The Journal of Systems & Software 188 (2022) 111271

Contents lists available at ScienceDirect

The Journal of Systems & Software

journal homepage: www.elsevier.com/locate/jss

Policy-driven Data Sharing over Attribute-Based Encryption supporting Dual Membership☆

Hai Lu [a], Ruyun Yu [b], Yan Zhu [a,*], Xiao He [a], Kaitai Liang [c], William Cheng-Chung Chu [d,*]

[a] School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, 10083, China
[b] China Electronics Technology Research Institute of Cyberspace Security CO. LTD., Beijing, 10085, China
[c] Cybersecurity Group at Delft University of Technology, Van Mourik Broekmanweg 6, 2628 XE Delft, Netherlands
[d] Department of Computer Science, Tunghai University, Taichung, 40704, Taiwan

ARTICLE INFO

Article history:
Received 20 September 2021
Received in revised form 12 January 2022
Accepted 9 February 2022
Available online 17 February 2022

Keywords:
Dual Membership
Secure Decision of Membership
Attribute-Based Encryption
Private data sharing

ABSTRACT

Attribute-Based Encryption (ABE) plays an important role in current secure data sharing through fine-grained customizable policies. However, the existing ABE schemes only support simple predicates, $=$ and $\neq$, but cannot express a more general membership predicates, $\in$ and $\notin$, in policies. The low expressivity of ABE will enlarge the ciphertext storage and reduce the communication efficiency. To overcome this problem, we propose an ABE supporting Dual Membership (DM-ABE). The core problem for implementing this scheme is how to use cryptographic methods to decide the membership between the verified element and the given set. In order to solve this problem, we design a cryptographic algorithm, called Secure Decision of Membership (SDM), based on aggregation functions. In this algorithm, any set can be aggregated into one cryptographic element, and the verified element and the given set can be converted into another cryptographic element in decision process. The membership between them can be decided by the above two cryptographic elements. Furthermore, we construct the DM-ABE by using SDM. Because of the good expressivity of our DM-ABE, we further propose a novel cryptographic data sharing framework by integrating DM-ABE and attribute-based access control to provide fine-grained access control and security protection for private data. In the security proof of DM-ABE, we prove that the DM-ABE satisfies the semantic security against chosen-plaintext attacks under the DBDHE assumption in the standard model through a unified way, considering both two encryption methods for $\in$ and $\notin$ at the same time. Finally, we analyze our scheme in terms of time and space complexity, and compare it with some existing schemes. The results show that our DM-ABE has a better expressive ability on the boolean logic of general membership predicates, $\in$ and $\notin$.

1. Introduction

Thanks to the rapid development of information technologies, such as Internet of things (Xu et al., 2020) and cloud computing (Fan, 2021a), large-scale data sharing has become more and more widespread in the recent decade. For example, many countries have seen a steep rise in the amount of health data being generated. These data come not only from professional health systems (MRI scanners, pathology slides, DNA tests, etc.), but also from wearable devices. With up-to-date patient data at their fingertips, accurate and efficient health service can be provided to the fully informed patients and even save their lives. However, private data sharing between different organizations and users comes the risk of privacy leakage and unauthorized access. For this problem, many countries and regions established strict laws to provide protection for data sharing, such as the EU General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA). However, only around 10% of the world's population has personal information covered by the GDPR or similar laws at the moment. How to use technical method to protect data privacy so that more people can enjoy the benefits from data sharing has become a popular challenge.

Attribute-Based Encryption (ABE), as a mature public-key cryptography technology, especially Ciphertext-Policy ABE (CP-ABE), is considered by many experts as an important technology to face the above challenge (Zhang et al., 2020a). The reason is that CP-ABE uses attributes as the minimum authorization units to describe subjects and objects (i.e. users and data), and encrypts private data according to access policies. The policy refers to the boolean statement represented by a formula involving logic operators ("AND($\wedge$)" and "OR($\vee$)") over a set of attribute rules, e.g., $Depart \in \{Surgery, Radiology\} \wedge Stuff \notin \{Patient, Nurse\}$. A

☆ Editor: W. Eric Wong.

* Corresponding authors.
E-mail addresses: zhuyan@ustb.edu.cn (Y. Zhu), cchu@thu.edu.tw (W.C.-C. Chu).

https://doi.org/10.1016/j.jss.2022.111271
0164-1212/

Table 1
Example policy expressed in different logics.

Type	Expression	Number of predicates	Number of logic operators	Expressivity
DML	$Depart \in \{Surgery, Radiology\} \wedge Stuff \notin \{Patient, Nurse\}$	2	1	High
EL	$(Depart = Surgery \vee Depart = Radiology) \wedge (Stuff = ChiefPhysician \vee Stuff = Anesthetist \vee Stuff = Pharmacist \vee \cdots)$	$\geqslant 5$	$\geqslant 4$	Low
EL and NEL	$(Depart = Surgery \vee Depart = Radiology) \wedge (Stuff \neq Patient \wedge Stuff \neq Nurse)$	4	3	Medium

user has the right to decrypt the encrypted data if and only if his attribute set corresponding to his private key satisfies the policy. Therefore, CP-ABE can protect data for the large-scale data sharing, since any subject satisfying the policy can decrypt private data.

Currently, various CP-ABE schemes have been proposed to meet different requirements, including outsourcing computing (Lai et al., 2013; Ma et al., 2017), fast decryption (Malluhi et al., 2017), multi-authority (Li et al., 2011; Chow, 2016), traceability (Zhang et al., 2020b), etc. However, the existing CP-ABE schemes only support simple predicates, mainly including equivalence decision ($=$) and non-equivalence decision ($\neq$), which correspond to Equivalence Logic (EL) and Non-Equivalence Logic (NEL), respectively. They are just extreme cases of membership logic ($\in$ and $\notin$), which is called Dual Membership Logic (DML). The DML can represent two opposite memberships, Positive Membership (PM) $\in$ and Negative Membership (NM) $\notin$. However, the existing CP-ABE schemes cannot efficiently express the DML-type predicates.

We take a policy $Depart \in \{Surgery, Radiology\} \wedge Stuff \notin \{Patient, Nurse\}$ as an example, and Table 1 presents the policy expressed in different logics. If a CP-ABE only supports EL, it will express this policy into $(Depart = Surgery \vee Depart = Radiology) \wedge (Stuff = ChiefPhysician \vee Stuff = Anesthetist \vee Stuff = Pharmacist \vee \cdots)$. However, if this CP-ABE can support both EL and NEL, it can express this policy into $(Depart = Surgery \vee Depart = Radiology) \wedge (Stuff \neq Patient \wedge Stuff \neq Nurse)$. Clearly, the original policy only involves 2 predicates and 1 AND operators, while the policy expressed by the CP-ABE with EL and NEL involves 4 predicates and 3 AND/OR operators, because the DML-type predicates must be divided into several simple predicates for expression.

The above comparison indicates that the existing CP-ABE schemes have a low expressivity for the DML-type predicates because the DML-type predicates must be divided into several simple predicates for EL or NEL. Considering that each of predicates will be converted into a subciphertext in the encryption process of CP-ABE, the low expressivity will further increase the cost of ciphertext storage and computation, especially for the complex policies involved several DML-type predicates.

1.1. Motivation and approach

To overcome the low expressivity for DML-type predicates in the existing CP-ABE schemes, we construct a new CP-ABE scheme supporting Dual Membership, called DM-ABE scheme. The challenge in achieving this goal is how to use a cryptographic method to securely decide the membership between the verified element and the given set. For this challenge, we design a cryptographic algorithm, called Secure Decision of Membership (SDM), to securely make decision for dual memberships, i.e. PM and NM. The core part of the algorithm is the aggregation function which can implement compact cryptographic representation of sets. In this algorithm, any element u or set S will be converted into an element of the cryptographic space, where S is firstly encoded into a binary code, and then aggregated into a cryptographic element $EAgg(S)$ through an aggregation function $EAgg()$. In order to verify whether the element u belongs to the set S, another aggregation function $DAgg()$ is constructed to aggregate u and S into a cryptographic element $DAgg(u, S)$. The membership between u and S can be decided according to $EAgg(S)$ and $DAgg(u, S)$.

Then, on the basis of the SDM algorithm, the DM-ABE scheme will be constructed so as to support the expression of dual membership. In this scheme, our approach is to convert the decision problem of SDM into a computation problem for a specified value. If the user's attribute satisfies the predicate, he can reconstruct the correct value for decryption; otherwise, he only obtains a random value. Considering that the two different encryption methods for dual memberships are involving in DM-ABE, we intend to prove the security of DM-ABE in a complete proof rather than in two parts.

1.2. Related work

Since Sahai and Waters (2004) proposed a prototype of ABE, where each user's identity is described by a set of attributes, various ABE schemes have been proposed. The existing ABE schemes can be divided into three types: Key-Policy ABE (KP-ABE) (Goyal et al., 2006; Kim et al., 2017), CP-ABE (Bethencourt et al., 2007; Waters, 2011) and Dual-Policy ABE (DP-ABE) (Attrapadung and Imai, 2009). This paper mainly focuses on CP-ABE because data owners can specify scope of authorized users in CP-ABE scheme.

Recently, CP-ABE has become a mature technology, in terms of outsourcing computing (Li et al., 2020; Ning et al., 2018a; Zhong et al., 2021), fast decryption (Agrawal and Chase, 2017; Tsuchida et al., 2018), traceability (Li et al., 2009; Ning et al., 2018b), multi-authority (Jiang et al., 2016; Yu et al., 2017), security proof (Ambrona et al., 2017; Lin and Luo, 2020), etc. For example, Li et al. (2020) proposed an outsourcing CP-ABE scheme in which both authorized users and unauthorized users can verify the correctness of ciphertext transformation. Tsuchida et al. (2018) proposed a CP-ABE scheme supporting fast decryption and NEL. This scheme only needs constant pairing operations in decryption. Ning et al. (2018b) proposed a fully secure white-box traceable CP-ABE scheme to capture malicious users who leak their access credentials. Yu et al. (2017) proposed a multi-authority ABE scheme, which avoids key escrow and prevents the malicious sharing of secret key by traceability mechanism.

However, for the expressivity, the existing CP-ABE schemes only support simple logic, and cannot express DML-type predicates efficiently. For example, Bethencourt et al. (2007) and Goyal et al. (2008) only support equivalence decision $=$. To improve the expressivity, Waters (2011) proposed a CP-ABE scheme to support NEL. However, this scheme regards the negative version of a positive attribute as an independent attribute. It results in the doubling of the attribute number in system.

To overcome this problem, Ostrovsky et al. (2007) proposed a method to convert a monotonic access structure into a non-monotonic access structure. By using this method, Yamada et al. (2014) proposed a CP-ABE scheme supporting NEL. The above schemes do not involve the concept of attribute variables. When these schemes decide whether an attribute set Φ satisfies a NEL

predicate $X \neq a$, they compare a with all attributes in Φ. If a is not equal to all attributes in Φ, the predicate is satisfied.

Furthermore, Okamoto and Takashima (2010) and Okamoto and Takashima (2012) proposed another method to express NEL. In this method, each attribute is described as a tuple $(t, \vec{v})$, where t denotes an unique number of the attribute variable, and the vector $\vec{v}$ denotes the assignment of this attribute variable. When these schemes decide whether an attribute set Φ satisfies a NEL predicate $X \neq (t_1, \vec{v}_1)$, they firstly choose the attribute $(t_2, \vec{v}_2) \in \Phi$ where $t_2 = t_1$, and then calculate the inner product of $\vec{v}_1, \vec{v}_2$ to make decision. It indicates this predicate is satisfied, i.e., $(t_2 = t_1, \vec{v}_2) \neq (t_1, \vec{v}_1)$, if the inner product satisfies a specified condition. By using this method, Tomida et al. (2020) proposed a CP-ABE supporting NEL based on Agrawal and Chase (2017). Then, Tsuchida et al. (2018) proposed a CP-ABE scheme supporting NEL and fast decryption.

1.3. Contribution

- We propose the concept of dual memberships and the problem of secure decision of dual memberships. To solve this problem, we design two aggregation functions, i.e., *EAgg*() and *DAgg*(), to compact the given set to a cryptographic element. Based on them, the verified element and the given set can be converted into cryptographic elements in decision process of dual membership. Furthermore, the membership between these elements can be decided by utilizing the shift-and-cancellation methods on a specific basis vector. Thus, we verify the feasibility of cryptographic dual memberships.
- We propose a new ABE scheme supporting Dual Membership, called DM-ABE. In this scheme, the SDM decision is converted into a computation problem on a specified value, and the membership decision for a single attribute variable in SDM is extended into that of multiple variables. Thus, this scheme has a high expressivity for DML-type predicates, and supports various types of predicates, $=, \neq, \in$ and $\notin$. Finally, a Policy-Driven Data Sharing Architecture (PDDSA) is presented to provide secure issuing and acquiring on private data sharing by integrating DM-ABE with Attribute-Based Access Control (ABAC).

The security proof depends on Decisional Bilinear Diffie–Hellman Exponent (DBDHE) assumption rather than the randomness hypothesis of random oracle model. Furthermore, we consider both two different encryption methods for dual memberships at same time, and prove the security of DM-ABE in the complete proof rather than in two parts.

Organization: In the rest of this paper, Section 2 describes the SDM algorithm and DM-ABE scheme. The security analysis of the DM-ABE is presented in Section 3. In Section 4, we provide the performance analysis and comparison. In Section 5, we propose the application of DM-ABE, i.e., PDDSA. The paper concludes in Section 6.

