Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 8513869, 15 pages
https://doi.org/10.1155/2021/8513869

Research Article

Enabling Efficient Decentralized and Privacy Preserving Data Sharing in Mobile Cloud Computing

Jiawei Zhang,[1] **Ning Lu**,[2,3] **Teng Li**,[1] **and Jianfeng Ma**[1]

[1]*School of Cyber Engineering, Xidian University, Xi'an, China*
[2]*School of Computer Science and Technology, Xidian University, Xi'an, China*
[3]*College of Computer Science and Engineering, Northeastern University, Qinhuangdao, China*

Correspondence should be addressed to Ning Lu; luning@neuq.edu.cn

Received 2 July 2021; Revised 31 July 2021; Accepted 10 August 2021; Published 2 September 2021

Academic Editor: Weizhi Meng

Mobile cloud computing (MCC) is embracing rapid development these days and able to provide data outsourcing and sharing services for cloud users with pervasively smart mobile devices. Although these services bring various conveniences, many security concerns such as illegally access and user privacy leakage are inflicted. Aiming to protect the security of cloud data sharing against unauthorized accesses, many studies have been conducted for fine-grained access control using ciphertext-policy attribute-based encryption (CP-ABE). However, a practical and secure data sharing scheme that simultaneously supports fine-grained access control, large university, key escrow free, and privacy protection in MCC with expressive access policy, high efficiency, verifiability, and exculpability on resource-limited mobile devices has not been fully explored yet. Therefore, we investigate the challenge and propose an Efficient and Multiauthority Large Universe Policy-Hiding Data Sharing (EMA-LUPHDS) scheme. In this scheme, we employ fully hidden policy to preserve the user privacy in access policy. To adapt to large scale and distributed MCC environment, we optimize multiauthority CP-ABE to be compatible with large attribute universe. Meanwhile, for the efficiency purpose, online/offline and verifiable outsourced decryption techniques with exculpability are leveraged in our scheme. In the end, we demonstrate the flexibility and high efficiency of our proposal for data sharing in MCC by extensive performance evaluation.

1. Introduction

As an emerging paradigm, mobile cloud computing (MCC) is growing exponentially and facilitates the deployment of enormous mobile devices covering public and private sectors [1]. The MCC systems provide not only strong mobility but also abundant computing and storage capacity for these resource-limited devices which prefer to outsource their data to MCC for cost saving [2]. Moreover, assisted by the data sharing service of MCC, users are able to conveniently enjoy various applications, such as smart home, smart office, and intelligent transportation, with pervasive and smart mobile devices [3]. In particular, this trend is being accelerated with the implementation of 5G communication network offering massive high-speed access capacity [4]. As shown in Figure 1, users can share their data in MCC conveniently with different kinds of mobile devices, e.g., laptops, cellphones, through gNBs (i.e., next generation NodeB) of 5G network, or even satellites receiving station on various sites (e.g., home, hotel, plain, or car). Although MCC, such as iCloud and OneDrive, can provide a variety of benefits to mobile users, the data security issues, i.e., data confidentiality and fine-grained access control, have become important stumbling blocks for the usage of MCC [5]. The data outsourced to MCC may contain numerous sensitive information or significant assets relevant to mobile users and terminates [6]. Thus, the most critical data security concern is the data access control issues that allows only authorized users while prevents unauthorized ones from accessing the shared data in MCC as it will cause severe consequences if the private information is leaked.

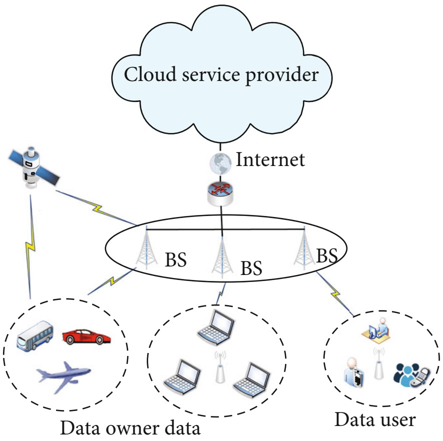


Figure 1: The architecture of mobile cloud computing.

Therefore, how to protect these sensitive data outsourced in MCC remains an urgent challenge. As a promising technique, ciphertext-policy attribute-based encryption (CP-ABE) [7–9] can be adopted to provide fine-grained data access control when user data is shared with multiple users. Nevertheless, the conventional CP-ABE schemes are unsuitable to be directly utilized in secure data sharing of the MCC system as there still exist several issues. First of all, in general CP-ABE schemes, ciphertexts are stored in Cloud Service Provider (CSP) and shared with multiple users together with the access policies which are in plaintext and may cause user privacy leakage [7]. Moreover, a MCC system involves large amount of mobile devices and users, and standard CP-ABE schemes with bounded attribute universe and single attribute authority are no longer satisfactory due to their inflexibility, key escrow, and single-point failure problems [10]. Furthermore, as the CSPs are untrusted in terms of users, they may misbehave in outsourced decryption by returning previous results or even random and false results to users [2]. In addition, the low efficiency in encryption and decryption is an inferior drawback for traditional CP-ABE schemes when used in MCC with enormous resource-limited mobile devices. Thus, it is urgent to design a practical data access control scheme for data sharing in MCC that can address these issues.

To find out a solution, many works have made great progresses. The scheme in [11] solves the problem of bounded attribute universe, key escrow, and single-point failure problem while it cannot support policy privacy preserving and decryption verifiability. Meanwhile, the schemes in [12, 13] support efficient encryption and exculpable decryption as well as multiauthority and policy hidden, respectively. Recently, the authors in [6, 7] proposed two CP-ABE schemes with privacy preserving and expressive policy, but both of them do not support large attribute universe and exculpability of decryption. Then, the scheme in [1] provides features of multiauthority, large attribute universe, and high efficiency to resist key escrow problem, but it cannot satisfy policy preserving and verifiability with exculpability. Besides, such schemes do not consider the issue of exculpability as in some cases, and CSP acts honestly may be framed by users. Although the scheme in [14] fixed this problem, it fails to protect the user privacy in policies. Hence, it is urgent to devise a significant data sharing scheme that is addressing all these drawbacks in traditional CP-ABE schemes at the same time when used in MCC, including key escrow resistance, large attribute universe, privacy preserving, expressive policy, and efficiency.

1.1. Our Contributions.

1.1. Our Contributions. Confronting the above problems, we propose an Efficient and Multiauthority Large Universe Policy-Hiding Data Sharing (EMA-LUPHDS) scheme to achieve key escrow resistance, expressive access policies, and high efficiency for data sharing of MCC with resource-limited mobile devices by extending the decentralized CP-ABE scheme [15]. In particular, our main contributions are listed as follows:

(i) *Single-Point Failure and Key Escrow Free.* To adapt to decentralized environment of the MCC system, EMA-LUPHDS introduces the architecture of multiple authority for user key distribution so as to prevent single-point failure and key escrow problem in a centralized single authority.

(ii) *Hidden Access Policy over Large Attribute Universe.* EMA-LUPHDS leverages fully hidden access policy to solve the user privacy leakage problem in most of current CP-ABE schemes with cleartext access policy shared with the ciphertexts in CSP which may lead to private information leakage. To be flexible in the setup of large-scale MCC systems, EMA-LUPHDS supports large attribute universe with constant size of system parameter.

(iii) *Cost Saving in Encryption and Decryption.* To save the computation cost in both encryption and decryption, online/offline technique is introduced into EMA-LUPHDS for efficient data encryption. Moreover, EMA-LUPHDS achieves outsourced decryption in order to improve efficiency by moving a majority of computation cost of mobile devices with poor resources to CSP.

(iv) *Verifiability and Exculpability.* To guarantee the correctness of outsourced decryption executed by CSP, EMA-LUPHDS can check the result of partially decrypted ciphertext transformed by untrusted CSP with a data verification approach. For the exculpability of CSP, it achieves a commitment mechanism using Pedersen commitment approach.

(v) *Security and Efficiency.* We present security analysis and performance evaluation of the proposed scheme. The result demonstrates that our proposal is secure and efficient, which is extremely practicable and suitable for MCC systems.

1.2. Organization. The remainder of the article is outlined below. Some relevant studies are reviewed in Section 2, and the preliminaries including related definitions and notations are introduced in Section 3. In Section 4, we give the system model, threat model, and design goals of our scheme together with the system definition. Based on this, we describe in detail the constructions of the proposal in Section 5. Section 6 follows this to discuss the security of the scheme, and its performance evaluation is conducted in Section 7. Finally, Section 8 makes a conclusion for the work in this article.

2. Related Work

Mobile cloud computing (MCC) is widely utilized in various applications in which huge number of data plays an important role. Thus, how to protect the security of such large volume data is a big challenge for MCC [3]. CP-ABE is a promising technique for data confidentiality and fine-grained access control which is first introduced in [16] based on the scheme in [17] aside from the user authentication protocols [18–21] as user-centric access control. Due to the data-centric and flexible access control, CP-ABE has been broadly studied and applied [8, 9, 22–26]. However, MCC is a large scale and distributed system involving mobile and resource-limited user devices with much privacy in their data, and the standard CP-ABE schemes cannot be directly employed in MCC applications due to their high cost in computation and dependence on centralized authority.

To confront the bottleneck of single authority, the study in [15] designed a scheme based on [27, 28] with fully multi-authority, but it is inefficient and cannot resist collusion attack. As a solution, borrowing the idea of outsourced decryption proposed in [29–33] based on [34], the scheme in [29] improved the decryption efficiency and the DACC in [35] utilized Key Distribution Centres (KDC) for user key generation across multiple groups to resist collusion attack. Later, the proposals in [30, 36] enhance the DACC scheme in addressing both user collusion and revocation problems. Recently, to solve the problem of deploying CP-ABE in MCC applications for data access control, the schemes in [2] proposed a solution based on [37] and outsourced decryption with anonymous techniques to achieve high decryption efficiency in distributed MCC systems, but it only improves efficiency in decryption and cannot support large attribute universe. Thus, motivated by online/offline CP-ABE proposed in [14, 38, 39] based on [40–42], De and Ruj [1] designed a multiauthority CP-ABE with outsourced decryption to achieve high efficiency in both encryption and decryption, whereas it fails to protect user privacy in access policy which is important for MCC applications containing massive private data.

