li J, Jia C, Ma J, Lou W. Fine-grained access control system based on outsourced attribute-based encryption. In *Computer Security – ESORICS 2013, Lecture Notes in Computer Science*, Vol. 8134, Crampton J, Jajodia S, Mayes K (eds). Springer: Berlin Heidelberg, 2013; 592–609.
34. Chen X, Li J, Ma J, Tang Q, Lou W. New algorithms for secure outsourcing of modular exponentiations. In *Computer Security–ESORICS 2012*. Springer: Berlin-Heidelberg, 2012; 541–556.
35. Chen X, Li J, Ma J, Tang Q, Lou W. New algorithms for secure outsourcing of modular exponentiations. *IEEE Transactions on Parallel and Distributed Systems* 2014; **25**(9): 2386–2396.
36. Hohenberger S, Waters B. Online/offline attribute-based encryption. In *Public-Key Cryptography–PKC 2014*. Springer: Berlin-Heidelberg, 2014; 293–310.

37. Even S, Goldreich O, Micali S. On-line/off-line digital signatures. *Journal of Cryptology* 1996; **9**(1): 35–67.
38. Shamir A, Tauman Y. Improved Online/offline signature schemes. In *Advances in Cryptology-CRYPTO 2001*. Springer: Berlin-Heidelberg, 2001; 355–367.
39. Zhang Y, Chen X, Li J, Li H. Generic construction for secure and efficient handoff authentication schemes in eap-based wireless networks. *Computer Networks* 2014; **75**: 192–211.
40. Chen X, Zhang F, Susilo W, Mu Y. Efficient generic on-line/off-line signatures without key exposure. In *Applied Cryptography and Network Security*. Springer: Berlin-Heidelberg, 2007; 18–30.
41. Chen X, Zhang F, Tian H, Wei B, Susilo W, Mu Y, Lee H, Kim K. Efficient generic on-line/off-line (threshold) signatures without key exposure. *Information Sciences* 2008; **178**(21): 4192–4203.
42. Guo F, Mu Y, Chen Z. Identity-based online/offline encryption. In *Financial Cryptography and Data Security*. Springer: Berlin-Heidelberg, 2008; 247–261.
43. Zhang Y, Chen X, Li H, Cao J. Identity-based construction for secure and efficient handoff authentication schemes in wireless networks. *Security and Communication Networks* 2012; **5**(10): 1121–1130.
44. Liu JK, Zhou J. An identity-based online/offline encryption scheme. In *Applied Cryptography and Network Security*. Springer: Berlin-Heidelberg, 2009; 156–167.
45. Lai J, Mu Y, Guo F, Susilo W. Improved identity-based online/offline encryption. In *Information Security and Privacy*. Springer: Berlin-Heidelberg, 2015; 160–173.
46. Beimel A. Secure schemes for secret sharing and key distribution. *PhD Thesis*, Technion-Israel Institute of Technology Faculty of Computer Science, 1996.