2. DM-ABE

In order to improve the expressivity of DML-type predicates, we propose a concrete construction of DM-ABE based on the SDM algorithm in this section. In this section, we use the bilinear map group system $\mathbb{S} = (p, \mathbb{G}, \mathbb{G}_T, e(\cdot,\cdot))$ to implement our SDM algorithm and DM-ABE scheme, where $\mathbb{G}, \mathbb{G}_T$ are two cyclic groups of prime order p, bilinear mapping $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a mapping function which satisfies $e(g^a, h^b) = e(g, h)^{ab}$ for $\forall g, h \in \mathbb{G}$ and $\exists a, b \in \mathbb{Z}_p^*$.

2.1. Construction of SDM

The algorithm, secure decision of membership, mainly focuses on the cryptographic representation of subsets in a set with fixed number of elements. For the set $U = \{e_1, e_2, \ldots, e_{n-1}\}$, a party with an element e_i can prove the membership (PM $\in$ or NM $\notin$) between e_i and a subset $S \subseteq U$ to a party with S by using SDM. The definition of SDM is shown as follows.

Definition 1 (*SDM*)**.** A Probabilistic Polynomial-Time (PPT) algorithm $P(e_i, S)$ is called a SDM algorithm, if for any e_i, the SDM can make decision for both PM and NM with the probability $1 - \epsilon$, where ϵ is negligible, and SDM satisfy the following inequality,

$$\Pr\left[P(e_i, S) = \begin{cases} 1, & e_i \in S \\ -1, & e_i \notin S, e_i \in U \\ 0, & e_i \notin U \end{cases}\right] \geq 1 - \epsilon. \tag{1}$$

As shown in Eq. (1), this algorithm shall make decision for both PM ($\in$) and NM ($\notin$). For two cases, i.e., $e_i \in S$ and $e_i \notin S \wedge e_i \in U$, this algorithm can output 1 and -1 with overwhelming probability, respectively. Moreover, a new case $e_i \notin U$ is also added into Eq. (1). It indicates that a malicious party intends to forge an element $e_i \notin U$ to pass the verification of SDM algorithm, and this algorithm can distinguish the forged element efficiently.

To implement a practical construction of SDM, we use an aggregation function to generate the compact cryptographic representation of subsets. For the set $U = \{e_1, e_2, \ldots, e_{n-1}\}$, the aggregation function can compact any subset $S \subseteq U$ into a value with fixed size. The definition of the aggregation function is shown as follows:

Definition 2 (*Aggregation Function*)**.** Let $\mathcal{PK}$ be the public parameters in a group $\mathbb{G}$, $U = \{e_1, e_2, \ldots, e_{n-1}\}$ be the set of all attributes. The aggregation function $Aggregate : \mathcal{PK} \times 2^U \to \mathbb{G}$ is a deterministic polynomial time algorithm and satisfies

$$Aggregate(mpk, S) = R_S, \tag{2}$$

where, mpk is the public key in $\mathcal{PK}$, $S \subseteq U$ is a subset, R_S is a sufficiently random value of group that prevents random guess.

According to the above definition, we design the aggregation function *EAgg*() shown as follows. Let n be a positive integer and g be a generator in $\mathbb{G}$. For a random value $\mu \in_R \mathbb{Z}_p$, a parameter sequence $\{g, g^{\mu}, g^{\mu^2}, \ldots, g^{\mu^{n-1}}, g^{\mu^n}, g^{\mu^{n+1}}, \cdots, g^{\mu^{2n-1}}\}$ consisting of $2n$ elements is constructed. By removing g^{μ^n} from this sequence, we can obtain the main public key

$$mpk = \{g, g^{\mu}, g^{\mu^2}, \ldots, g^{\mu^{n-1}}, g^{\mu^{n+1}}, \ldots, g^{\mu^{2n-1}}\}, \tag{3}$$

where μ is a secret. For any subset $S \subseteq U$, we map its element $e_i \in S$ to $g^{\mu^{n-i}}$, and construct *Eagg*() to generate the cryptographic representation of S, as shown in Eq. (4).

$$EAgg(S) = \prod_{e_i \in S} g^{\mu^{n-i}} = g^{\sum_{e_i \in S} \mu^{n-i}}. \tag{4}$$

The above function *EAgg*() aggregates S into a cryptographic element, and the discrete logarithm problem guarantees the randomness of $EAgg(S)$. Specifically, we take a subset $S = \{e_2, e_3\} \subseteq U = \{e_1, e_2, e_3\}$ as an example (for $n = 4$). S is firstly encoded as 011 according to the index of S in the power set of U. Then, the polynomial representation $\mu + \mu^2$ of this subset is generated by 011 on the random variable μ, where $(\mu + \mu^2)$'s binary code (we call it μ-code) for μ is 110. Finally, we extend this polynomial into an element $g^{\mu+\mu^2}$ in $\mathbb{G}$.

When we decide whether the verified element e_i is in a given subset S, the μ-code of S can be shifted i bits to the right, i.e., shift the $(n - i)$th bit of the μ-code to the nth bit. It indicates that

Table 2
The binary encoding for all subsets of $\{e_1, e_2, e_3\}$ with the verified element e_2.

Subset	$(e_1e_2e_3)$	Polynomial representation	$(\mu^1\mu^2\mu^3)$	Shifted polynomial representation	$(\mu^1\mu^2\mu^3[\mu^4]\mu^5)$
{}	(000)	(0)	(000)	(0)	(00 000)
$\{e_3\}$	(001)	(μ)	(100)	(μ^3)	(00 100)
$\{e_2\}$	(010)	(μ^2)	(010)	$([\mu^4])$	(00 010)
$\{e_2, e_3\}$	(011)	$(\mu + \mu^2)$	(110)	$(\mu^3 + [\mu^4])$	(00 110)
$\{e_1\}$	(100)	(μ^3)	(001)	(μ^5)	(00 001)
$\{e_1, e_3\}$	(101)	$(\mu + \mu^3)$	(101)	$(\mu^3 + \mu^5)$	(00 101)
$\{e_1, e_2\}$	(110)	$(\mu + \mu^2)$	(011)	$([\mu^4] + \mu^5)$	(00 011)
$\{e_1, e_2, e_3\}$	(111)	$(\mu + \mu^2 + \mu^3)$	(111)	$(\mu^3 + [\mu^4] + \mu^5)$	(00 111)

$e_i \notin S$ if the nth bit is 0; otherwise, $e_i \in S$. This is called shift-and-cancellation method. For example, if the verified element is e_2, the mentioned μ-code can be shifted 2 bits to the right so as to obtain 00110. Clearly, the fourth bit of it is 1, therefore e_2 is in $\{e_2, e_3\}$. According to the above approach, we convert the problem of deciding membership into the 0/1 decision of the nth bit. Table 2 presents the binary encoding for all subsets of $\{e_1, e_2, e_3\}$, where the verified element is e_2.

Furthermore, we convert this 0/1 decision problem into a cryptographic problem of computing the nth element g^{μ^n}. In order to achieve this conversion, another aggregation function $DAgg()$ is designed to aggregate the verified element e_i and the given subset S, defined as Eq. (5).

$$DAgg(e_i, S) = \begin{cases} \prod\limits_{e_j \in S, e_j \neq e_i} g^{\mu^{n-j+i}} & e_i \in S, \\ \prod\limits_{e_j \in S} g^{\mu^{n-j+i}} & e_i \notin S. \end{cases} \tag{5}$$

Based on two aggregation functions, $EAgg()$ and $DAgg()$, the SDM algorithm can be constructed as follows:

1. Randomly choose $h \in_R \mathbb{G}$ and secret $r \in_R \mathbb{Z}_p^*$, then generate public parameters:

$$PK = (g, h, v = g^r, \{g_i = g^{\mu^i}\}_{i=1,i\neq n}^{2n}, \{h_i = h^{\mu^i}/g_n\}_{i=1}^{n-1}). \tag{6}$$

2. For the verified element e_i, generate its cryptographic representation $E_i = g_i^r$ according to the secret r.
3. For the set $S \subseteq U$, randomly choose $t \in \mathbb{Z}_p$, then compute $C_0 = g^t$, $w = v \cdot h$ and decision basis $W = e(g_{n-1}, g_1)^t = e(g_n, g)^t$. Finally, generate the cryptographic representation of set S according to PK as follows:

$$C_S = \begin{cases} (v \cdot EAgg(S))^t & e_i \in S, \\ (w/EAgg(S))^t & e_i \notin S. \end{cases} \tag{7}$$

4. The following equation can be used to verify the dual membership between e_i and S:

$$P(e_i, S) = \begin{cases} 1 & W = e(C_S, g_i)/e(E_i \cdot DAgg(e_i, S), C_0), \\ -1 & W = e(C_S, g_i)/e(E_i \cdot h_i/DAgg(e_i, S), C_0), \\ 0 & \text{Otherwise.} \end{cases} \tag{8}$$

Clearly, this algorithm satisfies Eq. (1). It means that this algorithm can efficiently decide the membership between the verified element e_i and the given set $S \subseteq U$, i.e. $e \in S$ and $e \notin S \wedge e \in U$. Moreover, it can distinguish the forged element $e_i \notin U$.

2.2. Definition of DM-ABE

In this subsection, we will present the definition of the ABE scheme supporting dual membership. Let $\mathbb{A} = \{A_1, A_2, \ldots, A_m\}$ be the attribute variable set, where A_i is an attribute variable, m is the number of attribute variables. Suppose that there are at most $(n-1)$ assignments for each attribute variable $A_i \in \mathbb{A}$. Let $U_i = \{e_{i1}, e_{i2}, \ldots, e_{i(n-1)}\}$ be the assignment set of the attribute variable A_i, where e_{ij} represents the jth assignment of A_i. Let Φ be the user's attribute set, e.g. $\Phi = \{A_1 \leftarrow e_{11}, A_2 \leftarrow e_{21}, A_3 \leftarrow e_{32}\}$. We use $\Pi(\Phi) = 1$ to denote that the Φ satisfies the policy Π. Our DM-ABE scheme consists of four algorithms shown as follows:

- **Setup**$(\mathbb{A}, \kappa)$**:** This algorithm takes an attribute variable set $\mathbb{A}$, and a security parameter κ as inputs, then outputs a bilinear map group system $\mathbb{S}$, a public key PK and a master secret key MK.
- **KeyGen**(MK, Φ, ID_k)**:** This algorithm takes the master secret key MK, a user's attribute set Φ and his/her unique identity ID_k as inputs, then outputs this user's private key $sk_\Phi^{(k)}$.
- **DMABE-Enc**(PK, Π)**:** This algorithm takes the public key PK and an access policy Π as inputs, then outputs a session key ek and its ciphertext C_Π.
- **DMABE-Dec**$(PK, sk_\Phi^{(k)}, C_\Pi)$**:** This algorithm takes the public key PK, the user's private key $sk_\Phi^{(k)}$ and the ciphertext C_Π as inputs. If the user's attribute set Φ satisfies the policy Π, i.e., $\Pi(\Phi) = 1$, the session key ek can be reconstructed from the ciphertext C_Π; otherwise, this algorithm outputs a invalid value $\perp$.

Correctness. For all possible public keys PK and master keys MK from $(PK, MK) \leftarrow$ **Setup**$(\mathbb{S}, \mathbb{A}, \kappa)$, any user's private key $sk_\Phi^{(k)}$ corresponding to his/her attribute set Φ can be generated from $sk_\Phi^{(k)} \leftarrow$ **KeyGen**(MK, Φ, ϕ_k). Given any policy Π, a session key ek and its valid ciphertext C_Π can be yielded from $(C_\Pi, ek) \leftarrow$ **DMABE** $-$ **Enc**(PK, Π). If there exists a valid attribute set $\Phi' \subseteq \Phi$ satisfying the access policy Π, i.e., $\Pi(\Phi') = 1$, the correct ek can be reconstructed from C_Π by **DMABE** $-$ **Dec**, that is,

$$\Pr[\textbf{DMABE} - \textbf{Dec}(PK, sk_\Phi^{(k)}, C_\Pi) = ek : \exists \Phi' \subseteq \Phi, \Pi(\Phi') = 1] = 1. \tag{9}$$

2.3. Construction of DM-ABE

In this subsection, we will describe the construction of DM-ABE from the SDM algorithm. In this construction, a cryptographic hash function $hash : \{0,1\}^* \rightarrow \mathbb{Z}_p^*$ is used to map each attribute variable A_i described as a binary string to a random element $s_i = hash(A_i)$ for $i \in [1, m]$. For clarity, we use S_i and R_i to denote the designated set and the revocation set, respectively, so that the PM-type and NM-type predicates can be represented as $A_i \in S_i$ and $A_i \notin R_i$, respectively. For each predicate in the policy, the DM-ABE utilizes the SDM to determine the membership between the user's attribute and the given set for authorization decision.

The construction of **Setup** is proposed in Algorithm 1, where, m is the number of attribute variables, and $(n-1)$ is the number of assignments for each attribute variable. In order to apply

SDM to our DM-ABE, we generate the related parameters in this algorithm. Similarly to the SDM, $\{g_j\}_{j=1,j\neq n}^{2n}$ and $\{h_j\}_{j=1}^{n-1}$ are generated for the aggregation functions in encryption and decryption. Moreover, the SDM only considers one attribute variable, while the DM-ABE considers m attribute variables. Therefore, v in the SDM is extended to $\{v_i\}_{i=1}^{m}$ for m attribute variables.