To protect user privacy in plaintext access policy of standard CP-ABE schemes, the research in [43] first presents the idea of partial hidden-policy CP-ABE, but it only supports AND gate policy with weak security. Later, the study in [44] devised a fully secure and partial hidden-policy CP-ABE, but it still suffers from restricted expressiveness in access policy. Then, the scheme in [45] improves its expressiveness, and the work in [46] introduces decryption testing and large universe to improve efficiency and flexibility, but it is computation consuming with composite order groups. To solve this problem, the studies in [47, 48] design two efficient and partial hidden-policy CP-ABE schemes based on prime order groups that support expressive access policy and verifiable outsourced decryption. However, they are weak in the protection of access policy due to their partially hidden policies. As a solution, the research in [49] proposed fully hidden policy for CP-ABE, but it incurs high computation cost. Then, the work in [50] proposed an efficient fully hidden-policy CP-ABE scheme, while it only supports restricted access policy. Recently, the studies in [6, 7] devise two efficient CP-ABE schemes that support fully hidden and expressive access policy, but both schemes do not overcome the efficiency issue in encryption and small attribute universe. Moreover, these schemes fail to support exculpability which guarantees an authorized user has no way to accuse the cloud of outputting incorrect results in outsourced decryption while it was not the case. As a whole, these schemes cannot be used in MCC applications.

To seek a better solution, we propose EMA-LUPHDS for data access control in MCC applications. We make a function comparison in Table 1 between our scheme and several related state-of-the-art schemes in [1, 2, 6, 7, 10, 12–14] in the functionalities of access policy, large attribute universe, multiauthority, hidden access policy, efficient encryption, efficient decryption, verifiability, and exculpability. This demonstrates that our EMA-LUPHDS is more versatile and flexible than other schemes with richer advantages and satisfies the requirements of data access control in MCC applications.

In Table 1, the schemes are compared from the features of access policy, attribute universe, authority, policy hidden, encryption, and decryption efficiency as well as verifiability and exculpability. First of all, from Table 1, we note that the majority of schemes support expressive LSSS access policy which are flexible and expressive in access policy design. Only two schemes in comparison support “AND” threshold access policy which are lack in expressiveness and flexibility. Moreover, from the aspect of attribute universe, the schemes in [1, 7, 10, 14] and ours all support large universe, while only our scheme and the schemes in [6, 7, 13] provide the features of multiple authorities and hidden policy, which can prevent sensitive information leakage from access policy and resist single authority failure. Furthermore, from the aspect of efficiency, it is well accepted to adopt outsourced encryption and decryption in CP-ABE schemes. And we conclude that most of the schemes in Table 1 support outsourced decryption and verifiability simultaneously, while only our scheme and the schemes in [1, 2, 14] also improve the efficiency in encryption by introducing online/offline technique. In addition, to support a strong verifiability and exculpability for outsourced decryption, we also note that the feature of exculpability is only supported by our scheme and those in [12, 14], while the scheme in [12] does not support expressive policy and the scheme in [14] failed to protect sensitive data in policy and is lack of large attribute universe. In general, our proposal can simultaneously support all the features mentioned above.

Table 1: Function comparison.

Scheme	Access policy	Large universe	Multiple authority	Policy hidden	Efficient encryption	Outsourced decryption	Strong verifiability	Exculpability
Scheme in [12]	AND gate	×	×	×	×	√	×	√
Scheme in [11]	LSSS	√	√	×	×	×	×	×
Scheme in [13]	AND gate	×	√	√	×	×	×	×
Scheme in [7]	LSSS	√	√	√	×	√	×	×
Scheme in [1]	LSSS	√	√	×	√	√	×	×
Scheme in [14]	LSSS	√	×	×	√	√	√	√
Scheme in [6]	LSSS	×	√	√	×	√	×	×
Scheme in [2]	LSSS	×	√	×	√	√	×	×
EMA-LUPHDS	LSSS	√	√	√	√	√	√	√

3. Preliminaries

This section provides several notions and definitions in our proposal including access structure and bilinear maps.

3.1. Notations. In our work, $[l1, l2]$ is used to denote the set $\{l1, l1+1, \cdots, l2\}$ and $[n]$ is the set $1, 2, \cdots, n$, where $n \in Z_p^*$, while $|S|$ denotes the length of a string S.

3.2. Access Structure

Definition 1 (Access structures [8]). Let $E = E_1, \cdots, E_n$ be a entity collection. Given a set $C \subseteq 2^E \setminus \varnothing$, it is monotonic if $\forall D, F : D \subseteq F \bigcap D \in C \longrightarrow F \in C$. Then, the set C is also a monotonic access structure, and the subsets in C are called the authorized sets, otherwise the unauthorized sets.

3.3. Linear Secret Sharing Schemes (LSSSs)

Definition 2 (LSSS [25]). Given the attribute universe U_a, an LSSS on it involves (B, δ), where B is an $l \times n$ share-generating matrix on Z_p and the function δ maps a row of B into an attribute in U_a. There are two algorithms: Share and Reconstruction in an LSSS. The former is to create the shares for a secret value s based on B with $\bar{b} = (s, b_2, \cdots, b_n)^T$, where $b_2, \cdots, b_n \in_R Z_p$ by $\lambda_x = B_x \cdot \bar{b}$ as a share of the secret s, while the latter reconstructs s with the secret shares of an authorized set E by finding $I = \{i \mid \delta(i) \in E\} \subseteq \{1, 2, \cdots l\}$ and constances $\omega_i \in Z_p$ to make $\sum_{i \in I} w_i B_i = (1, 0, \cdots, 0)$ hold and compute $\sum_{i \in I} w_i \lambda_i = s$.

3.4. Cryptographic Background

Definition 3 (Bilinear maps [9]). Given p-ordered cyclic groups G and G_T with a generator $w \in G$, where p is a big prime, if a map $\widehat{e} : G \times G \longrightarrow G_T$ is bilinear, it must satisfy the following: (1) bilinearity: $\widehat{e}(u^e, h^f) = \widehat{e}(w, h)^{ef}, \forall e, f \in Z_p$, $u, h \in G$, (2) nondegeneracy: $\widehat{e}(w, w) \neq 1$, and (3) computability: $\widehat{e}(u, h)$ which can be efficiently computed by an algorithm $\forall u, h \in G$.

4. System Model and Design Goals

This section presents the system model, threat model, and design goals of our proposed system before giving the formal definition and security model for EMA-LUPHDS.

4.1. System Model. As detailed in Figure 2, our system involves Cloud Service Provider (CSP), trusted authority (TA), attribute authorities (AAs), data owner (DO), and data user (DU).

(i) CSP provides users with data outsourcing, sharing, and outsourced decryption services as well as unlimited storage and computational resources

(ii) TA is responsible for initiating whole system by generating global public parameters for the whole system and its master keys

(iii) AA takes charges of managing a disparate set of attributes and generating and distributing secret key and transformation key of the authenticated cloud users. The attribute sets managed by any two or more AAs are different from each other

(iv) DO collects important information from mobile devices in MCC and uploads the massive data to CSP. Before outsourcing, DO converts the data with symmetric algorithm and a symmetric key encrypted by a fully hidden access policy for fine-

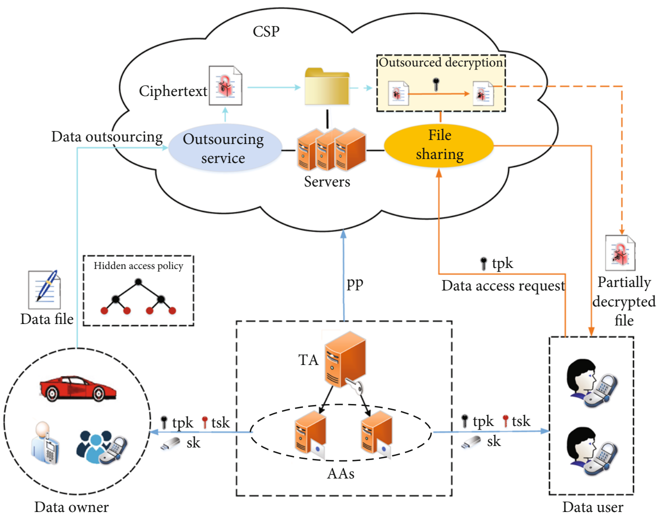


Figure 2: The system model of our EMA-LUPHDS.

grained access control and user privacy preserving. Besides, DO prepares ciphertext components while accessing the power source offline to save computational resource of mobile devices

(v) DU accesses the shared data in CSP on demand with his transformation key for outsourced decryption and recovers the symmetric key if authorized to further decrypt the partially decrypted ciphertext from CSP after verifying its correctness

Based on the above system model, we design our data sharing scheme suitable for the MCC system involving four phases as below.

(i) *Initialization*. TA creates the system global public parameters and master key at the first. All entities can obtain the global public parameters with which each AA can generate their public and secret key pair.

(ii) *User Enrollment*. Each AA issues a secret key and a pair of transformation key for DUs after receiving the joining requests from these DUs. Each AA manages the enrolled DUs as well as their attribute sets.

(iii) *Encryption*. DO encrypts the data (usually in form of files) collected from the smart mobile devices in MCC systems based on a designated access policy and outsources the final ciphertext with fully hidden policies to CSP for data sharing.

(iv) *Decryption*. DU downloads ciphertexts from CSP with his transformation public key for outsourced decryption by CSP. After receiving the partially decrypted data, the DU decrypts it based on transformation private key and checks its correctness.

4.2. Threat Model and Design Goals. In our EMA-LUPHDS, TA, AA, and DO are trusted entities while CSP is deemed to be a semihonest entity which is willing to act with honesty but may leak the private information in an "honest-but-curious" manner. In supplying the outsourced decryption service, CSP may misbehave in returning the result of the partially decrypted ciphertext to DU, such as returning false results or be lazy to return previous results. DUs are regarded as untrusted as they may illegally access the shared data in CSP without authorization or try to break the data security and privacy. Due to these threats on data sharing in MCC, we have the following design goals for our system:

(i) *Data Confidentiality*. The proposed scheme should protect sensitive information in the outsourced data from being leaked or eavesdropped during data sharing and outsourced decryption in CSP and the communication between DU and CSP.

(ii) *Fine-Grained Access Control and Collusion Resistance*. Malicious users who are unauthorized or intend to collude with each other in data access should have no way to recover the ciphertext by aggregating their keys while anyone of them is unauthorized to decrypt the ciphertext alone.

(iii) *Access Policy Hiding*. On account of the access policy shared with ciphertext, those sensitive or privacy-aware information contained and exposed in access policies should be concealed for the purpose of user privacy preserving

(iv) *Verifiability and Exculpability*. Due to the misbehaving CSP, the correctness of outsourced decryption by CSP should be verified. Also, any DU with authorized secret key cannot accuse the CSP of performing incorrectly in outsourced decryption while it acts honestly.

(v) *Efficient Encryption and Decryption*. With respect of resource-limited mobile devices in the MCC system, the computation should be as little as possible for

DU by moving a majority of preparation work offline and offloading the highly operations in decryption to CSP.

4.3. System Definition. We present the definition of our proposed EMA-LUPHDS scheme with the following algorithms:

(i) $Setup_{Global}(\lambda)$. The global setup algorithm is executed by TA. On inputting the security parameter λ, it outputs the global public parameters pp and master key msk

(ii) $Setup_{AA}(pp)$. The authority setup algorithm is in the charge of each AA AA_i managing a disparate set of attributes. It takes as input system global public parameters and outputs their public and secret key pair $(\mathrm{pk}_{\mathrm{AA}_i}, \mathrm{sk}_{\mathrm{AA}_i})$.

(iii) $KeyGen(pp, sk_{AA_i}, GID, S_{i,GID})$. The key generation algorithm is executed by each attribute authority AA_i. It takes as input system global public parameters pp, their secret key $\mathrm{sk}_{\mathrm{AA}_i}$, the global identity GID, and an attribute set $S_{i,\mathrm{GID}}$ of each cloud user. Then, AA_i outputs $\mathrm{sk}_{i,\mathrm{GID}}$ as the secret key associated with the DU identified by GID and their attribute set $S_{i,\mathrm{GID}}$.

(iv) $TKeyGen(pp, sk_{i,GID})$. The transformation key generation algorithm is run by each attribute authority AA_i. It takes the global public parameters pp and the secret key $\mathrm{sk}_{i,\mathrm{GID}}$ of DU identified by GID and outputs the transformation key pair tk of the DU identified by GID.

(v) $Encrypt_{off}(pp)$. The offline encryption algorithm is executed by DO. On inputting the system global public parameters pp, DO generates offline ciphertext component $\mathrm{Key}_{\mathrm{off}}$ and $\mathrm{VC}_{\mathrm{off}}$.

(vi) $Encrypt_{on}(pp, M, (A, \rho))$. The online encryption algorithm is run by DO. On inputting the system global public parameters pp, the specific message M, and the designated access policy (A, ρ), DO generates the final ciphertext CT and outsources it to CSP

(vii) $Decrypt_{\mathrm{OUT}}(pp, tpk_{i,GID}, CT)$. The outsourced decryption algorithm is executed by CSP. It takes as input system global public parameters pp, transformation public key $\mathrm{tpk}_{i,\mathrm{GID}}$, and ciphertext CT and then outputs the partial decrypted ciphertext CT^*.

(viii) $Decrypt_U(pp, tsk_{i,GID}, CT^*)$. The user decryption algorithm is run by DU. It takes the system global public parameters pp, transformation private key $\mathrm{tsk}_{i,\mathrm{GID}}$, and partially decrypted ciphertext CT^* as input and outputs the recovered ciphertext components R^* and key^*.

(ix) $DecVerify(pp, \bar{C}, V_m, key^*, R^*)$. The user decryption verification algorithm is executed by DU. Given the recovered random element R^* and encapsulated key key^*, the DU checks if the session key and encrypted data are valid and output the plaintext M.

5. The Proposed EMA-LUPHDS Scheme

In this section, we describe the overview of our EMA-LUPHDS scheme and its concrete construction.

5.1. Overview. To adapt to the large-scale MCC system, we first design a large universe multiauthority hidden-policy CP-ABE scheme with verifiable and exculpable outsourced decryption to realize efficient data sharing in MCC. Each user in such a distributed architecture is bound up with a global identity (GID) [51] to avoid collision. Moreover, we introduce online/offline technique to further reduce the overhead in data encryption. Before displaying the detailed construction of EMA-LUPHDS scheme, we define that in our EMA-LUPHDS, U_a is the attribute universe which contains arbitrary string, U_A is the authority universe with N different AAs and a public function $F : U_a \longrightarrow U_A$, which maps each attribute $j \in U_a$ to a specific authority $\mathrm{AA}_i \in U_A$, denoting that the attribute j is managed by authority AA_i, and I_{AA} is the index of relevant authorities of a user. For simplicity, here we introduce another symbol $\delta(j) = F(j), j \in U_a$.

5.2. Construction of EMA-LUPHDS. Here, the detail of each phase and corresponding algorithms in the formal definition of our proposal are given.

5.2.1. Initialization Phase. In this phase, TA generates system global public parameters and master key and each AA generates their public and secret key pair by the following steps.

(i) $Setup_{Global}(\lambda)$. Given the security parameter λ, TA generates groups G and G_T of prime order p with a bilinear map $\hat{e} : G \times G \longrightarrow G_T$. Then, it chooses random generators $g, g_1, g_2 \in G$ and four collision-resistant hash functions $H : \{0,1\}^* \longrightarrow Z_p^*, H_0 : U_a \longrightarrow G, H_1 : \{0,1\}^* \longrightarrow G, H_2 : G_T \longrightarrow \{0,1\}^{n_1}, H_3 : \{0,1\}^* \longrightarrow \{0,1\}^{n_2}$, where M is the message universe and n_1 and n_2 are output sizes of H_2 and H_3 hash functions, respectively. Next, TA creates a $\mathscr{L}$-length key derivation function (KDF) K, where $\mathscr{L} = |\mathrm{key}| + |p|$ and set the global public parameters as follows:

$$\mathrm{pp} = \{G, G_T, H, H_0, H_1, H_2, H_3, F, K, \mathscr{L}, \hat{e}, g, g_1, g_2\}. \quad (1)$$

Finally, TA publishes the global public parameters pp.

(ii) $Setup_{AA}(pp)$. Each AA AA_i manages a set of attributes S_{AA_i}. As to each attribute authority AA_i, it chooses two random number $y_i, \alpha_i, \beta_i, t_i \in Z_p{}^*$ for itself. Thus, each attribute authority AA_i generates its key pair as follows:

$$\mathrm{sk}_{\mathrm{AA}_i} = (y_i, \alpha_i, \beta_i, t_i), \mathrm{pk}_{\mathrm{AA}_i} = \left(g^{y_i}, g^{\alpha_i}, g^{\beta_i}\right). \tag{2}$$

Finally, the attribute authority AA_i outputs their public and secret key pair $(\mathrm{pk}_{\mathrm{AA}_i}, \mathrm{sk}_{\mathrm{AA}_i})$.

5.2.2. User Enrollment Phase. Upon receiving the enrollment request from DU with their global identities and attribute sets, attribute authorities generate a secret key and a transformation key pair for DU based on the following algorithms.

(i) *KeyGen*$(pp, sk_{AA_i}, GID, S_{i,GID})$. If a DU has a global identity GID and a set of attributes $S_{i,\mathrm{GID}}$ which is related to an attribute authority AA_i, the AA_i chooses a random number $t \in Z_p^*$ and computes the secret key $\mathrm{sk}_{i,\mathrm{GID}}$ for the DU as follows:

$$\begin{aligned}\mathrm{sk}_{i,\mathrm{GID}} &= \left\{K_{1,j}, K_{2,j}, K_3\right\}_{j\in S_{i,\mathrm{GID}}} \\ &= \left\{g^{\alpha_i} H_1(\mathrm{GID})^{y_i} H_0(j)^{\beta_i}, H_0(j)^{t_i}, g^{\beta_i}\right\}_{j\in S_{i,\mathrm{GID}}}.\end{aligned} \tag{3}$$

Finally, the attribute authority AA_i outputs $\mathrm{sk}_{i,\mathrm{GID}}$ and sends it to the DU identified by GID through secure channel.

(ii) *TKeyGen*$(pp, sk_{i,GID})$. The authority AA_i generates transformation key for DU identified by GID on giving the DU's secret key $\mathrm{sk}_{i,\mathrm{GID}}$. We assume that as for each attribute $j \in S_{i,\mathrm{GID}}$, if $F(j) = i$, the attribute set $S_{i,\mathrm{GID}}$ of DU with GID is managed by AA_i. The authority AA_i chooses a random number $\mu \in Z_p^*$ and computes the transformation key $\mathrm{tk} = (\mathrm{tpk}, \mathrm{tsk})$ as follows:

$$\begin{aligned}\mathrm{tpk}_{i,\mathrm{GID}} &= \left\{\mathrm{TK}_{1,j}, \mathrm{TK}_{2,j}, \mathrm{TK}_3, \mathrm{TK}_4, \mathrm{TK}_5\right\}_{j\in S_{i,\mathrm{GID}}} \\ &= \left\{K_{1,j}^{1/\mu}, K_{2,j}^{1/\mu}, K_3^{1/\mu}, g^{1/\mu}, H_1(\mathrm{GID})^{1/\mu}\right\}_{j\in S_{i,\mathrm{GID}}} \\ &= \left\{g^{\alpha_i/\mu} H_1(\mathrm{GID})^{y_i/\mu} H_0(j)^{\beta_i/\mu}, H_0(j)^{t_i/\mu}, g^{\beta_i/\mu}, g^{1/\mu}, H_1(\mathrm{GID})^{1/\mu}\right\}_{j\in S_{i,\mathrm{GID}}} \\ \mathrm{tsk}_{i,\mathrm{GID}} &= \{\mu\}.\end{aligned} \tag{4}$$

Finally, AA_i outputs transformation key $\mathrm{tk} = (\mathrm{tpk}, \mathrm{tsk})$ for DU with identity GID.