Algorithm 1 Setup

Input: The attribute set $\mathbb{A}$ and security parameter κ;
Output: Public key PK and master secret key MK;

1: generate $\mathbb{S} = (p, \mathbb{G}, \mathbb{G}_T, e(\cdot,\cdot))$ according to κ;
2: randomly pick elements $\alpha, \beta, \mu, \gamma \in_R \mathbb{Z}_p^*$, $h \in_R \mathbb{G}$ and a generator g of $\mathbb{G}$;
3: compute $\eta = g^{\alpha}$, $\xi = g^{\beta}$;
4: **for** $i \in [1, m]$ **do**
5: compute $v_i = g^{s_i\gamma}$, where $s_i = hash(A_i)$;
6: **for** $j \in [1, 2n] \wedge j \neq n$ **do**
7: compute $g_j = g^{\mu^j}$;
8: **for** $j \in [1, n-1]$ **do**
9: compute $h_j = h^{\mu^j}/g_n$;
10: set $V = e(g_n, g)$;
11: **return** $MK = (\alpha, \beta, \mu, \gamma)$, $PK := (g, h, \eta, \xi, \{v_i\}_{i=1}^{m}, \{g_j\}_{j=1,j\neq n}^{2n}, \{h_j\}_{j=1}^{n-1})$.

The Algorithm 2 shows the construction of **KeyGen**. Note that, the user's attribute subkey for $(A_i \leftarrow e_{ij})$ is not only related to the attribute e_{ij}, but also related to the attribute variable A_i and identity ID_k in this algorithm. Therefore, the sub-key is set in the form $d_{ij}^{(k)} = g_j^{s_i\gamma} \cdot g^{-\phi_k}$ rather than the form $E_j = g_j^r$ in the SDM.

Algorithm 2 KeyGen

Input: Master secret key MK, user's attribute set Φ and identity ID_k;
Output: User's private key $sk_{\Phi}^{(k)}$;

1: randomly pick $\phi_k \in_R \mathbb{Z}_p^*$ for ID_k;
2: **for** $\forall(A_i \leftarrow e_{ij}) \in \Phi$ **do**
3: compute its sub-key $d_{ij}^{(k)} = g_j^{s_i\gamma} \cdot g^{-\phi_k} = v_i^{\mu^j} \cdot g^{-\phi_k}$;
4: compute $d^{(k)} = g^{\frac{\alpha+\phi_k}{\beta}}$;
5: **return** $sk_{\Phi}^{(k)} = \{\{d_{ij}^{(k)}\}_{(A_i\leftarrow e_{ij})\in\Phi}, d^{(k)}\}$.

The Algorithm 3 gives the construction of **DMABE-Enc**. In this algorithm, Linear Secret Sharing Scheme (LSSS) (Li, 2013) is used to represent the access policy. The policy Π involving l predicates will be converted into (M, ρ) by LSSS to share a random secret t, where M is an $l \times b$ matrix, and ρ is a permutation function to map each row of M to a predicate. For each predicate in the policy, i.e., $A_i \in S_i$ or $A_i \notin R_i$, this algorithm utilizes $EAgg()$ of the SDM to aggregate S_i or R_i, and then generate the subciphertext (c_{i1}, c_{i2}), where, c_{i1} corresponds to C_0, c_{i2} corresponds to C_s in the SDM.

The Algorithm 4 proposes the construction of **DMABE-Dec**. If the user's attribute set Φ satisfies the policy Π, there exist an authorized set F and a index set $I = \{i : \rho(i) \in F\}$. Then, the reconstruction vector $\{\omega_i \in \mathbb{Z}_p\}_{i\in I}$ used for decryption can be computed according to M and I, such that $\Sigma_{i\in I}\omega_i\lambda_i = t$. Then, for each subsiphertext (c_{i1}, c_{i2}), this algorithm uses Eq. (11), which is similar to Eq. (8) in the SDM, to compute $c_i = e(g_n g^{\phi_k}, g)^{\lambda_i}$, where $DAgg()$ is utilized to aggregate the user's attribute and the given set. Finally, ek can reconstructed by $\{\omega_i\}_{i\in I}$ and $\{c_i\}_{i\in I}$.

Correctness. Next, we will discuss the correctness of our DM-ABE scheme. There are two cases shown as follows:

- **Case 1:** If the predicate is $A_i \in S_i$, and the attribute $A_i \leftarrow e_{ij}$ corresponding to the $d_{ij}^{(k)}$ satisfies this predicate, c_i can be computed as follows:

$$\begin{aligned} c_i &= \frac{e(c_{i2}, g_j)}{e(d_{ij}^{(k)} \cdot DAgg(e_{ij}, S_i), c_{i1})} = \frac{e((v_i \cdot EAgg(S_i))^{\lambda_i}, g_j)}{e(g_j^{s_i\gamma-\phi_k} \cdot DAgg(e_{ij}, S_i), g^{\lambda_i})} \\ &= \frac{e\left((g^{s_i\gamma} \cdot \prod_{e_{ik}\in S_i} g_{n-k})^{\lambda_i}, g_j\right) \cdot e(g^{\phi_k}, g^{\lambda_i})}{e\left(g_j^{s_i\gamma} \cdot \prod_{e_{ik}\in S_i, k\neq j} g_{n-k+j}, g^{\lambda_i}\right)} \\ &= \frac{e\left(g_j^{s_i\gamma} \cdot \prod_{e_{ik}\in S_i, k\neq j} g_{n-k+j}, g^{\lambda_i}\right) \cdot e(g_n g^{\phi_k}, g^{\lambda_i})}{e\left(g_j^{s_i\gamma} \cdot \prod_{e_{ik}\in S_i, k\neq j} g_{n-k+j}, g^{\lambda_i}\right)} \\ &= e(g_n g^{\phi_k}, g)^{\lambda_i}. \end{aligned} \tag{12}$$

Algorithm 3 DMABE-Enc

Input: Public key PK and access policy Π;
Output: Session key ek and ciphertext C_{Π};

1: convert Π into (M, ρ) by LSSS;
2: randomly pick a vector $v = (t, r_2, \cdots, r_b) \in \mathbb{Z}_p^b$ to share the secret t;
3: **for** $i \in [1, l]$ **do**
4: compute $\lambda_i = M_i \cdot v$;
5: extract the predicate $A_i \in S_i$ or $A_i \notin R_i$ from the ith literal of Π;
6: **if** the predicate is $A_i \notin R_i$ **then**
7: compute $w_i = v_i \cdot h$;
8: compute (c_{i1}, c_{i2}) according to S_i or R_i as follows:

$$(c_{i1}, c_{i2}) = \begin{cases} (g^{\lambda_i}, (v_i \cdot EAgg(S_i)) & A_i \in S_i; \\ (g^{\lambda_i}, (w_i/EAgg(R_i)) & A_i \notin R_i; \end{cases} \tag{10}$$

9: compute $c' = g^t$ and $c_0 = g^{\beta\cdot t} = \xi^t$;
10: **return** $ek = \frac{e(g^{\alpha}, g^t)}{V^t}$ and $C_{\Pi} = (\Pi, (M, \rho), c', c_0, \{c_{i1}, c_{i2}\}_{i=1}^{l})$.

Algorithm 4 DMABE-Dec

Input: Public key PK, user's private key $sk_{\Phi}^{(k)}$ and ciphertext C_{Π};
Output: Session key ek;

1: **if** the user's attribute set Φ satisfies the policy Π **then**
2: set F as the authorized set according to the user's private key $sk_{\Phi}^{(k)}$;
3: set $I = \{i : \rho(i) \in F\}$;
4: compute the reconstruction vector $\{\omega_i \in \mathbb{Z}_p\}_{i\in I}$ according to M and I;
5: **for** $\forall i \in I$ **do**
6: extract the S_i or R_i from the ith predicate, $A_i \in S_i$ or $A_i \notin R_i$, from Π;
7: compute c_i according to the sub-key $d_{ij}^{(k)}$ as follows:

$$c_i = \begin{cases} \dfrac{e(c_{i2}, g_j)}{e(d_{ij}^{(k)} \cdot DAgg(e_{ij}, S_i), c_{i1})} & A_i \in S_i \wedge A_i \leftarrow e_{ij}; \\ \dfrac{e(c_{i2}, g_j)}{e(d_{ij}^{(k)} \cdot h_j/DAgg(e_{ij}, R_i), c_{i1})} & A_i \notin R_i \wedge A_i \leftarrow e_{ij}; \end{cases} \tag{11}$$

8: compute $T = \prod_{i\in I} c_i^{\omega_i} = e(g_n g^{\phi_k}, g)^t$;
9: compute $ek = \frac{e(d^{(k)}, c_0)}{T} = \frac{e(g^{\alpha}, g^t)}{V^t}$;
10: **return** ek.
11: **else**
12: **return** $\perp$.

- **Case 2:** If the predicate is $A_i \notin R_i$, and the attribute $A_i \leftarrow e_{ij}$ corresponding to the $d_{ij}^{(k)}$ satisfies this predicate, c_i can be computed as follows:

$$\begin{aligned} c_i &= \frac{e(c_{i2}, g_j)}{e(d_{ij}^{(k)} \cdot h_j / DAgg(e_{ij}, R_i), c_{i1})} \\ &= \frac{e((v_i \cdot h/EAgg(R_i))^{\lambda_i}, g_j)}{e(g_j^{s_i\gamma-\phi_k} \cdot h_j/DAgg(e_{ij}, R_i), g^{\lambda_i})} \\ &= \frac{e\left((g^{s_i\gamma} \cdot h/\prod_{e_{ik}\in R_i} g_{n-k})^{\lambda_i}, g_j\right) \cdot e(g^{\phi_k}, g^{\lambda_i})}{e\left(g_j^{s_i\gamma} \cdot h^{\mu^j}/(g_n \cdot \prod_{e_{ik}\in R_i} g_{n-k+j}), g^{\lambda_i}\right)} \\ &= \frac{e\left(g^{s_i\gamma} \cdot h/\prod_{e_{ik}\in R_i} g_{n-k}, g_j^{\lambda_i}\right) \cdot e(g_n g^{\phi_k}, g^{\lambda_i})}{e\left(g^{s_i\gamma} \cdot h/\prod_{e_{ik}\in R_i} g_{n-k}, g_j^{\lambda_i}\right)} \\ &= e(g_n g^{\phi_k}, g)^{\lambda_i}. \end{aligned} \tag{13}$$

If the user's attribute set Φ satisfies the access policy Π, the reconstruction vector $\{\omega_i \in \mathbb{Z}_p\}_{i\in I}$ can be computed, and the intermediate value T can be obtained as follows,

$$T = \prod_{i\in I} c_i^{\omega_i} = \prod_{i\in I} e(g_n g^{\phi_k}, g)^{\lambda_i\omega_i} = e(g_n g^{\phi_k}, g)^t. \tag{14}$$

Finally, the session key ek' can be computed as follows:

$$ek' = \frac{e(d^{(k)}, c_0)}{T} = \frac{e(g^{\frac{\alpha+\phi_k}{\beta}}, g^{\beta t})}{e(g_n g^{\phi_k}, g)^t} = \frac{e(g^\alpha, g^t)}{e(g_n, g^t)} = ek. \tag{15}$$

3. Security analysis

In this section, we firstly describe a game to define the security of DM-ABE scheme. Then, the DBDHE assumption is reduced to prove that our scheme satisfies semantic security.

3.1. Security requirements

We require that the DM-ABE scheme satisfies semantic security under the chosen-plaintext attacks (i.e., IND-CPA). Specifically, given two messages, m_0 and m_1, their corresponding ciphertexts, C_0 and C_1, are indistinguishable. DM-ABE is semantically secure against IND-CPA for a given challenge policy, if the advantage of any PPT adversary is negligible in the following game.

- **Setup**. $\mathcal{B}$ runs **Setup** algorithm to generate (PK, MK), then $\mathcal{B}$ sends the public key PK to $\mathcal{A}$.
- **Learning**. $\mathcal{A}$ makes private key queries for some attribute sets Φ, which do not satisfy the challenged policy Π^*, i.e., $\Pi^*(\Phi) = 0$. $\mathcal{B}$ runs **KeyGen** algorithm to generate the private key $sk_\Phi^{(\phi)}$, then $\mathcal{B}$ sends it to $\mathcal{A}$.
- **Challenge**. $\mathcal{A}$ randomly selects two messages, m_0 and m_1, and sends them to $\mathcal{B}$. $\mathcal{B}$ runs **DMABE**$-$ **Enc**$(PK, \Pi^*) \rightarrow (C_{\Pi^*}, ek)$ under the challenged policy Π^*. Then he flips a random coin $\sigma \in \{0, 1\}$, and computes the ciphertext C_σ of m_σ. At last, $\mathcal{B}$ sends the ciphertext to $\mathcal{A}$.
- **Guess**. $\mathcal{A}$ outputs a guess σ' of σ, and he wins the game if $\sigma' = \sigma$.

The advantage of $\mathcal{A}$ in this game is defined as $Adv_{\mathrm{DM-ABE}}^{\mathrm{IND}}(\mathcal{A}) \stackrel{\mathrm{def}}{=} \left|\Pr[\sigma' = \sigma] - \frac{1}{2}\right|$. The security definition of the DM-ABE scheme is shown as follow:

Definition 3 (*Semantic Security*)**.** We say the DM-ABE scheme is semantically secure under the IND-CPA game, if the advantage $Adv_{\mathrm{DM-ABE}}^{\mathrm{IND-CPA}}(\mathcal{A})$ is negligible, i.e. $Adv_{\mathrm{DM-ABE}}^{\mathrm{IND-CPA}}(\mathcal{A}) < \epsilon(\kappa)$ in the security parameter κ for all PPT adversaries $\mathcal{A}$.

In the security proof, we prove our DM-ABE scheme satisfies IND-CPA security under the DBDHE assumption. The definition of the DBDHE problem and the DBDHE assumption are shown as follows:

Definition 4 (*DBDHE Problem Xie and Ren, 2014*)**.** Given a $(2n+1)$-tuple $(g, g^t, \{g^{\mu^i}\}_{i=1,i\neq n}^{2n}) \in \mathbb{G}^{2n+1}$ and a random element $W \leftarrow_R \mathbb{G}_T$ as input, output 1 if $W = e(g^{\mu^n}, g)^t$ and 0 otherwise.

We define the advantage of algorithm $\mathcal{B}$ to solve the DBDHE problem as follows:

$$Adv_{DBDHE}^{IND}(\mathcal{B}) = \left| \begin{matrix} \Pr[\mathcal{B}(\mathcal{R}, e(G^{\mu^n}, G)^t) = 1 : G \stackrel{R}{\leftarrow} \mathbb{G}, \mu, t \stackrel{R}{\leftarrow} \mathbb{Z}_p^*] \\ -\Pr[\mathcal{B}(\mathcal{R}, W) = 1 : G \stackrel{R}{\leftarrow} \mathbb{G}, \mu, t \stackrel{R}{\leftarrow} \mathbb{Z}_p^*, W \stackrel{R}{\leftarrow} \mathbb{G}_T] \end{matrix} \right|. \tag{16}$$

where $\mathcal{R} = (G, G^t, \{G^{\mu^i}\}_{i=1,i\neq n}^{2n})$.

Definition 5 (*(ϵ, n)-DBDHE Assumption Xie and Ren, 2014*)**.** We say that the DBDHE assumption is (ϵ, n)-secure in $\mathbb{S}$, if for all PPT algorithms $\mathcal{B}$, the advantage of solving the DBDHE problem is at most ϵ, i.e., $Adv_{\mathrm{DBDHE}}^{\mathrm{IND}}(\mathcal{B}) < \epsilon$.

3.2. Security proof

In this paper, we prove that our DM-ABE scheme is IND-CPA under the assumption that the DBDHE problem is hard. The DM-ABE utilizes two different encryption methods for dual predicates, $\in$ and $\notin$, however, we prove the security of DM-ABE in the complete proof rather than two parts. During describing the game, the validation of simulated private keys is additionally provided in the learning stage. More preciously, we have the following theorem.

Theorem 1 (*Semantic Security of DM-ABE*)**.** *The DM-ABE scheme is (ϵ, n)-semantically secure against chosen-plaintext attack under (ϵ, n)-DBDHE assumption in $\mathbb{S}$, and advantage of $\mathcal{A}$ is $Adv_{\mathrm{DM-ABE}}^{\mathrm{IND-CPA}}(\mathcal{A}) < \epsilon$.*

Proof. Suppose there exists an adversary $\mathcal{A}$ that can break our DM-ABE scheme under a non-negligible advantage, that is, $Adv_{\mathrm{DM-ABE}}^{\mathrm{IND-CPA}}(\mathcal{A}) \geq \epsilon$. Our objective is to build a PPT algorithm $\mathcal{B}$ to solve the DBDHE problem. We utilize the following game to depict the construction of $\mathcal{B}$.

- **Setup.** The simulator $\mathcal{B}$ runs **Setup** algorithm, and sends public parameters PK to the adversary $\mathcal{A}$. This process is divided into the following six steps:
  - Step 1. Set $g = G$, and set $G_j = G^{\mu^j}$ for $j \in [1, n-1]$, s.t., $g_j = g^{\mu^j} = G^{\mu^j}$, where G is a generator of $\mathbb{G}$.
  - Step 2. Randomly select $\alpha, \beta \in_R \mathbb{Z}_p^*$, further compute $\eta = g^\alpha = G^\alpha$ and $\xi = g^\beta = G^\beta$;
  - Step 3. Compute $s_i = hash(A_i)$ for $i = [1, m]$;
  - Step 4. Randomly select $\zeta \in_R \mathbb{Z}_p^*$, and set $\gamma = \zeta - (\sum_{e_{ik}\in S_i} \mu^{n-k})/s_i$ (γ is unknown, because μ is unknown). Then compute v_i for $i \in [1, m]$ as follows:

$$v_i = g^{s_i\gamma} = G^{s_i(\zeta - \frac{\sum_{e_{ik}\in S_i} \mu^{n-k}}{s_i})} = G^{s_i\zeta} / \prod_{e_{ik}\in S_i} G_{n-k}, \tag{17}$$

    where, v_i can be computed, although γ is unknown. The reason is that G, s_i and ξ are known, and

$\prod_{e_{ik}\in S_i} G_{n-k}$ can be computed by $\{G_j\}_{j=1}^{n-1}$ in DBDHE instance.

- Step 5. Randomly select $\delta \in_R \mathbb{Z}_p^*$ and compute $h = G^{\delta} \cdot \prod_{e_{ik}\in U_i} G_{n-k}$. Note that, h can be computed since G, δ and $\{G_j\}_{j=1}^{n-1}$ are known.
- Step 6. For $j \in [1, n-1]$, h_j can be computed shown as follows:

$$h_j = \frac{h^{\mu^j}}{g_n} = \frac{G_j^{\delta} \cdot \prod_{e_{ik}\in U_i} G_{n-k+j}}{G_n} = G_j^{\delta} \cdot \prod_{e_{ik}\in U_i, k\neq j} G_{n-k+j}. \quad (18)$$

At last, $\mathcal{B}$ sends $PK = (G, \eta, \xi, h, \{v_i\}_{i=1}^{m}, \{g_j\}_{j=1,j\neq n}^{2n}, \{h_j\}_{j=1}^{n-1})$ to $\mathcal{A}$.

- **Learning.** $\mathcal{A}$ makes secret key queries for any attribute set Φ^* which satisfies $\Pi^*(\Phi^*) = 0$. For the secret key query for each attribute $A_i \leftarrow e_{ij} \in \Phi^*$, $\mathcal{B}$ randomly chooses $\phi \in_R \mathbb{Z}_p^*$ according to the user's *ID*. Then, for each attribute $A_i \leftarrow e_{ij}$ of this user, $d_{ij}^{(\phi)}$ can be computed as follows:

$$d_{ij}^{(\phi)} = g_j^{s_i\gamma} \cdot g^{-\phi} = G_j^{s_i\zeta} / \prod_{e_{ik}\in S_i} G_{n-k+j} \cdot G^{-\phi}, \quad (19)$$

Note that, if $e_{ij} \in S_i$, i.e., $k = j$, the key $d_{ij}^{(\phi)}$ of e_{ij} cannot be generated because G_n is unknown. Therefore, $d_{ij}^{(\phi)}$ is valid only if $e_{ij} \notin S_i$. Then $\mathcal{B}$ computes $d^{(\phi)} = g^{\frac{\alpha+\phi}{\beta}}$. Finally, $\mathcal{B}$ sends $sk_{\Phi^*}^{(\phi)} = \{\{d_{ij}^{(\phi)}\}_{(A_i\leftarrow e_{ij})\in\Phi^*}, d^{(\phi)}\}$ to $\mathcal{A}$.

- **Challenge.** $\mathcal{A}$ randomly chooses two messages, m_0 and m_1, and sends them to $\mathcal{B}$. $\mathcal{B}$ runs **DMABE-Enc** algorithm to generate the ciphertext C_{Π^*} under Π^*. This process is described as follows:
  - Step 1. Convert the policy Π^* into (M, ρ) through LSSS, and M is a $l \times b$ matrix.
  - Step 2. Randomly select $r_i \in \mathbb{Z}_p^*$ for $i \in [2, b]$ and set $v = (t, r_2, \ldots, r_b)^T$(where, t is unknown).
  - Step 3. Compute $\lambda_i = M_i \cdot v$ for $i \in [1, l]$, where M_i is the ith row of M. Let $M_i = (x_1, x_2, \ldots, x_b)$, and

$$\lambda_i = M_i \cdot v = (x_1, x_2, \ldots, x_b) \cdot (t, r_2, \ldots, r_b)^T = x_1 t + \sum_{i=2}^{b} x_i r_i, \quad (20)$$

  where, t is unknown, so λ_i is unknown.
  - Step 4. Set $c' = g^t = G^t$ and compute $c_0 = g^{\beta t} = (G^t)^{\beta}$.
  - Step 5. For $i \in [1, l]$, the sub-cipher (c_{i1}, c_{i2}) can be computed as follows:
    If the ith predicate is $A_i \in S_i$, set $c_{i1} = G^{\lambda_i}$, and compute c_{i2} as follows:

$$c_{i2} = \left(v_i \cdot \prod_{e_{ik}\in S_i} g_{n-k}\right)^{\lambda_i} = \left(G^{s_i\zeta} / \prod_{e_{ik}\in S_i} g_{n-k} \cdot \prod_{e_{ik}\in S_i} g_{n-k}\right)^{\lambda_i} = G^{s_i\zeta\lambda_i}, \quad (21)$$

where, G^{λ_i} and $(G^{s_i\zeta})^{\lambda_i}$ can be computed, since G^t is known, despite t is unknown. The computation processes are shown as follows:

$$G^{\lambda_i} = G^{x_1 t + \sum_{i=2}^{b} x_i r_i} = (G^t)^{x_1} \cdot \prod_{i=2}^{b} G^{x_i r_i}; \quad (22)$$

$$(G^{s_i\zeta})^{\lambda_i} = (G^{s_i\zeta})^{x_1 t + \sum_{i=2}^{b} x_i r_i} = (G^t)^{s_i\zeta x_1} \cdot \prod_{i=2}^{b} G^{s_i\zeta x_i r_i}. \quad (23)$$

If the ith predicate is $A_i \notin R_i$, set $R_i = U_i/S_i$, then compute w_i as follows:

$$w_i = v_i \cdot h = G^{s_i\zeta+\delta} / \prod_{e_{ik}\in S_i} G_{n-k} \cdot \prod_{e_{ik}\in U_i} G_{n-k} = G^{s_i\zeta+\delta} \cdot \prod_{e_{ik}\in R_i} G_{n-k}. \quad (24)$$

Moreover, set $c_{i1} = G^{\lambda_i}$, and compute c_{i2} as follows:

$$c_{i2} = \left(w_i / \prod_{e_{ik}\in R_i} g_{n-k}\right)^{\lambda_i} = \left(G^{s_i\zeta+\delta} \cdot \prod_{e_{ik}\in R_i} G_{n-k} / \prod_{e_{ik}\in R_i} G_{n-k}\right) = (G^{s_i\zeta+\delta})^{\lambda_i}, \quad (25)$$

where, $G^{\lambda_i} = (G^t)^{x_1} \cdot \prod_{i=2}^{b} G^{x_i r_i}$, and

$$(G^{s_i\zeta+\delta})^{\lambda_i} = (G^{s_i\zeta})^{\lambda_i} \cdot (G^{\delta})^{\lambda_i} = (G^{s_i\zeta})^{x_1 t + \sum_{i=2}^{b} x_i r_i} \cdot (G^{\delta})^{x_1 t + \sum_{i=2}^{b} x_i r_i} = (G^t)^{s_i\zeta x_1} \cdot \prod_{i=2}^{b} G^{s_i\zeta x_i r_i} \cdot (G^t)^{\delta x_1} \cdot \prod_{i=2}^{b} G^{\delta x_i r_i}. \quad (26)$$

Based on the above, the ciphertext is $C_{\Pi^*} = (\Pi, M, \rho, c', c_0, \{c_{i1}, c_{i2}\}_{i=1}^{l})$. $\mathcal{B}$ randomly picks $\sigma \in \{0, 1\}$. Then he computes $C_\sigma = m_\sigma \oplus W$. At last, $\mathcal{B}$ sends the tuple (C_{Π^*}, C_σ) to $\mathcal{A}$.

- **Response.** $\mathcal{A}$ outputs σ' as the guess. If $\sigma' = \sigma$, $\mathcal{B}$ outputs 1; otherwise outputs 0.

The validation of secret keys. Next we analyze validation of the simulated private keys in the following cases.