5.2.3. Encryption Phase. On input globally public parameters pp and public key of AA_i, the encryption process contains the following three steps:

(i) *Encrypt*$_{off}(pp)$. DO selects a random secret $s \in_R Z_p$ to compute the encapsulated key key. Then, the DO generates the corresponding session key ssk for data encryption/decryption and the commitment $\bar{C}$ (e.g., Pedersen commitment algorithm) for key verification. The algorithm is executed as follows:

$$\begin{aligned}\mathrm{key} &= \prod_{i\in I_A A} \hat{e}(g, g)^{\alpha_i s} = \hat{e}(g, g)^{\sum_{i\in I_{AA}} \alpha_i s}, \\ \mathrm{KDF}(\mathrm{key}, \mathscr{L}) &= \mathrm{ssk} \| d, \bar{C} = g_1^{H(\mathrm{ssk})} g_2^{H(d)}.\end{aligned} \tag{5}$$

As a result, the DO sets $\mathrm{Key}_{\mathrm{off}} = (s, \mathrm{key}, \mathrm{ssk}, \bar{C})$ and creates a pool of offline keys. Next, the DO picks a random element $R \in G_T$ and computes $R' = H_2(R)$. Finally, the DO sets $\mathrm{VC}_{\mathrm{off}} = (R, R')$ and constructs a pool of offline verification code.

(ii) *APTransform*. To achieve access policy anonymity, the DO first designates an original access policy (A_o, ρ), where A_o is a $l \times n$ matrix and ρ is a function that maps each row of A_o to an attribute. Then, the DO selects a random value $a \in Z_p{}^*$ and computes $m_x = \hat{e}((g^{t_i})^a, H_0(\rho(x)))$, where $x \in [l]$ is each row of access policy (A_o, ρ) and l is the number of rows in A_o. To preserve the privacy of access policy, the data owner replaces each attribute $\rho(x)$ in A_o with m_x, and then, the original access policy (A_o, ρ) can be transformed to LSSS access policy matrix $(A_{l\times n}, \rho)$ which can be denoted by (A, ρ) for simplicity.

(iii) *Encrypt*$_{on}(pp, M, (A, \rho))$. DO chooses any one pair of offline components $\mathrm{Key}_{\mathrm{off}} = (s, \mathrm{key}, \mathrm{ssk}, \bar{C})$ and $\mathrm{VC}_{\mathrm{off}} = (R, R')$ to encrypt the data M gathered from smart devices to generate encryped data C $\mathrm{T}_s = E_{\mathrm{symm}}(M, \mathrm{ssk})$ with the symmetric encyption algorithm and the symmetric key ssk and compute the verification code $V_m = H_3(R' \| \mathrm{CT}_s)$ for CT_s. With the specific access policy (A, ρ), where A is a $l \times n$ matrix and ρ is a mapping from each row A_x to a certain attribute att_x, DO picks $p_x \in_R Z_p{}^*$ for each row A_x of A and computes $\lambda_x = A_x$

$\cdot \gamma$, where $\gamma = \{s, \gamma_2, \cdots, \gamma_l\} \in_R {Z_p}^l$ and $\theta_x = A_x \cdot \nu$, where $\nu = \{0, \nu_2, \cdots, \nu_l\} \in_R {Z_p}^l$. Next, DO outputs ciphertext $\mathrm{CT} = \{(A, \rho), \bar{C}, \mathrm{CT}_s, V_m, C', C_0, \{C_{1,x}, C_{2,x}, C_{3,x}, C_{4,x}\}_{x \in [l]}\}$, where

$$\begin{aligned} C' &= g^a, C_0 = R \cdot \mathrm{key}, C_{1,x} = g^{\alpha_{\delta(x)} \lambda_x} g^{\alpha_{\delta(x)} p_x}, \\ C_{2,x} &= g^{p_x}, C_{3,x} = (H_0(\rho(x)))^{p_x}, C_{4,x} = g^{y_{\delta_x} p_x} g^{\theta_x}. \end{aligned} \tag{6}$$

Finally, the DO uploads the ciphertext CT to CSP.

5.2.4. Decryption Phase. After DU requesting for specific data CT with his transformation public key (tpk), the CSP executes outsourced access policy recovery operations and sends back the intermediate anonymous attributes to the DU. Then, the CSP executes outsourced decryption with the recovered index of access policy received from the DU. Next, with the partially decrypted ciphertext from CSP, the DU can get decrypted plaintext at the end. Finally, with verification execution, the DU can verify whether the ciphertext and session key is valid. The phase involves the following algorithms:

(i) *APRecover*. First of all, the CSP computes ${m_x}' = \hat{e}(C', H_0(\rho(x))^{t_i/\mu})$, where $x \in S_{i,\mathrm{GID}}$ with the DU's tpk and sends the ${m_x}'$ together with access policy of CT, i.e., (A, ρ), to the DU. Then, the DU replaces the attribute x with $({m_x}')^{\mathrm{tsk}}$, and the result attribute set ${S_{i,\mathrm{GID}}}'$ is constructed, and the attribute index set is ${I_s}' = \{x : (\rho(x) \bigcap {S_{\mathrm{GID}}}')_{x \in [l]}\}$. Then, the DU sends ${I_s}'$ to CSP for outsourced decryption.

(ii) $Decrypt_{\mathrm{OUT}}(pp, tpk_{i,GID}, CT)$. Let each matrix row A_x of access policy (A, ρ) correspond to an attribute $\rho(x)$, and CSP executes as follows:

$$\begin{aligned} C^*_{\delta(x)} &= \frac{\hat{e}(\mathrm{TK}_4, C_{1,x}) \hat{e}(\mathrm{TK}_5, C_{4,x}) \hat{e}(\mathrm{TK}_3, C_{3,x})}{\hat{e}(\mathrm{TK}_{1,x}, C_{2,x})} \\ &= \frac{\hat{e}\left(g^{1/\mu}, C_{1,x}\right) \hat{e}\left(H_1(\mathrm{GID})^{1/\mu}, C_{4,x}\right)}{\hat{e}\left(g^{\alpha_{\delta(x)}/\mu} H_1(\mathrm{GID})^{y_{\delta(x)}/\mu} H_0(\rho(x))^{\beta_{\delta(x)}/\mu}, C_{2,x}\right)} \\ &\quad \cdot \frac{\hat{e}\left(g^{\beta_{\delta(x)}/\mu}, H_0(\rho(x))^{p_x}\right)}{\hat{e}\left(g^{\alpha_{\delta(x)}/\mu} H_1(\mathrm{GID})^{y_{\delta(x)}/\mu} H_0\right)\left(\rho(x)^{\beta_{\delta(x)}/\mu}, C_{2,x}\right)} \\ &= \left(e\wedge(g, g)^{\alpha_{\delta(x)} \lambda_x} e\wedge(H_1(\mathrm{GID}), g)^{\theta_x}\right)^{1/\mu}. \end{aligned} \tag{7}$$

Then, as mentioned before, $\lambda_x = A_x \cdot \gamma$, where $\gamma = \{r, \gamma_2, \cdots, \gamma_l\} \in {Z_p}^l$ and $\theta_x = A_x \cdot \nu$, where $\nu = \{0, \nu_2, \cdots, \nu_l\} \in {Z_p}^l$, we note that there exists coefficients $\omega_x \in Z_p$ where $x \in {I_s}'$ such that $\sum_{x \in {I_s}'} \omega_x A_x = (1, 0, \cdots, 0)$. Thus, we have $\sum_{x \in {I_s}'} \omega_x \lambda_x = s$ and $\sum_{x \in {I_s}'} \omega_x \theta_x = 0$.

Subsequently, the DU can computes $\prod_{x \in {I_s}'} (C^*)^{\omega_x}$, so that,

$$\begin{aligned} {\mathrm{CT}_{\delta(x)}}' &= \prod_{x \in I_s'} \left(C^*_{\delta(x)}\right)^{\omega_x} = \prod_{x \in I_s'} \left(e\wedge(g, g)^{\alpha_{\delta(x)} \omega_x \lambda_x} e\wedge(H_1(\mathrm{GID}), g)^{\omega_x \theta_x}\right)^{1/\mu} \\ &= \left(e\wedge(g, g)^{\alpha_{\delta(x)} \sum_{x \in {I_s}'} \omega_x \lambda_x}\right)^{1/\mu} \cdot \left(e\wedge(H_1(\mathrm{GID}), g)^{\sum_{x \in {I_s}'} \omega_x \theta_x}\right)^{1/\mu} \\ &= \hat{e}(g, g)^{\alpha_{\delta(x)} s/\mu}. \end{aligned} \tag{8}$$

Let $i = \delta(x)$, and we have the following equation:

$$\mathrm{CT}' = \prod_{i \in I_{\mathrm{AA}}} \mathrm{CT}_i' = \prod_{i \in I_{\mathrm{AA}}} \hat{e}(g, g)^{\alpha_i s/\mu} = \hat{e}(g, g)^{\sum_{i \in I_{\mathrm{AA}}} \alpha_i s/\mu}. \tag{9}$$

Finally, the CSP returns partially decrypted ciphertext $\mathrm{CT}^* = \{C_0, \bar{C}, \mathrm{CT}', \mathrm{CT}_s, V_m\}$ to the DU.