- **Case 1:** The predicate is $A_i \in S_i'$ and the attribute is $A_i \leftarrow e_{ij}$. In this case, we verify the secret key $d_{ij}^{(\phi)}$ of this attribute is valid if $e_{ij} \in S_i'$, where $S_i' \subseteq U_i/S_i$. Under this condition, we can compute

$$c_{i2} = \left(G^{s_i\zeta} / \prod_{eik\in S_i} G_{n-k} \cdot \prod_{e_{ik}\in S_i'} G_{n-k}\right)^{\lambda_i}; \quad (27)$$

$$d_{ij}^{(\phi)} = G_j^{s_i\zeta} / \prod_{e_{ik}\in S_i} G_{n-k+j} \cdot G^{-\phi}, \quad (28)$$

and c_i can be computed as follows:

$$\begin{aligned} c_i &= \frac{e(c_{i2}, g_j)}{e\left(d_{ij}^{(\phi)} \cdot \prod_{e_{ik}\in S_i', k\neq j} g_{n-k+j}, c_{i1}\right)} \\ &= \frac{e\left((G^{s_i\zeta} / \prod_{e_{ik}\in S_i} G_{n-k} \cdot \prod_{e_{ik}\in S_i'} G_{n-k})^{\lambda_i}, G_j\right)}{e\left(G_j^{s_i\zeta} / \prod_{e_{ik}\in S_i} G_{n-k+j} \cdot G^{-\phi} \cdot \prod_{e_{ik}\in S_i', k\neq j} G_{n-k+j}, G^{\lambda_i}\right)} \\ &= \frac{e\left(G^{s_i\zeta} \cdot \prod_{e_{ik}\in S_i} G_{n-k} \cdot \prod_{e_{ik}\in S_i'} G_{n-k}, G_j^{\lambda_i}\right) \cdot e(G^{\phi}, G^{\lambda_i})}{e\left(G^{s_i\zeta} \cdot \prod_{e_{ik}\in S_i} G_{n-k} \cdot \prod_{e_{ik}\in S_i', k\neq j} G_{n-k}, G_j^{\lambda_i}\right)} \\ &= e(G_{n-j}, G_j^{\lambda_i}) \cdot e(G^{\phi}, G^{\lambda_i}) = e(G_n \cdot G^{\phi}, G^{\lambda_i}). \end{aligned} \quad (29)$$

- **Case 2:** The predicate is $A_i \notin R_i'$ and the attribute is $A_i \leftarrow e_{ij}$. In this case, we verify the secret key $d_{ij}^{(\phi)}$ of the attribute e_{ij} is

valid if $e_{ij} \in S_i'$, where $S_i' \subseteq U_i/(S_i \cup R_i')$. Under this condition, we have

$$\begin{aligned} c_{i2} &= \left(w_i / \prod_{e_{ij} \in R_i'} g_{n-j} \right)^{\lambda_i} \\ &= \left(G^{s_i\zeta+\delta} / \prod_{e_{ik} \in S_i} G_{n-k} \cdot \prod_{e_{ik} \in U_i} G_{n-k} / \prod_{e_{ik} \in R_i'} G_{n-k} \right)^{\lambda_i} \\ &= \left(G^{s_i\zeta+\delta} / \prod_{e_{ik} \in S_i} G_{n-k} \cdot \prod_{e_{ik} \in S_i'} G_{n-k} \right)^{\lambda_i}; \end{aligned} \tag{30}$$

$$d_{ij}^{(\phi)} = G_j^{s_i\zeta} / \prod_{e_{ik} \in S_i} G_{n-k+j} \cdot G^{-\phi}, \tag{31}$$

and c_i can be computed as follows:

$$\begin{aligned} c_i &= \frac{e(c_{i2}, g_j)}{e\left(d_{ij}^{(\phi)} \cdot h_j / \prod_{e_{ik} \in R_i'} g_{n-k+j}, c_{i1}\right)} \\ &= \frac{e\left(G^{s_i\zeta+\delta} / \prod_{e_{ik} \in S_i} G_{n-k} \cdot \prod_{e_{ik} \in S_i'} G_{n-k}, G_j^{\lambda_i}\right) \cdot e(G^{\phi}, G^{\lambda_i})}{e\left(G_j^{s_i\zeta+\delta} / \prod_{e_{ik} \in S_i} G_{n-k+j} \cdot \prod_{e_{ik} \in U_i, k \neq j} G_{n-k+j} / \prod_{e_{ik} \in R_i'} G_{n-k+j}, G^{\lambda_i}\right)} \\ &= \frac{e\left(G^{s_i\zeta+\delta} / \prod_{e_{ik} \in S_i} G_{n-k} \cdot \prod_{e_{ik} \in S_i'} G_{n-k}, G_j^{\lambda_i}\right) \cdot e(G^{\phi}, G^{\lambda_i})}{e\left(G^{s_i\zeta+\delta} / \prod_{e_{ik} \in S_i} G_{n-k} \cdot \prod_{e_{ik} \in S_i', k \neq j} G_{n-k}, G_j^{\lambda_i}\right)} \\ &= e(G_{n-j}, G_j^{\lambda_i}) \cdot e(G^{\phi}, G^{\lambda_i}) = e(G_n \cdot G^{\phi}, G^{\lambda_i}). \end{aligned} \tag{32}$$

Advantage Evaluation Now we analysis the advantage of $\mathcal{A}$ as follows:

$$\begin{aligned} Adv_{\mathrm{DM-ABE}}^{\mathrm{IND-CPA}}(\mathcal{A}) &= \left| \Pr[\sigma' = \sigma] - \frac{1}{2} \right| = \frac{1}{2} |\Pr[\sigma' = \sigma] - 1| \\ &= \frac{1}{2} \left| \Pr[\sigma' = 1 | \sigma = 1] - \Pr[\sigma' = 1 | \sigma = 0] \right|. \end{aligned} \tag{33}$$

Note that, the advantage of $\mathcal{A}$ is based on the condition that $W = e(G^{\mu^n}, G)^t$, which ensures the ciphertext is valid. The advantage of $\mathcal{B}$ is based on the advantage of $\mathcal{A}$, and it can be computed as follows:

$$\begin{aligned} Adv_{\mathrm{DBDHE}}^{\mathrm{IND}}(\mathcal{B}) &= |\Pr[\mathcal{B}(\mathcal{R}, e(G^{\mu^n}, G)^t) = 1 : G \xleftarrow{R} \mathbb{G}, \mu, t \xleftarrow{R} \mathbb{Z}_p^*] \\ &- \Pr[\mathcal{B}(\mathcal{R}, W) = 1 : G \xleftarrow{R} \mathbb{G}, \mu, \\ &\quad t \xleftarrow{R} \mathbb{Z}_p^*, W \xleftarrow{R} \mathbb{G}_T]| \\ &= |\Pr[\sigma' = \sigma : W = e(G^{\mu^n}, G)^t] \\ &- \Pr[\sigma' = \sigma : W \xleftarrow{R} \mathbb{G}_T]| \\ &= \left| \frac{1}{2} \Pr[\sigma' = 1 : \sigma = 1 \wedge W = e(G^{\mu^n}, G)^t] \right. \\ &+ \left. \frac{1}{2} \Pr[\sigma' = 0 : \sigma = 0 \wedge W = e(G^{\mu^n}, G)^t] - \frac{1}{2} \right| \\ &= \frac{1}{2} |\Pr[\sigma' = 1 | \sigma = 1 \wedge W = e(G^{\mu^n}, G)^t] \\ &- \Pr[\sigma' = 1 | \sigma = 0 \wedge W = e(G^{\mu^n}, G)^t]| \end{aligned} \tag{34}$$

where $\mathcal{R} = (G, G^t, \{G^{\mu^i}\}_{i=1, i \neq n}^{2n})$. Based on the advantage evaluation, we get $Adv_{\mathrm{DBDHE}}^{\mathrm{IND}}(\mathcal{B}) = Adv_{\mathrm{DM-ABE}}^{\mathrm{IND-CPA}}(\mathcal{A})$. According to the hypothesis that $Adv_{\mathrm{DM-ABE}}^{\mathrm{IND-CPA}}(\mathcal{A}) \geq \epsilon$, we have $Adv_{\mathrm{DBDHE}}^{\mathrm{IND}}(\mathcal{B}) \geq \epsilon$. This is opposite to the definition of DBDHE assumption, so the hypothesis is wrong, i.e., $Adv_{\mathrm{DM-ABE}}^{\mathrm{IND-CPA}}(\mathcal{A}) < \epsilon$. Consequently, our DM-ABE scheme is (ϵ, n)-semantically secure against chosen plaintext attack under the (ϵ, n)-DBDHE assumption, and the advantage of $\mathcal{A}$ is $Adv_{\mathrm{DM-ABE}}^{\mathrm{IND-CPA}}(\mathcal{A}) < \epsilon$.

4. Performance analysis

In this section, we will analyze the performance of the DM-ABE in terms of computation and storage complexity, and then simulate the scheme to evaluate the execution time. Moreover, we provide the performance comparison between the DM-ABE and some existing CP-ABE schemes.

4.1. Complexity analysis

Before the complexity analysis of DM-ABE, we present some symbols for clarity shown in Table 3. Note that, we neglect the execution time of the operations in $\mathbb{Z}_p^*$, the hash function and the matrix multiplication, since they are much more efficient than exponentiation and pairing operations. The computation complexity analysis of DM-ABE is proposed in Table 4.

As shown in Table 4, the computation complexity of **Setup** is directly proportional to both m and n. The computation complexity of **KeyGen** is directly proportional to $|\Phi|$. The computation complexities of both **DMABE** − **Enc** and **DMABE** − **Dec** are directly proportional to $|S_i|$ and $|R_i|$. However, the computation complexity of **DMABE** − **Enc** is also directly proportional to l ($|\in| + |\notin| = l$), while the one of **DMABE** − **Dec** is directly proportional to $|I|$ ($|\in_d| + |\notin_d| = |I|$). Then, we propose the storage complexity analysis of DM-ABE in Table 5.

As shown in Table 5, the storage complexity of public key is directly proportional to m and n. The storage complexity of private key is directly proportional to $|\Phi|$. Moreover, the communication complexities of **DMABE** − **Enc** and **DMABE** − **Dec** are directly proportional to l and $|I|$, respectively.

4.2. Experimental analysis

In order to evaluate the practical performance of our DM-ABE scheme, we simulate four algorithms, **Setup**, **KeyGen**, **DMABE-Enc** and **DMABE-Dec**, based on Java Pairing Based Cryptographic Library (JPBC). The experiments are executed on 64-bit Windows 10 under Intel(R) Core(TM) i3-3240 CPU @3.40 GHz, 12.0 G ROM. In the experiment, we use a 160-bit elliptic curve group of type A, $y^2 = x^3 + x$, over a 512-bit finite field. The reason of choosing type A is that the size of public key PK will be increased if we choose an asymmetric elliptic curve group.

For the algorithm **Setup**, we carry out two experiments to evaluate the execution time shown as follows:

(1) The number of assignments for each attribute variable (i.e., n) is set as 100, while the number of attribute variables (i.e., m) increases from 10 to 100. The execution time is shown in Fig. 1(a).
(2) The number of attribute variables (i.e., m) is set as 100, while the number of assignments for each attribute variable (i.e., n) increases from 10 to 100. The execution time is shown in Fig. 1(b).

As shown in Fig. 1, the change trends of execution time for **Setup** are incremented. Specifically, the execution time of **Setup** rises from 3579 ms to 4653 ms, with m increasing from 10 to 100 in Fig. 1(a). In Fig. 1(b), the execution time of **Setup** rises from 1643 ms to 4573 ms, with n increasing from 10 to 100. It is easy to find that the execution time of **Setup** is not only directly proportional to m, but also to n.

Table 3
Definitions of the symbols used in complexity analysis.

Symbol	Description
$E(\mathbb{G})$, $E(\mathbb{G}_T)$	The execution time of one exponentiation operation in $\mathbb{G}$ and $\mathbb{G}_T$, respectively
$M(\mathbb{G})$	The execution time of one multiplication operation in $\mathbb{G}$
$D(\mathbb{G})$, $D(\mathbb{G}_T)$	The execution time of one division operation in $\mathbb{G}$ and $\mathbb{G}_T$, respectively
B	The execution time of one bilinear pairing operation $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$
$l\mathbb{Z}_p$, $l_{\mathbb{G}}$, $l\mathbb{G}_T$	The length of one element in $\mathbb{Z}_p$, $\mathbb{G}$ and $\mathbb{G}_T$, respectively
m	The number of attribute variables
n	The number of assignments for each attribute variable
l	The number of predicates in the policy
$\|\Phi\|$, $\|I\|$	The number of elements in Φ and I, respectively
$\| \in \|$, $\| \notin \|$	The number of PM-type predicates and NM-predicates in the policy, respectively
$\| \in_d \|$, $\| \notin_d \|$	The number of attributes used for PM-type predicates and NM-type predicates in decryption, respectively
$\|S_i\|$	The number of elements in S_i for the PM-type predicate $A_i \in S_i$ in the policy
$\|R_i\|$	The number of elements in R_i for the NM-type predicate $A_i \notin R_i$ in the policy

Table 4
Computation complexity analysis of DM-ABE.

Algorithm	Computation complexity
Setup	$(m + 3n + 2) \cdot E(\mathbb{G}) + n \cdot D(\mathbb{G})$
KeyGen	$(\|\Phi\| + 2) \cdot E(\mathbb{G}) + \|\Phi\| \cdot M(\mathbb{G})$
DMABE − Enc	$(2l + 2) \cdot E(\mathbb{G}) + (\| \in \| \cdot (\|S_i\| + 1) + \| \notin \| \cdot \|R_i\|) \cdot M(\mathbb{G}) + \| \notin \| \cdot D(\mathbb{G}) + 2 \cdot B + D(\mathbb{G}_T)$
DMABE − Dec	$(2\|I\| + 1) \cdot B + [\| \in_d \| \cdot \|S_i\| + \| \notin_d \| \cdot (\|R_i\| + 2)] \cdot M(\mathbb{G}) + \|I\| \cdot M(\mathbb{G}_T) + \|I\| \cdot E(\mathbb{G}_T) + \| \in_d \| \cdot D(\mathbb{G}) + D(\mathbb{G}_T)$

Table 5
Storage complexity of our ABE scheme.