(i) $Decrypt_U(pp, tsk_{i,GID}, CT^*)$. After receiving the partially decrypted ciphertext CT^*, the DU recovers the random element R and the encapsulated key key used for generating symmetric session key as follows:

$$\begin{aligned} \mathrm{key}^* &= \left(\mathrm{CT}'\right)^{\mathrm{tsk}} = \left(e\wedge(g, g)^{\sum_{i \in I_{\mathrm{AA}}} \alpha_i s/\mu}\right)^{\mu} = \hat{e}(g, g)^{\sum_{i \in I_{\mathrm{AA}}} \alpha_i s}, \\ R^* &= \frac{C_0}{\mathrm{key}^*} = \frac{R \cdot \hat{e}(g, g)^{\sum_{i \in I_{\mathrm{AA}}} \alpha_i s}}{\hat{e}(g, g)^{\sum_{i \in I_{\mathrm{AA}}} \alpha_i s}}. \end{aligned} \tag{10}$$

Finally, DU outputs the recovered random element R^* and encapsulated key key*.

(ii) $DecVerify(pp, \bar{C}, V_m, key^*, R^*)$. On input recovered encapsulated key key* and random element R^*, the DU computes as follows:

$$\begin{aligned} \mathrm{KDF}(\mathrm{key}^*, \mathscr{L}) &= \mathrm{ssk}^* \| d^*, \bar{C}^* = g_1^{H(\mathrm{ssk}^*)} g_2^{H(d^*)}, \\ \bar{R}^* &= H_2(R^*), V_m^* = H_3\left(\bar{R}^* \| \mathrm{CT}_s\right). \end{aligned} \tag{11}$$

Then, the DU checks if the following equations

$$\begin{aligned} \bar{C} &\overset{?}{=} \bar{C}^*, \\ V_m &\overset{?}{=} V_m^*. \end{aligned} \tag{12}$$

hold, and it outputs $M = D_{\mathrm{symm}}(\mathrm{CT}_s, \mathrm{ssk}^*)$, otherwise $\perp$.

6. Security Analysis

In this section, we present a brief security analysis of our proposed EMA-LUPHDS scheme concerning the design goals mentioned in Section 4.2.

Theorem 1. *The proposed scheme satisfies the properties of correctness.*

Proof. We can prove the correctness of outsourced decryption in our scheme by the following equation:

$$\begin{aligned}
\mathrm{CT}_{\delta(x)}{}' &= \prod_{x=1}^{l}\left(C^{*}_{\delta(x)}\right)^{\omega_x} = \prod_{x=1}^{l}\left(\frac{\widehat{e}(\mathrm{TK}_4, C_{1,x})\widehat{e}(\mathrm{TK}_5, C_{4,x})\widehat{e}(\mathrm{TK}_3, C_{3,x})}{\widehat{e}(\mathrm{TK}_{1,x}, C_{2,x})}\right)\\
&= \prod_{x=1}^{l}\left(\frac{e\wedge\left(g^{1/\mu}, C_{1,x}\right)e\wedge\left(H_1(\mathrm{GID})^{1/\mu}, C_{4,x}\right)}{e\wedge\left(g^{\beta_{\delta(x)}/\mu}H_1(\mathrm{GID})^{y_{\delta(x)}/\mu}H_0(\rho(x))^{\beta_{\delta(x)}/\mu}, C_{2,x}\right)}\right.\\
&\quad\left.\cdot\frac{e\wedge\left(g^{\beta_{\delta(x)}/\mu}, H_0(\rho(x))^{p_x}\right)}{e\wedge\left(g^{\alpha_{\delta(x)}/\mu}H_1(\mathrm{GID})^{y_{\delta(x)}/\mu}H_0(\rho(x))^{\beta_{\delta(x)}/\mu}, C_{2,x}\right)}\right)^{\omega_x}\\
&= \prod_{x=1}^{l}\left(\frac{e\wedge\left(g^{1/\mu}, g^{\alpha_{\delta(x)}\lambda_x}g^{\alpha_{\delta(x)}p_x}\right)}{e\wedge\left(g^{\alpha_{\delta(x)}/\mu}H_1(\mathrm{GID})^{y_{\delta(x)}/\mu}H_0(\rho(x))^{\beta_{\delta(x)}/\mu}, g^{p_x}\right)}\right.\\
&\quad\cdot\frac{e\wedge\left(H_1(\mathrm{GID})^{1/\mu}, g^{y_{\delta_x}p_x g^{\theta_x}}\right)}{e\wedge\left(g^{\alpha_{\delta(x)}/\mu}H_1(\mathrm{GID})^{y_{\delta(x)}/\mu}H_0(\rho(x))^{\beta_{\delta(x)}/\mu}, g^{p_x}\right)}\\
&\quad\left.\cdot\frac{e\wedge\left(g^{\beta_{\delta(x)}/\mu}, H_0(\rho(x))^{p_x}\right)}{e\wedge\left(g^{\alpha_{\delta(x)}/\mu}H_1(\mathrm{GID})^{y_{\delta(x)}/\mu}H_0(\rho(x))^{\beta_{\delta(x)}/\mu}, g^{p_x}\right)}\right)^{\omega_x}\\
&= \prod_{x=1}^{l}\left(e\wedge(g, g)^{\alpha_{\delta(x)}\lambda_x}e\wedge(H_1(\mathrm{GID}), g)^{\theta_x}\right)^{\omega_x/\mu}\\
&= \prod_{x=1}^{l}\left(e\wedge(g, g)^{\alpha_{\delta(x)}\omega_x\lambda_x}e\wedge(H_1(\mathrm{GID}), g)^{\omega_x\theta_x}\right)^{1/\mu}\\
&= \left(e\wedge(g, g)^{\alpha_{\delta(x)}\sum_{x=1}^{l}\omega_x\lambda_x}e\wedge(H_1(\mathrm{GID}), g)^{\sum_{x=1}^{l}\omega_x\theta_x}\right)^{1/\mu}\\
&= \widehat{e}(g, g)^{\alpha_{\delta(x)}s/\mu}.
\end{aligned}\tag{13}$$

□

Theorem 2. *The proposed scheme satisfies the properties of data confidentiality.*

Proof. In our scheme, the data is first encrypted using a symmetric encryption algorithm, and the key is encapsulated by access policy. As for the data confidentiality, the symmetric encryption algorithm, such as AES, can guarantee the feature. With respect to the fine-grained data access control, for the transformation public key tpk of a unauthorized DU whose attribute set does not satisfy the access policy, CSP cannot get an authorized index set $I_s{}'$ so as to calculate the correct constants $\{\omega_x\}$ to make the equation $\sum_{x\in I_s{}'}\omega_x A_x = s$ holds. Thus, the CSP will fail to return a correct partially decrypted ciphertext, and the DU also cannot obtain the encapsulated symmetric key to further get the plaintext of data. Moreover, in outsourced decryption, the CSP also cannot get the symmetric key from partially decrypted ciphertext to recover plaintext of data because it cannot get the transformation secret key tsk of the DU to further decrypt the partially decrypted ciphertext. Furthermore, the secret key of each DU is embedded with his unique global identity, and the transformation public key of each DU is also confused with his unique transformation secret key tsk which is secret by the DU himself, and any two or more DUs have no way to collude for data access. □

Theorem 3. *The proposed scheme satisfies the properties of access policy hiding.*

Proof. In our scheme, when the DO encrypts the symmetric key used in symmetric encryption based on a designated access policy, he first transforms each attribute in access policy according to the one-way anonymous key agreement protocol in [52] by computing $m_x = \widehat{e}((g^{t_i})^a, H_0(\rho(x)))$ for each row x of access policy, where a is a random number. Then, DO replaces each attribute in access policy by m_x, which can obfuscate each attribute $\rho(x)$ in access policy. In decryption phase, the DU cannot compute m_x only if he has the key component $H_0(\rho(x))^{t_i}$. Otherwise, DU cannot distinguish m_x from x. Therefore, malicious DU cannot infer the access policy, and thus, the attribute information in access policy is protected. □

Theorem 4. *The proposed scheme satisfies the properties of collusion resistance.*

Proof. The malicious users may collude to combine their secret keys and transformation keys to access the shared data which they cannot access individually. In our scheme, different attribute authority generates secret keys for different users, and the secret keys are associated with users' GID, specific attribute set and random, which are uniquely related to each user and make the combination of attributes in different secret keys useless. As a result, collusive users cannot compute $\mathrm{key}^* = \widehat{e}(g, g)^{\sum_{i\in I_{\mathrm{AA}}}\alpha_i s}$ cooperatively in the outsourced decryption even if the combined attributes of these users satisfy the access policy. Thus, our scheme is collision-resistant. □

Theorem 5. *The proposed scheme satisfies the properties of verifiability.*

Proof. Suppose that KDF is secure and H, H_2, and H_3 are three collision-resistant hash functions. Thus, the output of KDF is indistinguishable from a random string. In the encryption phase of our scheme, $\mathrm{KDF}(\mathrm{key}, \mathscr{L}) = \mathrm{ssk}\|d$ and $\bar{C} = g_1^{H(\mathrm{ssk})}g_2^{H(d)}$. As it is difficult to distinguish the output of KDF from a random string and H is a deterministic collision-resisstant hash function, the untrusted CSP has no way to guess the random $H(d), H(\mathrm{ssk})$ and thus fails to tamper the Pedersen commitment $\bar{C}$ which is computationally hiding. Moreover, since H_2 and H_3 are two collision-resistant functions, it is hard to guess a random R^* to construct $H_3(R^*\|\mathrm{CT}_s) = H_3(R\|\mathrm{CT}_s)$, which is in negligible probability. Therefore, the validity of key and ciphertexts CT_s can be guaranteed. □

Theorem 6. *The proposed scheme satisfies the properties of exculpability.*

Proof. Suppose that KDF is secure and H is a deterministic collision-resistant hash functions. Thus, the output of KDF is indistinguishable from a random string. If a malicious DU with transformation secret key tsk wants to accuse CSP of returning incorrect results, he has to have the ability of forging a fake transformation secret key tsk* that can generate the same commitment. Suppose that $g_1 = g^{\varphi}, g_2 = g^{\psi}$ and the malicious DU constructs $\mathrm{KDF}(\mathrm{key}, \mathscr{L}) = \mathrm{ssk}\|d$ and $\mathrm{KDF}(\mathrm{key}^*, \mathscr{L}) = \mathrm{ssk}^*\|d^*$, where key and key* are partially decrypted results with tsk and tsk*, respectively. The commitment must be equal, that is, $g_1^{H(\mathrm{ssk})} g_2^{H(d)} = g_1^{H(\mathrm{ssk}^*)} g_2^{H(d^*)}$. Then, the malicious DU can get $\varphi = \psi(H(d^*) - H(d))/H(\mathrm{ssk}) - H(\mathrm{ssk}^*)$, which means that the malicious DU can solve DL problem. However, it is of negligible probability according to DL assumption. Therefore, our scheme is exculpable for decryption. □

7. Performance Evaluation

This section evaluates the performance by comparing our EMA-LUPHDS scheme with several existing schemes in efficiency aspects. We give the comparison in computation and space complexity in theoretical aspects between our scheme and the schemes in [6, 7]. Furthermore, we focus on experiment implementation to precisely evaluate the efficiency of EMA-LUPHDS. By comparing with several excellent similar schemes, we demonstrate that our scheme is more efficient and practicable for data sharing in MCC.