Algorithm	Storage/Communication complexity
Setup	$4l_{\mathbb{Z}_p}$ (for MK); $(m + 3n + 4) \cdot l_{\mathbb{G}}$ (for PK)
KeyGen	$(\|\Phi\| + 1) \cdot l_{\mathbb{G}}$ (for $sk_{ij}^{(\phi)}$)
DMABE − Enc	$(l \cdot b) \cdot l_{\mathbb{Z}_p} + (2l + 2) \cdot l_{\mathbb{G}}$ (for C_{Π}); $l_{\mathbb{G}_T}$ (for ek)
DMABE − Dec	$(\|I\| + 2) \cdot l_{\mathbb{G}_T}$

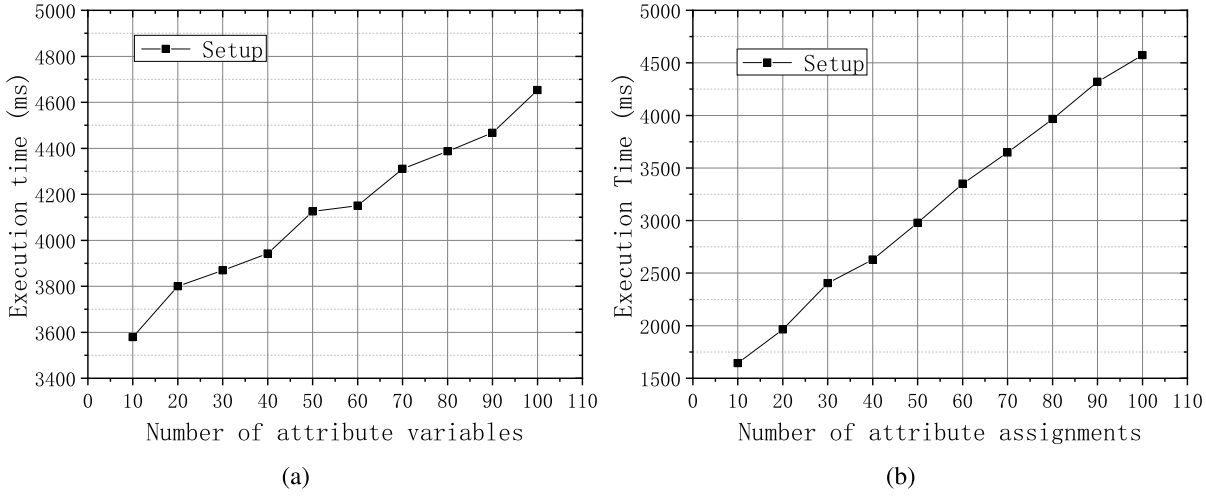

Fig. 1. Execution time of **Setup**.

For the algorithm **KeyGen**, we design an experiment to evaluate the execution time. In this experiment, the number of the user's attributes (i.e., $|\Phi|$) increases from 10 to 100. The experimental results are shown in Fig. 2.

As shown in Fig. 2, the execution time of **KeyGen** is about 221 ms–2182 ms, where $|\Phi|$ increases from 10 to 100. It indicates that the execution time of **KeyGen** is directly proportional to $|\Phi|$.

Moreover, we carry out the experiments to evaluate the execution time of two algorithms, **DMABE-Enc** and **DMABE-Dec**, in two cases shown as follows:

- Case 1: In this case, all policies are set in the form, $(A_1 \in S_1) \wedge (A_2 \notin R_2) \wedge \cdots (A_{l-1} \in S_{l-1}) \wedge (A_l \notin R_l)$.
- Case 2: In this case, all policies are set in the form, $(A_1 \in S_1) \vee (A_2 \notin R_2) \vee \cdots (A_{l-1} \in S_{l-1}) \vee (A_l \notin R_l)$.

Note that, the user's attribute set should satisfy all predicates of the policy for decryption in Case 1 while only one predicate of the policy is required to be satisfied for decryption in Case 2.

For Case 1, we design two experiments with different parameters shown as follows:

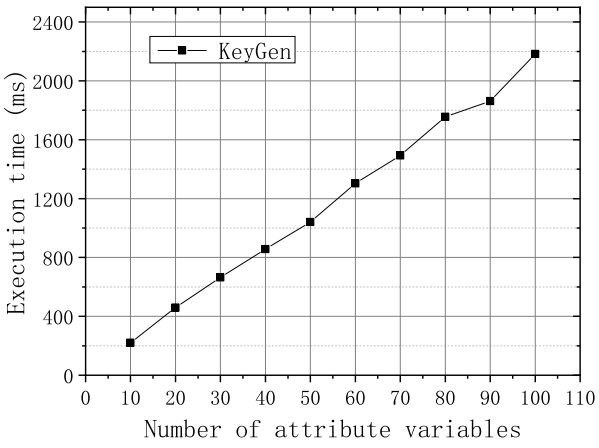

Fig. 2. Execution time of **KeyGen**.

(1) The number of attributes in S_i or R_i for each predicate (we call it n_{att}) in the policies is fixed to 10, while the number of predicates in the policies (i.e., $l = |\in| + |\notin|$) increases from 10 to 100. The execution time of **DMABE-Enc** and **DMABE-Dec** is shown in Fig. 3(a).
(2) l is fixed to 50, while n_{att} increases from 10 to 100. The execution time of **DMABE-Enc** and **DMABE-Dec** is shown in Fig. 3(b).

Since all predicates involved in the policy should be satisfied for correct decryption in the worst situation, the number of attributes used for decryption equals to the number of predicates, i.e., $|I| = l$. In Fig. 3(a), the execution time of **DMABE-Enc** and **DMABE-Dec** raise linearly with the increasing l, and in Fig. 3(b), the execution time of these two algorithms is directly proportional for n_{att}. Specifically, in Fig. 3(a), the execution time of **DMABE-Enc** rises from 240 ms to 2142 ms, and the execution time of **DMABE-Dec** rises from 158 ms to 1878 ms, with the increase of l from 10 to 100. In Fig. 3(b), with increase of n_{att} from 10 to 100, the execution time of **DMABE-Enc** rises from 1084 ms to 1387 ms, and the one of **DMABE-Dec** rises from 814 ms to 1021 ms.

Similarly, we also evaluate the execution time of these two algorithms in Case 2. In this case, the experiments are designed with the parameters similar to Fig. 3(a) and (b), respectively. The experimental results are shown in Fig. 4.

As shown in Fig. 4(a), the execution time of **DMABE-Enc** raises linearly with the increasing l, while the one of **DMABE-Dec** is stable. The reason is that the correct decryption only requires that only one predicate in the policy is satisfied in this case, i.e., $|I| = 1$. In details, with l increasing from 10 to 100, the execution time of **DMABE-Enc** rises from 241 ms to 2090 ms, but the one of **DMABE-Dec** is around 22 ms. As shown in Fig. 4(b), the execution time of **DMABE-Enc** and **DMABE-Dec** raises linearly with the increasing n_{att}, where, the execution time of **DMABE-Enc** rises from 1073 ms to 1291 ms, and that of **DMABE-Dec** raises from 21 ms to 26 ms.

4.3. Performance comparison

In this subsection, we present a comparison between our DM-ABE and some existing schemes in terms of the storage complexity of keys and ciphertexts, supporting logics and predicates, as well as hardness shown in Table 6. Here, suppose that $|sk|$ and $|C|$ denote the storage complexities of private key and ciphertext, respectively. Then, we use d to denote the maximum number of multi-use of attributes in Tomida et al. (2020).

As shown in Table 6, Bethencourt et al. (2007) and Goyal et al. (2008) only support EL, and just can express AND/OR and equivalence decision (=). Comparing with Bethencourt et al. (2007) and Goyal et al. (2008), Waters (2011), Tsuchida et al. (2018), Yamada et al. (2014) and Tomida et al. (2020) can support NEL, where Tsuchida et al. (2018), Yamada et al. (2014) and Tomida et al. (2020) can express non-equivalence decision ($\neq$), but Waters (2011) cannot. The reason is that Waters (2011) regards the negative attribute, e.g. $\neg a$, as an independent attribute. Suppose that x is an attribute variable, this scheme will express $x \neq a$ into $x = \neg a$, which is still equivalence decision in essence. Being different form those schemes, our DM-ABE is more expressive because it supports DML, and can express various types of predicates, $=, \neq, \in$ and $\notin$.

For the hardness, Bethencourt et al. (2007) is proven semantically secure under the generic group model. Goyal et al. (2008) is proven semantically secure under DBDH assumption in generic group model. Waters (2011) is proven semantically secure d-parallel BDHE assumption in standard model. Yamada et al. (2014) is proven semantically secure under n-(B) assumption in generic group model. Tomida et al. (2020) is proven secure under $\mathcal{D}_k$-MDDH assumption in random oracle model. Tsuchida et al. (2018) is proven secure under the $q-$DBDHE assumption in the standard model. Our DM-ABE is proven semantically secure under DBDHE assumption in standard model.

However, Table 6 cannot clearly reflect the outstanding advantage of DM-ABE on communication cost. Suppose that $U = \{u_i\}_{i=1}^n$ is a set of n different assignments for x. We take Yamada et al. (2014) as an example, and use it to express two predicates, $x \in U$ and $x \notin U$. The results are shown as follows:

- For $x \in U$, the scheme needs to divide it into n simple predicates $x = u_i$, and connects them by the logic symbol $\vee$ to get $(x = u_1) \vee (x = u_2) \vee \cdots \vee (x = u_n)$ involving n predicates and $n - 1$ OR operators.
- For $x \notin U$, the scheme needs to divide it into n simple predicates $x \neq u_i$, and connects them by the logic symbol $\wedge$ to get $(x \neq u_1) \wedge (x \neq u_2) \wedge \cdots \wedge (x \neq u_n)$ involving n predicates and $n - 1$ AND operators.

Clearly, Yamada et al. (2014) needs to divide $x \in U$ or $x \notin U$ into n simple predicates. It will result in the number of predicates increasing from 1 to n, and further raise the communication cost because each predicate will be converted into a subciphertext. Being different from the existing CP-ABE schemes, the DM-ABE scheme supports DML, and can efficiently express $x \in U$ or $x \notin U$ without dividing.

5. Policy-driven data sharing architecture based on DM-ABE

5.1. Overview of PDDSA

Nowadays, many novel technologies, e.g., Internet of things (Fan, 2021b), cloud computing (Li et al., 2021a,b; Shantharajah and Maruthavani, 2021), improve the efficiency of data sharing, and enable data sharing more and more widespread in people's daily lives. However, how to overcome the security risk of private data is a popular challenge currently. To face this challenge, we propose the policy-driven data sharing architecture by integration of DM-ABE and Attribute-Based Access Control (ABAC) to provide protection for private data. The model framework of the proposed PDDSA is given in Fig. 5. Here, PDDSA includes two mechanisms, i.e., access control mechanism and ciphertext sharing mechanism, their details are shown as follows:

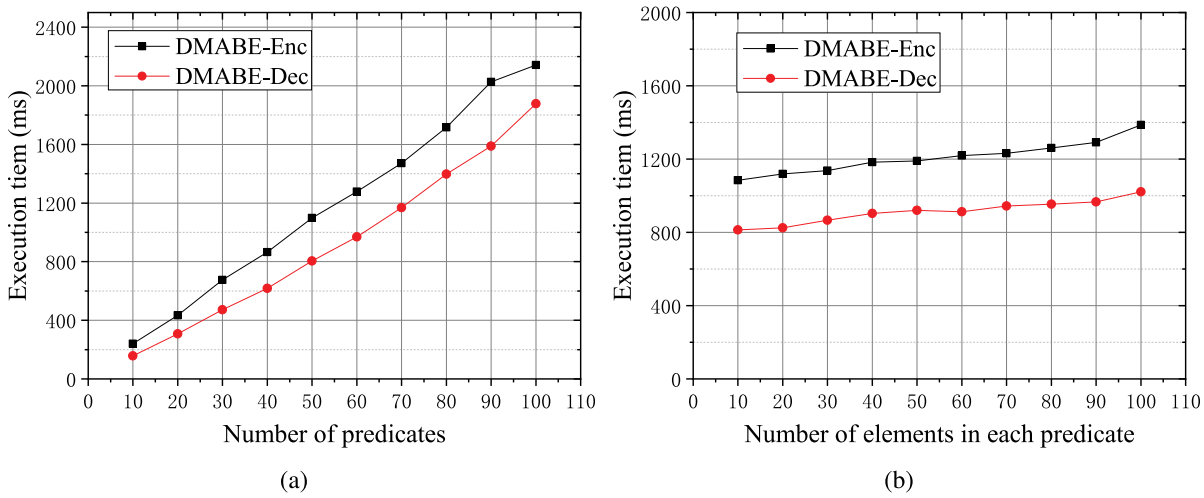

Fig. 3. Execution time of **DMABE-Enc** and **DMABE-Dec** in the worst situation.