7.1. Theoretical Analysis. We thoroughly analyzes the computation and space complexity by comparing our EMA-LUPHDS and other schemes [2, 6, 7] in detail from the aspects of public parameter size (pp size), user key size (UKey size), transformation key size (TKey size), ciphertext size (ciphertext size), encryption cost, user decryption cost, and outsourced decryption cost (out decryption cost), as the former four metrics measure the space complexity of each scheme and the remains are used to evaluate the computation cost in execution of each scheme. The comparison result is summarized in Table 2.

Here, we first stipulate some denotions in the theoretical analysis. E_G, E_{G_T} denotes the exponentiation operations in G, G_T, M_G, M_{G_T} denotes the multiplication operations in G, G_T, P denotes the pairing operation $\hat{e}$, H denotes the computation cost of a hash function and $\mathrm{Enc}_{\mathrm{sym}}, \mathrm{Dec}_{\mathrm{sym}}$ denotes the computation costs of symmetric encryption and decryption. In addition, l denotes the number of attributes in access structure, $|S|$ denotes the number of attributes owned by DU, $|G|, |G_T|$ denotes the length of elements in group G, G_T, $|A|$ denotes the number of attributes managed by each authority, $|V|$ denotes the length of verification code, and $|\mathrm{CT}_{\mathrm{sym}}|$ denotes the length of symmetric encrypted ciphertext.

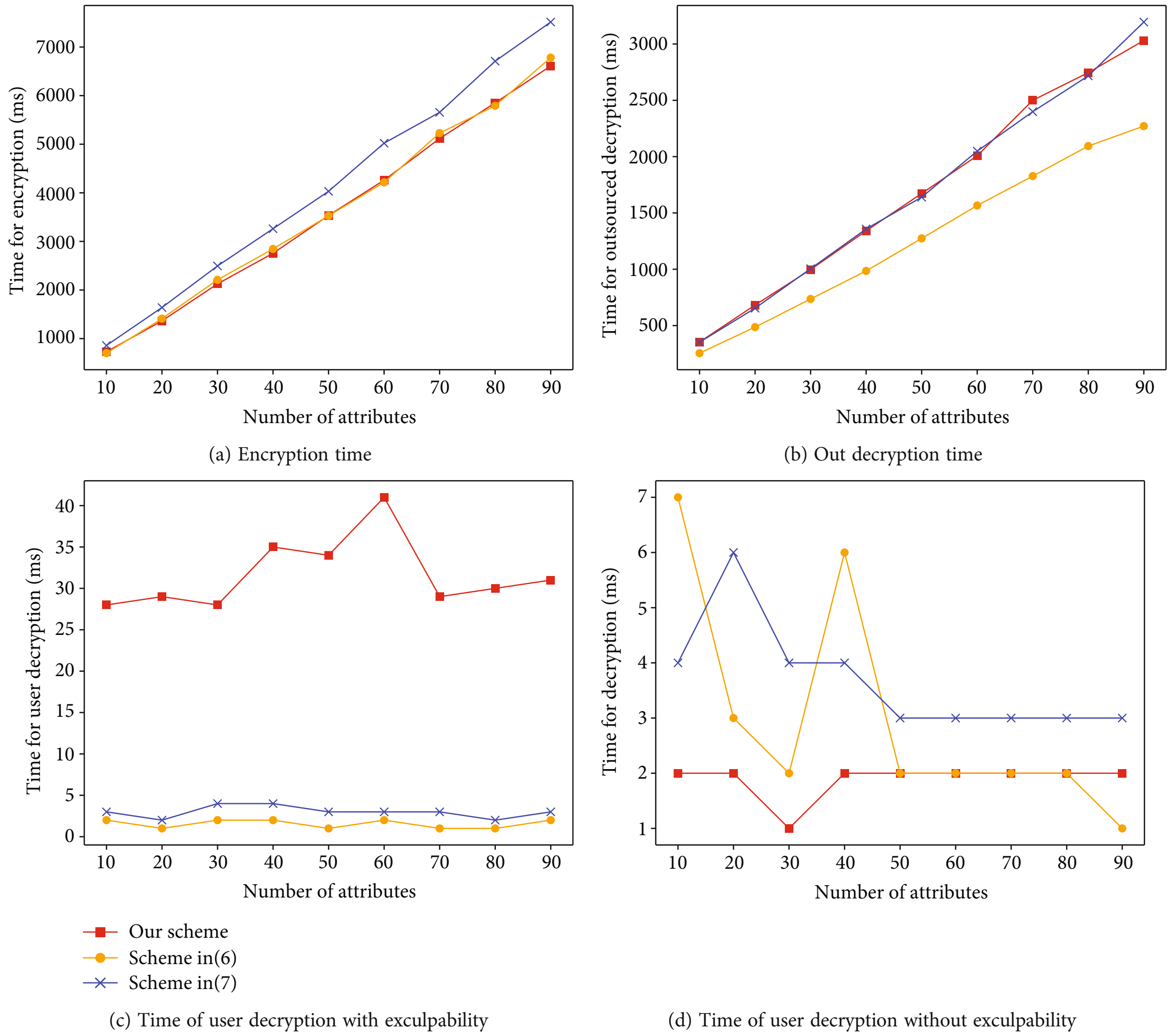
In Table 2, we first analyze the space complexity comparison. First of all, the pp size of schemes in [2, 6, 7] are $|G| + 1 + (2 + |G_T| + |G|)|A|$, $|A|(|G| + 1) + |G_T| + 1$, and $|G| + |G_T| + 2 + (3 + |G_T| + |G|)|A|$, respectively. We note that these sizes are all growing with the increase of access policy number. However, the pp size in our scheme is $3|G| + 4$, which shows that our scheme can support large attribute universe because the public parameter size is constant and very small. Moreover, the transformation key sizes of the four schemes are $|G|(2 + |S|)$, $2|G|(|S| + 1)$, $2|G|(|S| + 3/2) + 1$, and $2|G|(|S| + 1) + 1$, respectively. This means that the transformation key sizes in four related schemes are of the same case. Furthermore, we analyze the ciphertext size. In schemes [2, 7], the ciphertext sizes are $(3|G| + 4)l + |G| + |G_T|$ and $|G_T| + |V| + |G| + (3|G| + |G_T|)l$, while the sizes in our scheme and the scheme [6] are $2|G|(1 + 2l) + |\mathrm{CT}_{\mathrm{sym}}| + |V| + |G_T|$ and $|G|(3l + 1) + |G_T| + |V| + |\mathrm{CT}_{\mathrm{sym}}|$. We note that the former two schemes support smaller ciphertext. However, as the latter two schemes are suitable for scalable plaintext encryption, the ciphertext size may be larger. Later, we will analyze the experiment result and use the base ciphertext size to compare the practical result.

Then, we analyze the computation complexity comparison. First, as for encryption time, the complexity in scheme is [2] while in our scheme and the schemes [6, 7] are $(5E_G + 2M_G)l + E_G + P + M_{G_T} + H + \mathrm{Enc}_{\mathrm{sym}}$, $2M_{G_T} + H + E_G + (5E_G + 2M_G + E_{G_T})l$, and $E_G + M_{G_T} + (6E_G + 2M_G)l + \mathrm{Enc}_{\mathrm{sym}} + H$. We can infer that the computation complexity in [2] is a little less while the other three schemes cost more. Moreover, the user decryption in schemes [2, 7] is $E_{G_T} + M_{G_T}$ and $E_{G_T} + M_{G_T} + H$, while the schemes in [6] and our scheme cost $E_{G_T} + M_{G_T} + \mathrm{Dec}_{\mathrm{sym}} + 2H$ and $E_{G_T} + M_{G_T} + \mathrm{Dec}_{\mathrm{sym}} + 2H + 2E_G + M_G$. We note that the latter two schemes cost more than the former two schems because the latter two schemes support large plaintext encryption and decryption, which means that the user needs to decrypt the symmetric ciphertext after obtaining encapsulated symmetric key. In our scheme, we need more computation for commitment recover and add more computation overhead. Furthermore, as for out decryption, we infer from Table 2 that the four schemes outsource similar workload to third party.

In conclusion, we know that although in our scheme, the transformation key is a little larger than other schemes in Table 2, and it has far smaller public parameters in constant size. Also, our scheme supports scalable ciphertext though it may take up a lot of space. As our scheme supports flexible functions, to increase the efficiency, we also introduce online/offline and outsourced computing techniques. We note that from Table 2, the computation cost in encryption and user decryption of our scheme is greatly reduced and approaches other schemes in Table 2. In general, our scheme can achieve more reasonable computation complexity compared with other relevant schemes in theoretical analysis.

Table 2: Computation complexity comparison.