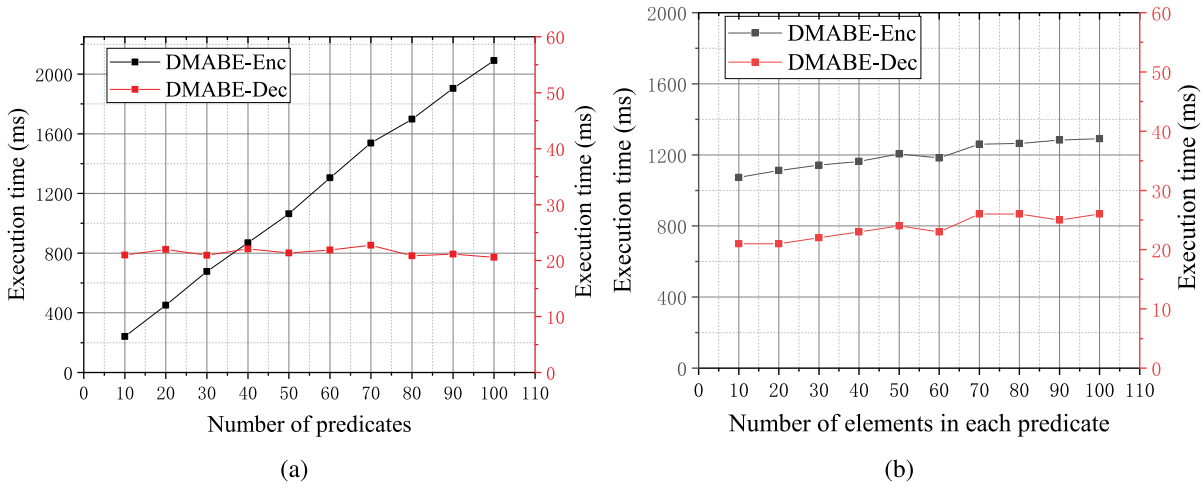

Fig. 4. Execution time of **DMABE-Enc** and **DMABE-Dec** in the best situation.

Table 6
The comparison between DM-ABE and some existing ABE schemes.

Scheme	$\|sk\|$	$\|C\|$	$\wedge$	$\vee$	$=$	$\neq$	$\in$	$\notin$	Supporting logics	Hardness
Bethencourt et al. (2007)	$O(\|\Phi\|)$	$O(l)$	✓	✓	✓	×	×	×	EL	General group model (Boneh et al., 2005)
Goyal et al. (2008)	$O(\|\Phi\| \cdot l^{3.42})$	$O(m \cdot l^{3.42})$	✓	✓	✓	×	×	×	EL	DBDH (Sahai and Waters, 2005)
Waters (2011)	$O(\|\Phi\|)$	$O(l)$	✓	✓	✓	×	×	×	EL, NEL	d-parallel BDHE (Waters, 2011)
Yamada et al. (2014)	$O(\|\Phi\|)$	$O(l)$	✓	✓	✓	✓	×	×	EL, NEL	$n-$(B) (Yamada et al., 2014)
Tomida et al. (2020)	$O(\|\Phi\|)$	$O(l)+O(d)$	✓	✓	✓	✓	×	×	EL, NEL	$\mathcal{D}_k$-MDDH (Escala et al., 2017)
Tsuchida et al. (2018)	$O(\|\Phi\|)$	$O(l)$	✓	✓	✓	✓	×	×	EL, NEL	$q-$DBDHE (Rouselakis and Waters, 2013)
Ours	$O(\|\Phi\|)$	$O(l)$	✓	✓	✓	✓	✓	✓	DML	DBDHE (Xie and Ren, 2014)

- **Access control mechanism**: It corresponds to the orange, the lavender and the green rectangular boxes in Fig. 5. This mechanism is designed on basis of ABAC, because ABAC can provide high flexibility, rich semantics, fine granularity and other properties. Being different from the other access control models, ABAC regards an attribute as the minimum authorization unit, and makes authorization decision by policies and various types of attributes including Subject (*Sub*), Object (*Obj*), Action (*Act*) and Environment (*Env*). ABAC model can be implemented by XACML language, as shown in Fig. 6. PEP is used to interpret the user's requests. PDP is applied to make authorization decision. PAP and PIP are responsible for management of policy and attribute repositories, respectively. PEP and PDP construct the so-called Authorization Services. When a user sends a request to an edge node through a terminal, the PEP of this node interprets this request into the one in the XACML form, and sends it to PDP. The PDP queries the corresponding access policy and attribute set from PAP and PIP, respectively. Then PDP makes authorization decision according to them, and returns the decision result to the PEP. Finally, the PEP performs this request if it is permitted.
- **Ciphertext sharing mechanism**: It is on the basis of the DM-ABE scheme, because of the high expressivity on DMl-type predicates. Under this mechanism, private data is shared in a ciphertext form between edge nodes to avoid

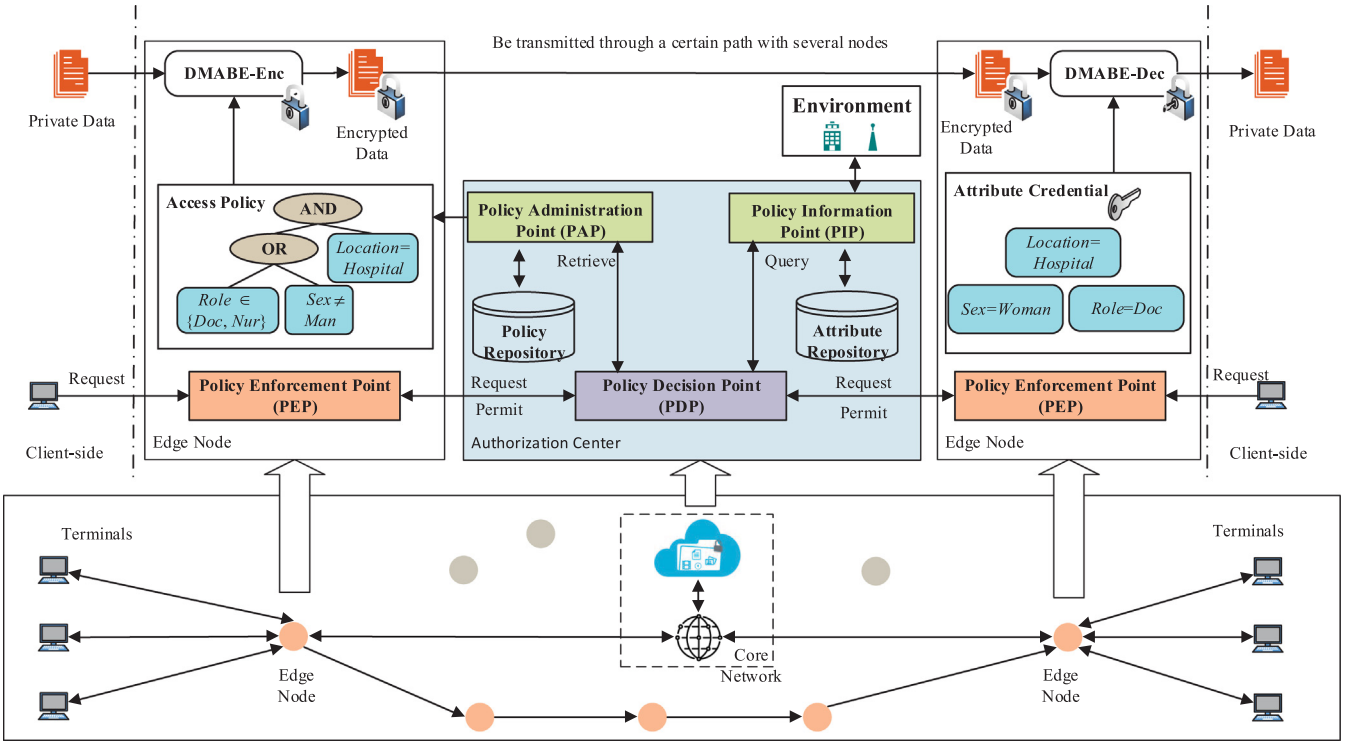

Fig. 5. The model framework of PDDSA. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

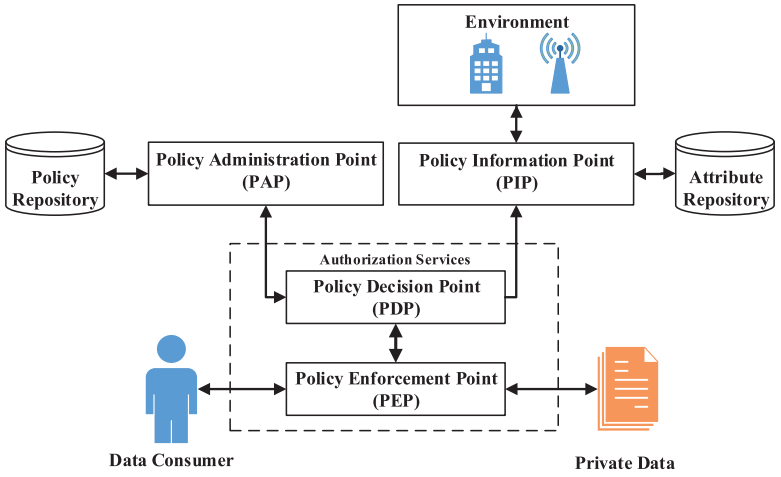

Fig. 6. The model framework of distributed ABAC.

the potential security risks, e.g., privacy leakage. More concretely, private data is issued after it is automatically encrypted into the ciphertext through DM-ABE under the policy of ABAC/XACML. For the receiver, the ciphertext can be decrypted if and only if his/her attribute credentials satisfy the policy.

In PDDSA, the core components include edge nodes and core network, where PEP is deployed in edge node, while the core network consists of PDP, PAP and PIP. We give their details shown as follows:

- **Edge nodes**: which are used for implementing 3 functionalities. The first one is encryption and decryption of data. When a user's request is permitted, PEP automatically encrypts or decrypts data for this user. The second one is storage of encrypted data. The third one is the transmission of encrypted data. If a request for accessing certain encrypted data is permitted, the edge node that possesses the data will transmit the data to the target edge node through a specified path.
- **Core network**: which is applied for authorization decision, and includes PDP, PAP and PIP. When PEP sends some requests to the core network, the PDP will make decision for

these requests with the help of the PAP and the PIP. Moreover, the policy and attribute repositories can be deployed in a certain platform, e.g., cloud.

5.2. Workflow of PDDSA

In this paper, we assume that requests of users only include two basic operations, i.e., *Issue* (for issuing private data) and *Acquire* (for acquiring private data), in our PDDSA. For these two types of requests, PDDSA provides two processes for data issuing and data acquiring. Before description of them, we assume **Setup**() and **KeyGen**() have been invoked, so each of users shares the DM-ABE cryptographic system and public key PK, and further obtains his/her own private key $sk_{\Phi}^{(k)}$.

The process of data issuing: For a data issuing request made by a user, it can be performed only if this user's attribute credentials satisfy a specified policy. The steps are shown as follows:

- Step 1. A user sends his/her data issuing request to an edge node through a terminal. The PEP in this node interprets it into Req, i.e., $Req \leftarrow \mathbf{PEP}(Sub, Act = Issue, Obj, Env)$, and then sends Req to PDP.
- Step 2. After the PDP receives Req, the PDP firstly queries the policies from PAP, i.e., $(\Pi_S, \Pi_O) \leftarrow \mathbf{PDP}^{\mathrm{PAP}}(Req)$, where Π_S is the policy for Req, Π_O is the access policy for the private data $Obj.data$ corresponding to Obj. Furthermore, the PDP queries the attribute set Att form PIP, i.e., $Att \leftarrow \mathbf{PDP}^{\mathrm{PIP}}(\Pi_S, Req)$. Finally, the PDP makes an authorization decision according to Π_S and Att, i.e., $Aut \leftarrow \mathbf{PDP}(\Pi_S, Att)$, where $Aut \in \{0, 1\}$, if $Aut = 1$, this request is permitted; otherwise, it is denied.
- Step 3. If $Aut = 1$, the PDP sends Aut and Π_O to the PEP. This PEP performs $(ek, C_{\Pi_O}) \leftarrow \mathbf{DMABE} - \mathbf{Enc}(PK, \Pi_O)$ to generate the session key ek and ciphertext C_{Π_O}. Furthermore, the PEP encrypts $Obj.data$ into C_O by ek through a certain symmetric encryption algorithm. Finally, PEP destroys ek, $Obj.data$ and saves the encrypted data C_O, C_{Π_O}. Otherwise, the PEP denies this request.

The process of data acquiring: For a data acquiring request made by a user, it can be performed only if this user's attribute credentials satisfy a specified access policy. The steps are shown as follows:

- Step 1. A user sends his/her data acquiring request to an edge node, and the PEP in this node interprets it into Req, i.e., $Req \leftarrow \mathbf{PEP}(Sub, Act = Acquire, Obj, Env)$ further sends Req to PDP.
- Step 2. After the PDP receives Req, it subsequently performs $\Pi_O \leftarrow \mathbf{PDP}^{\mathrm{PAP}}(Req)$ and $Att \leftarrow \mathbf{PDP}^{\mathrm{PIP}}(\Pi_O, Req)$, where Π_O is the access policy for $Obj.data$. Furthermore, the PDP performs $Aut \leftarrow \mathbf{PDP}(\Pi_O, Att)$ for this request.
- Step 3. If $Aut = 1$, the PDP sends Aut to the PEP, and the encrypted data C_O, C_{Π_O} is transmitted to the PEP through a certain path. Then, the PEP obtains the user's $sk_{\Phi}^{(k)}$, and performs $ek \leftarrow \mathbf{DMABE} - \mathbf{Dec}(sk_{\Phi}^{(k)}, C_{\Pi_O})$. Moreover, the PEP decrypts C_O into $Obj.data$ by ek through the corresponding symmetric decryption algorithm. Finally, the PEP returns $Obj.data$ to this user. Otherwise, the PEP denies this request.

In this section, we give the PDDSA and two processes of data issuing and acquiring. In PDDSA, the access control mechanism can be provided for data sharing with high flexibility and fine granularity. Moreover, the ciphertext sharing mechanism based on DM-ABE can provide automatic encryption and decryption of private data according to access policies, without the data owner's intervention. Moreover, this mechanism supports the expression for various types of predicates, e.g., $=$, $\neq$, $\in$ and $\notin$, because of the high expressivity of DM-ABE.