Scheme	pp size	TKey size	Ciphertext size	Encryption cost	User decryption cost	Our decryption cost
Scheme in [6]	$\lvert G\rvert + 1 + (2 + \lvert G_T\rvert + \lvert G\rvert)\lvert A\rvert$	$2\lvert G\rvert\lvert S\rvert$	$\lvert G\rvert(3l+1) + \lvert G_T\rvert + \lvert V\rvert + \lvert \mathrm{CT}_{\mathrm{sym}}\rvert$	$(5E_G + 2M_G)l + E_G + P + M_{G_T} + H + \mathrm{Enc}_{\mathrm{sym}}$	$E_{G_T} + M_{G_T} + \mathrm{Dec}_{\mathrm{sym}} + 2H$	$\left(3P + 2M_{G_T} + E_{G_T}\right)\lvert S\rvert$
Scheme in [2]	$\lvert A\rvert + (\lvert G\rvert + 1) + \lvert G_T\rvert + 1$	$\lvert G\rvert(2 + \lvert S\rvert)$	$(3\lvert G\rvert + 4)l + \lvert G\rvert + \lvert G_T\rvert$	$M_{G_T} + E_{G_T} + E_G + 2l$	$E_{G_T} + M_{G_T}$	$\left(2E_G + 3M_{G_T}\right)\lvert S\rvert + P + \left(2P + M_{G_T} + E_{G_T}\right)\lvert S\rvert$
Scheme in [7]	$\lvert G\rvert + \lvert G_T\rvert + 2 + (3 + \lvert G_T\rvert + \lvert G\rvert)\lvert A\rvert$	$2\lvert G\rvert(\lvert S\rvert + 1) + 1$	$\lvert G_T\rvert + \lvert V\rvert + \lvert G\rvert + (3\lvert G\rvert + \lvert G_T\rvert)l$	$2M_{G_T} + H + E_G + \left(5E_G + 2M_G + E_{G_T}\right)l$	$E_{G_T} + M_{G_T} + H$	$\left(4P + 3M_{G_T} + 2E_{G_T}\right)\lvert S\rvert + P + M_{G_T} + H$
Our scheme	$3\lvert G\rvert + 4$	$2\lvert G\rvert\left(\lvert S\rvert + \frac{3}{2}\right) + 1$	$2\lvert G\rvert(1 + 2l) + \lvert \mathrm{CT}_{\mathrm{sym}}\rvert + \lvert V\rvert + \lvert G_T\rvert$	$E_G + M_{G_T} + (6E_G + 2M_G)l + \mathrm{Enc}_{\mathrm{sym}} + H$	$E_{G_T} + M_{G_T} + \mathrm{Dec}_{\mathrm{sym}} + 2H + 2E_G + M_G$	$\left(4P + 3M_{G_T} + E_{G_T}\right)\lvert S\rvert$



(a) Encryption time

(b) Out decryption time

(c) Time of user decryption with exculpability

(d) Time of user decryption without exculpability

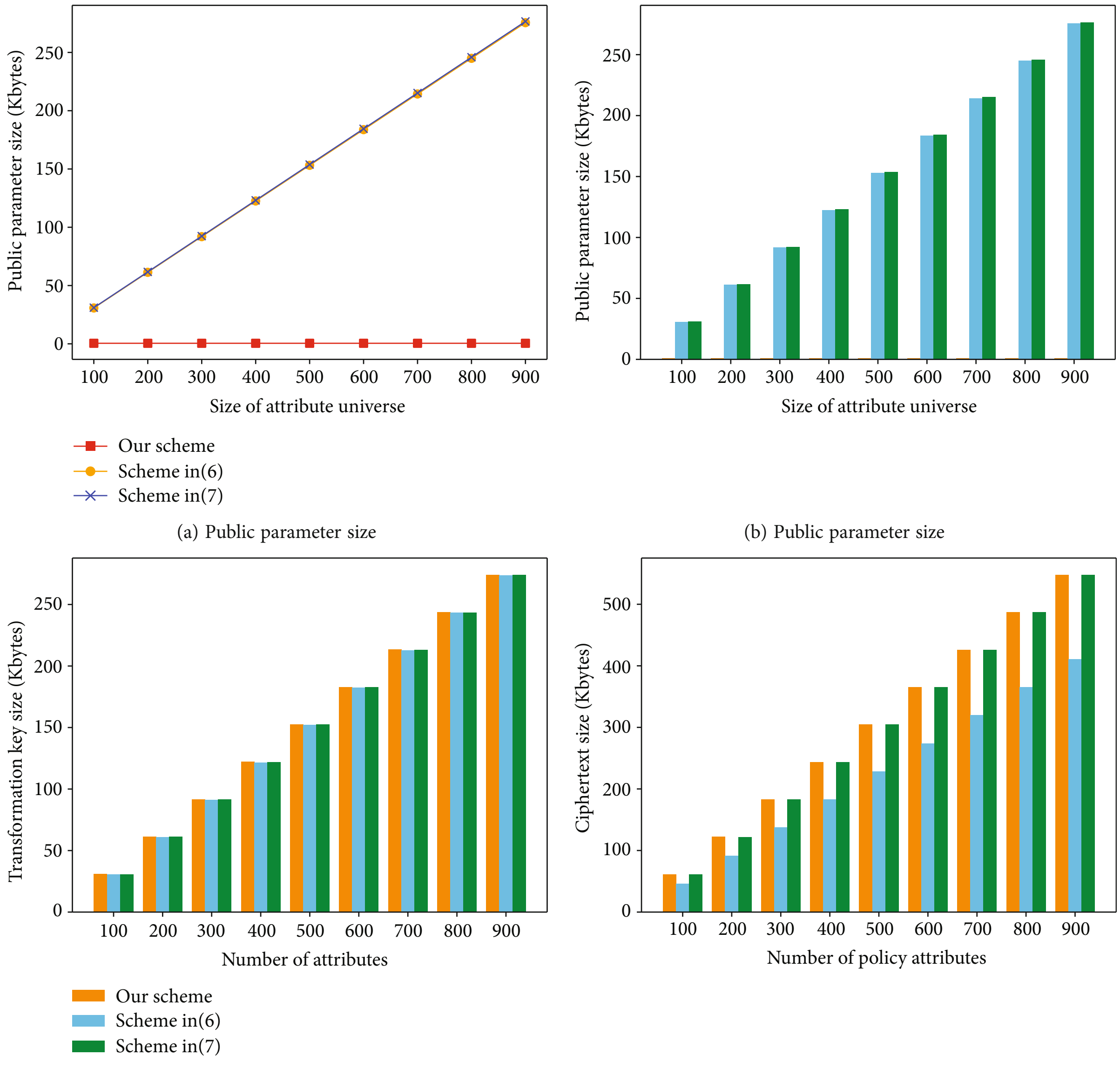
Figure 3: Comparison of the computation cost.

7.2. Experimental Analysis. To precisely evaluate the performance of EMA-LUPHDS, we implement our scheme and the schemes in [6, 7] and compare their actual computation and space cost with EMA-LUPHDS, and the result of which is summarized in Figures 3 and 4.

We implement and develop these schemes using Java Programming Language with the Java Pairing-Based Cryptography library (JPBC) [53] for various operations in finite field and groups. Type A pairing is adopted in our implementations which is defined over a 160-bit elliptic curve group over 512-bit finite field, that is, the supersingular elliptic curve $E(F_p)$: $y^2 = x^3 + x$ with embedding degree 2, where p is a 512-bit Solinas prime. Moreover, our simulation experiments are run on Windows10 system with Intel Core i5 CPU 2.13 GHz and 8.00 GB RAM. In addition, we use SHA256 algorithm to generate the V_m for correctness verification of ciphertext in our experiments.

Figure 3 shows the computation comparison from the point of the time cost in encryption, outsourced decryption, and user decryption. We note that in Figure 3(a), our scheme performs approximate to that of schemes in [6] and is superior to the scheme in [7] in encryption. From Figure 3(b), we know that the computation cost of outsourced decryption for our scheme is a little larger than that of [6] and nearly the same as that of [7]. Figures 3(c) and 3(d) present the computation cost of Pedersen commitment for supporting exculpability. We note that in Figure 3(d), the three schemes perform similarly, and in Figure 3(c), the computation cost of our scheme is larger than the other two schemes, which shows the trade-off between the function of exculpability and efficiency cost.

From Figure 4, we note that the storage complexity of our scheme is approximate to that in [6, 7] while takes only constant-sized public parameters that are far smaller than that in [6, 7]. We can infer from Figure 4(a) that the size of public parameters in our scheme is very small and constant. Thus, in Figure 4(b), the public parameter size of our scheme is nearly invisible. In Figure 4(c), we



(a) Public parameter size

(b) Public parameter size

(c) Transformation key size

(d) Ciphertext size

Figure 4: Comparison of the storage cost.

know that the three schemes take up similar size in transformation key. Figure 4(d) shows that our scheme takes up a little larger space for ciphertext as we support exculpability and flexible policy hiding. We also note that the ciphertext size is approximate to that of the scheme in [7] which is not flexible as our scheme. And both the scheme in and our scheme can support scalable ciphertext, which means that the user does not need to map plaintext to the bilinear group.

It is obvious that the results of our experiment simulation indicate that our scheme is flexible and versatile. It is also efficient in encryption cost, user decryption cost, and out decryption cost and has far smaller and constant public parameter size. Therefore, we argue that EMA-LUPHDS proposed in our work is more suitable for resource-constraint mobile devices in MCC system.

8. Conclusion

In this paper, we propose an Efficient and Multiauthority Large Universe Policy-Hiding Data Sharing (EMA-LUPHDS) scheme to achieve key escrow resistance, expressive access policies without user privacy leakage, and high efficiency for data sharing of MCC with resource-limited mobile devices. In our proposal, we adopt fully hidden strategy to protect sensitive information about attributes of users and access policy. To achieve high efficiency, we introduce outsourced decryption to reduce the computational cost and the online/offline technique to trade off the overhead in encryption operation. In addition, we add into the ciphertext with verification code and Pedersen commitment to ensure the correctness of the partially decrypted result got from misbehaving CSP and the exculpability for CSP accused by DU maliciously. Moreover, the security analysis and thorough performance evaluation show that our proposal is practicable for resource-restraint mobile devices in the MCC system.