6. Conclusion

The existing ABE schemes has a low expressivity on DML-type predicate, and just support simple predicates $=$ and $\neq$. To improve the expressivity, we construct the DM-ABE scheme. In order to implement the DM-ABE, the SDM algorithm is designed by aggregation functions to securely decide the membership between the verified element and the given set. In this algorithm, any element and set will be converted into a cryptographic element. Furthermore, we construct the DM-ABE based on SDM. Due to the good expressivity of our DM-ABE, a new cryptographic data sharing framework by integrating DM-ABE and ABAC is designed to provide fine-grained access control and security protection for private data. Moreover, we prove that the proposed scheme satisfies semantic security in the standard model under DBDHE assumption. The performance analysis and comparison show that our DM-ABE has a better expressive ability for DML than the existing ABE schemes.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Key Technologies R&D Programs of China under Grant (2018YFB1402702), Beijing Natural Science Foundation, China (4192036) and the National Natural Science Foundation of China under Grant (61972032).

References

Agrawal, S., Chase, M., 2017. FAME: Fast attribute-based message encryption. In: Thuraisingham, B.M., Evans, D., Malkin, T., Xu, D. (Eds.), Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017. ACM, pp. 665–682. http://dx.doi.org/10.1145/3133956.3134014.

Ambrona, M., Barthe, G., Gay, R., Wee, H., 2017. Attribute-based encryption in the generic group model: Automated proofs and new constructions. In: Thuraisingham, B.M., Evans, D., Malkin, T., Xu, D. (Eds.), Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017. ACM, pp. 647–664. http://dx.doi.org/10.1145/3133956.3134088.

Attrapadung, N., Imai, H., 2009. Dual-policy attribute based encryption. In: Abdalla, M., Pointcheval, D., Fouque, P., Vergnaud, D. (Eds.), Applied Cryptography and Network Security, 7th International Conference, ACNS 2009, Paris-Rocquencourt, France, June 2-5, 2009. Proceedings. In: Lecture Notes in Computer Science, vol. 5536, pp. 168–185. http://dx.doi.org/10.1007/978-3-642-01957-9_11.

Bethencourt, J., Sahai, A., Waters, B., 2007. Ciphertext-policy attribute-based encryption. In: 2007 IEEE Symposium on Security and Privacy (S&P 2007), 20-23 May 2007, Oakland, California, USA. IEEE Computer Society, pp. 321–334. http://dx.doi.org/10.1109/SP.2007.11.

Boneh, D., Boyen, X., Goh, E.-J., 2005. Hierarchical identity based encryption with constant size ciphertext. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, pp. 440–456.

Chow, S.S., 2016. A framework of multi-authority attribute-based encryption with outsourcing and revocation. In: Proceedings of the 21st ACM on Symposium on Access Control Models and Technologies. In: SACMAT '16, Association for Computing Machinery, New York, NY, USA, pp. 215–226. http://dx.doi.org/10.1145/2914642.2914659.

Escala, A., Herold, G., Kiltz, E., Rafols, C., Villar, J., 2017. An algebraic framework for Diffie–Hellman assumptions. J. Cryptol. 30 (1), 242–288.

Fan, X., 2021a. Cloud computing task scheduling based on improved bird swarm algorithm. Int. J. Perform. Eng. 17 (1), 85–94. http://dx.doi.org/10.23940/ijpe.21.01.p8.8594.

Fan, X., 2021b. Mobile internet of things dynamic grid QoS service matching mechanism and simulation analysis. Complex. 2021, 2884700:1–2884700:11. http://dx.doi.org/10.1155/2021/2884700.

Goyal, V., Jain, A., Pandey, O., Sahai, A., 2008. Bounded ciphertext policy attribute based encryption. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (Eds.), Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part II - Track B: Logic, Semantics, and Theory of Programming & Track C: Security and Cryptography Foundations. In: Lecture Notes in Computer Science, vol. 5126, Springer, pp. 579–591. http://dx.doi.org/10.1007/978-3-540-70583-3_47.

Goyal, V., Pandey, O., Sahai, A., Waters, B., 2006. Attribute-based encryption for fine-grained access control of encrypted data. In: Juels, A., Wright, R.N., di Vimercati, S.D.C. (Eds.), Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006, Alexandria, VA, USA, October 30 - November 3, 2006. ACM, pp. 89–98. http://dx.doi.org/10.1145/1180405.1180418.

Jiang, R., Wu, X., Bhargava, B., 2016. SDSS-MAC: Secure data sharing scheme in multi-authority cloud storage systems. Comput. Secur. 62, 193–212.

Kim, J., Susilo, W., Guo, F., Au, M.H., Nepal, S., 2017. An efficient KP-ABE with short ciphertexts in prime ordergroups under standard assumption. In: Karri, R., Sinanoglu, O., Sadeghi, A., Yi, X. (Eds.), Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, AsiaCCS 2017, Abu Dhabi, United Arab Emirates, April 2-6, 2017. ACM, pp. 823–834. http://dx.doi.org/10.1145/3052973.3053003.

Lai, J., Deng, R.H., Guan, C., Weng, J., 2013. Attribute-based encryption with verifiable outsourced decryption. IEEE Trans. Inf. Forensics Secur. 8 (8), 1343–1354. http://dx.doi.org/10.1109/TIFS.2013.2271848.

Li, K., 2013. Matrix access structure policy used in attribute-based proxy re-encryption. CoRR abs/1302.6428. arXiv:1302.6428. URL http://arxiv.org/abs/1302.6428.

Li, J., Huang, Q., Chen, X., Chow, S.S.M., Wong, D.S., Xie, D., 2011. Multi-authority ciphertext-policy attribute-based encryption with accountability. In: Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security. In: ASIACCS '11, Association for Computing Machinery, New York, NY, USA, pp. 386–390. http://dx.doi.org/10.1145/1966913.1966964.

Li, K., Jia, L., Shi, X., 2021a. IPSOMC: An improved particle swarm optimization and membrane computing based algorithm for cloud computing. Int. J. Perform. Eng. 17 (1), 135–142. http://dx.doi.org/10.23940/ijpe.21.01.p13.135142.

Li, H., Li, D., Wong, W.E., Wang, D., Zhao, M., 2021b. Kubernetes virtual warehouse placement based on reinforcement learning. Int. J. Perform. Eng. 17 (7), 579. http://dx.doi.org/10.23940/ijpe.21.07.p2.579588.

Li, J., Ren, K., Kim, K., 2009. A2BE: Accountable attribute-based encryption for abuse free access control. IACR Cryptol. ePrint Arch. 2009, 118.

Li, J., Wang, Y., Zhang, Y., Han, J., 2020. Full verifiability for outsourced decryption in attribute based encryption. IEEE Trans. Serv. Comput. 13 (3), 478–487. http://dx.doi.org/10.1109/TSC.2017.2710190.

Lin, H., Luo, J., 2020. Compact adaptively secure ABE from k-lin: Beyond NC[1] and towards NL. In: Canteaut, A., Ishai, Y. (Eds.), Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part III. In: Lecture Notes in Computer Science, vol. 12107, Springer, pp. 247–277. http://dx.doi.org/10.1007/978-3-030-45727-3_9.

Ma, H., Zhang, R., Wan, Z., Lu, Y., Lin, S., 2017. Verifiable and exculpable outsourced attribute-based encryption for access control in cloud computing. IEEE Trans. Dependable Secure Comput. 14 (6), 679–692. http://dx.doi.org/10.1109/TDSC.2015.2499755.

Malluhi, Q.M., Shikfa, A., Trinh, V.C., 2017. A ciphertext-policy attribute-based encryption scheme with optimized ciphertext size and fast decryption. In: Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security. In: ASIA CCS '17, Association for Computing Machinery, New York, NY, USA, pp. 230–240. http://dx.doi.org/10.1145/3052973.3052987.

Ning, J., Cao, Z., Dong, X., Liang, K., Ma, H., Wei, L., 2018a. Auditable σ -time outsourced attribute-based encryption for access control in cloud computing. IEEE Trans. Inf. Forensics Secur. 13 (1), 94–105. http://dx.doi.org/10.1109/TIFS.2017.2738601.

Ning, J., Cao, Z., Dong, X., Wei, L., 2018b. White-box traceable CP-ABE for cloud storage service: How to catch people leaking their access credentials effectively. IEEE Trans. Dependable Secure Comput. 15 (5), 883–897. http://dx.doi.org/10.1109/TDSC.2016.2608343.

Okamoto, T., Takashima, K., 2010. Fully secure functional encryption with general relations from the decisional linear assumption. In: Rabin, T. (Ed.), Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings. In: Lecture Notes in Computer Science, vol. 6223, Springer, pp. 191–208. http://dx.doi.org/10.1007/978-3-642-14623-7_11.

Okamoto, T., Takashima, K., 2012. Fully secure unbounded inner-product and attribute-based encryption. In: Wang, X., Sako, K. (Eds.), Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings. In: Lecture Notes in Computer Science, vol. 7658, Springer, pp. 349–366. http://dx.doi.org/10.1007/978-3-642-34961-4_22.

Ostrovsky, R., Sahai, A., Waters, B., 2007. Attribute-based encryption with non-monotonic access structures. In: Ning, P., di Vimercati, S.D.C., Syverson, P.F. (Eds.), Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007, Alexandria, Virginia, USA, October 28-31, 2007. ACM, pp. 195–203. http://dx.doi.org/10.1145/1315245.1315270.

Rouselakis, Y., Waters, B., 2013. Practical constructions and new proof methods for large universe attribute-based encryption. In: Sadeghi, A., Gligor, V.D., Yung, M. (Eds.), 2013 ACM SIGSAC Conference on Computer and Communications Security, CCS'13, Berlin, Germany, November 4-8, 2013. ACM, pp. 463–474. http://dx.doi.org/10.1145/2508859.2516672.

Sahai, A., Waters, B., 2004. Fuzzy identity based encryption. IACR Cryptol. ePrint Arch. 2004, 86, URL http://eprint.iacr.org/2004/086.

Sahai, A., Waters, B., 2005. Fuzzy identity-based encryption. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, pp. 457–473.

Shantharajah, S., Maruthavani, E., 2021. A survey on challenges in transforming No-SQL data to SQL data and storing in cloud storage based on user requirement. Int. J. Perform. Eng. 17 (8), 703. http://dx.doi.org/10.23940/ijpe.21.08.p6.703710.

Tomida, J., Kawahara, Y., Nishimaki, R., 2020. Fast, compact, and expressive attribute-based encryption. In: Kiayias, A., Kohlweiss, M., Wallden, P., Zikas, V. (Eds.), Public-Key Cryptography - PKC 2020 - 23rd IACR International Conference on Practice and Theory of Public-Key Cryptography, Edinburgh, UK, May 4-7, 2020, Proceedings, Part I. In: Lecture Notes in Computer Science, vol. 12110, Springer, pp. 3–33. http://dx.doi.org/10.1007/978-3-030-45374-9_1.

Tsuchida, H., Nishide, T., Okamoto, E., 2018. Expressive ciphertext-policy attribute-based encryption with fast decryption.. J. Internet Serv. Inf. Secur. 8 (4), 37–56.

Waters, B., 2011. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (Eds.), Public Key Cryptography - PKC 2011 - 14th International Conference on Practice and Theory in Public Key Cryptography, Taormina, Italy, March 6-9, 2011. Proceedings. In: Lecture Notes in Computer Science, vol. 6571, Springer, pp. 53–70. http://dx.doi.org/10.1007/978-3-642-19379-8_4.

Xie, L., Ren, Y., 2014. Efficient anonymous identity-based broadcast encryption without random oracles. Int. J. Digit. Crime Forensics (IJDCF) 6 (2), 40–51.

Xu, J., Chaoran, L., Lu, L., Wang, X., 2020. Smart mattress system based on internet of things. Int. J. Perform. Eng. 16 (9), 1460–1467. http://dx.doi.org/10.23940/ijpe.20.09.p15.14601467.

Yamada, S., Attrapadung, N., Hanaoka, G., Kunihiro, N., 2014. A framework and compact constructions for non-monotonic attribute-based encryption. In: Krawczyk, H. (Ed.), Public-Key Cryptography - PKC 2014 - 17th International Conference on Practice and Theory in Public-Key Cryptography, Buenos Aires, Argentina, March 26-28, 2014. Proceedings. In: Lecture Notes in Computer Science, vol. 8383, Springer, pp. 275–292. http://dx.doi.org/10.1007/978-3-642-54631-0_16.

Yu, G., Ma, X., Cao, Z., Zhu, W., Zeng, J., 2017. Accountable multi-authority ciphertext-policy attribute-based encryption without key escrow and key abuse. In: International Symposium on Cyberspace Safety and Security. Springer, pp. 337–351.

Zhang, Y., Deng, R.H., Xu, S., Sun, J., Li, Q., Zheng, D., 2020a. Attribute-based encryption for cloud computing access control: A survey. ACM Comput. Surv. 53 (4), 83:1–83:41. http://dx.doi.org/10.1145/3398036.

Zhang, Z., Zeng, P., Pan, B., Choo, K.-K.R., 2020b. Large-universe attribute-based encryption with public traceability for cloud storage. IEEE Internet Things J. 7 (10), 10314–10323. http://dx.doi.org/10.1109/JIOT.2020.2986303.

Zhong, H., Zhou, Y., Zhang, Q., Xu, Y., Cui, J., 2021. An efficient and outsourcing-supported attribute-based access control scheme for edge-enabled smart healthcare. Future Gener. Comput. Syst. 115, 486–496.