In our future work, we would dedicate into the efficient attribute and user revocation in data sharing scheme for mobile cloud environment.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research is funded by the National Natural Science Foundation of China (Nos. 62072093, 62072092, 61601107, and U1708262), the China Postdoctoral Science Foundation (No. 2019M653568), the Fundamental Research Funds for the Central Universities (No. N2023020), and the Natural Science Foundation of Hebei Province of China (No. F2020501013).

References

[1] S. J. De and S. Ruj, “Efficient decentralized attribute based access control for mobile clouds,” *IEEE Transactions on Cloud Computing*, vol. 8, no. 1, pp. 124–137, 2020.

[2] M. Lyu, X. Li, and H. Li, “Efficient, verifiable and privacy preserving decentralized attribute-based encryption for mobile cloud computing,” in *2017 IEEE Second International Conference on Data Science in Cyberspace (DSC)*, pp. 195–204, Shenzhen, China, 2017.

[3] Y. Wu, X. Wang, W. Susilo et al., “Efficient server-aided secure two-party computation in heterogeneous mobile cloud computing,” *IEEE Transactions on Dependable and Secure Computing*, p. 1, 2021.

[4] H. Riasudheen, K. Selvamani, S. Mukherjee, and I. Divyasree, “An efficient energy-aware routing scheme for cloud-assisted MANETs in 5G,” *Ad Hoc Networks*, vol. 97, p. 102021, 2020.

[5] Y. Xie, H. Wen, B. Wu, Y. Jiang, and J. Meng, “A modified hierarchical attribute-based encryption access control method for mobile cloud computing,” *IEEE Transactions on Cloud Computing*, vol. 7, no. 2, pp. 383–391, 2019.

[6] S. Belguith, N. Kaaniche, M. Laurent, A. Jemai, and R. Attia, “Phoabe: securely outsourcing multi-authority attribute based encryption with policy hidden for cloud assisted iot,” *Computer Networks*, vol. 133, pp. 141–156, 2018.

[7] K. Fan, H. Xu, L. Gao, H. Li, and Y. Yang, “Efficient and privacy preserving access control scheme for fog-enabled IoT,” *Future Generation Computer Systems*, vol. 99, pp. 134–142, 2019.

[8] N. Zhang, Q. Jiang, L. Li, X. Ma, and J. Ma, “An efficient three-factor remote user authentication protocol based on BPV-FourQ for internet of drones,” *Peer-to-Peer Networking and Applications*, 2021.

[9] J. Zhang, T. Li, M. S. Obaidat, C. Lin, and J. Ma, “Enabling efficient data sharing with auditable user revocation for IoV systems,” *IEEE Systems Journal*, pp. 1–12, 2021.

[10] K. Zhang, H. Li, J. Ma, and X. Liu, “Efficient large-universe multiauthority ciphertext-policy attribute-based encryption with white-box traceability,” *Science China Information Sciences*, vol. 61, no. 3, 2018.

[11] P. Zhang, Z. Chen, J. K. Liu, K. Liang, and H. Liu, “An efficient access control scheme with outsourcing capability and attribute update for fog computing,” *Future Generation Computer Systems*, vol. 78, pp. 753–762, 2018.

[12] J. Li, F. Sha, Y. Zhang, X. Huang, and J. Shen, “Verifiable outsourced decryption of attribute-based encryption with constant ciphertext length,” *Security and Communication Networks*, vol. 2017, Article ID 3596205, 11 pages, 2017.

[13] J. Li, X. Chen, S. S. M. Chow, Q. Huang, D. S. Wong, and Z. Liu, “Multi-authority fine-grained access control with accountability and its application in cloud,” *Journal of Network and Computer Applications*, vol. 112, pp. 89–96, 2018.

[14] H. Ma, R. Zhang, Z. Wan, Y. Lu, and S. Lin, “Verifiable and exculpable outsourced attribute-based encryption for access control in cloud computing,” *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 6, pp. 679–692, 2017.

[15] A. Lewko and B. Waters, “Decentralizing attribute-based encryption,” in *Advances in Cryptology – EUROCRYPT 2011*, vol. 6632 of Lecture Notes in Computer Science, pp. 568–588, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[16] V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-based encryption for fine-grained access control of encrypted data,” in *Proceedings of the 13th ACM conference on Computer and communications security - CCS '06*, pp. 89–98, New York, NY, USA, 2006.

[17] A. Sahai and B. Waters, “Fuzzy identity-based encryption,” in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 457–473, Springer, Berlin, Heidelberg., 2005.

[18] D. Wang and P. Wang, “Two birds with one stone: two-factor authentication with security beyond conventional bound,” *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 1–722, 2016.

[19] Z. Li, D. Wang, and E. Morais, “Quantum-safe round-optimal password authentication for mobile devices,” *IEEE Transactions on Dependable and Secure Computing*, 2020.

[20] S. Qiu, D. Wang, G. Xu, and S. Kumari, “Practical and provably secure three-factor authentication protocol based on extended chaotic-maps for mobile lightweight devices,” *IEEE Transactions on Dependable and Secure Computing*, 2020.

[21] C. Wang, D. Wang, Y. Tu, G. Xu, and H. Wang, “Understanding node capture attacks in user authentication schemes for wireless sensor networks,” *IEEE Transactions on Dependable and Secure Computing*, 2020.

[22] Y. Rouselakis and B. Waters, “Practical constructions and new proof methods for large universe attribute-based encryption,” in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security - CCS '13*, pp. 463–474, New York, NY, USA, 2013.

[23] G. Zhao, Q. Jiang, X. Huang, X. Ma, Y. Tian, and J. Ma, “Secure and usable handshake based pairing for wrist-worn smart devices on different users,” *Mobile Networks and Applications*, 2021.

[24] Q. Jiang, N. Zhang, J. Ni, J. Ma, X. Ma, and K.-K. R. Choo, "Unified biometric privacy preserving three-factor authentication and key agreement for cloud-assisted autonomous vehicles," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 9390–9401, 2020.

[25] J. Zhang, J. Ma, Z. Ma et al., "Efficient hierarchical data access control for resource-limited users in cloud-based e-health," in *2019 International Conference on Networking and Network Applications (NaNA)*, pp. 319–324, Daegu, Korea (South), 2019.

[26] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attributebased encryption," in *2007 IEEE Symposium on Security and Privacy (SP '07)*, pp. 321–334, Berkeley, CA, USA, 2007.

[27] M. Chase, "Multi-authority attribute based encryption," in *Theory of Cryptography Conference*, pp. 515–534, Springer, 2007.

[28] M. Chase and S. S. Chow, "Improving privacy and security in multiauthority attribute-based encryption," in *Proceedings of the 16th ACM conference on Computer and communications security - CCS '09*, pp. 121–130, New York, NY, USA, 2009.

[29] M. Green, S. Hohenberger, and B. Waters, "Outsourcing the decryption of abe ciphertexts," *USENIX Security Symposium*, vol. 2011, no. 3, 2011.

[30] K. Yang and X. Jia, "Expressive, efficient, and revocable data access control for multi-authority cloud storage," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 7, pp. 1735–1744, 2014.

[31] K. Yang and X. Jia, "Attributed-based access control for multi-authority systems in cloud storage," in *2012 IEEE 32nd International Conference on Distributed Computing Systems*, pp. 536–545, Macau, China, 2012.

[32] J. Lai, R. H. Deng, C. Guan, and J. Weng, "Attribute-based encryption with verifiable outsourced decryption," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 8, pp. 1343–1354, 2013.

[33] Baodong Qin, R. H. Deng, Shengli Liu, and Siqi Ma, "Attribute-based encryption with efficient verifiable outsourced decryption," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 7, pp. 1384–1393, 2015.

[34] S. Hohenberger and B. Waters, "Attribute-based encryption with fast decryption," in *Public-Key Cryptography – PKC 2013*, pp. 162–179, Springer, Berlin, Heidelberg, 2013.

[35] S. Ruj, A. Nayak, and I. Stojmenovic, "DACC: distributed access control in clouds," in *2011IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications*, pp. 91–98, Changsha, China, 2011.

[36] K. Yang, X. Jia, K. Ren, and B. Zhang, "DAC-MACS: effective data access control for multi-authority cloud storage systems," in *2013 Proceedings IEEE INFOCOM*, pp. 2895–2903, Turin, Italy, 2013.

[37] S. Ruj, M. Stojmenovic, and A. Nayak, "Decentralized access control with anonymous authentication of data stored in clouds," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 2, pp. 384–394, 2014.

[38] S. Hohenberger and B. Waters, "Online/offline attribute-based encryption," in *Public-Key Cryptography – PKC 2014*, vol. 8383, pp. 293–310, Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[39] P. Datta, R. Dutta, and S. Mukhopadhyay, "Fully secure online/offline predicate and attribute-based encryption," in *Information Security Practice and Experience*, vol. 9065, pp. 331–345, Springer International Publishing, 2015.

[40] S. Even, O. Goldreich, and S. Micali, "On-line/off-line digital signatures," *Journal of Cryptology*, vol. 9, no. 1, pp. 35–67, 1996.

[41] A. Shamir and Y. Tauman, "Improved online/offline signature schemes," in *Advances in Cryptology — CRYPTO 2001*, pp. 355–367, Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[42] F. Guo, Y. Mu, and Z. Chen, "Identity-Based Online/Offline Encryption," in *Financial Cryptography and Data Security*, vol. 5143 of Lecture Notes in Computer Science, pp. 247–261, 2008.

[43] T. Nishide, K. Yoneyama, and K. Ohta, "Attribute-based encryption with partially hidden encryptor-specified access structures," in *Applied Cryptography and Network Security*, pp. 111–129, Springer, Berlin, Heidelberg, 2008.

[44] J. Lai, R. H. Deng, and Y. Li, "Fully Secure Cipertext-Policy Hiding CP-ABE," in *Information Security Practice and Experience vol. 6672*, pp. 24–39, Springer, Berlin, Heidelberg, 2011.

[45] J. Lai, R. H. Deng, and Y. Li, "Expressive CP-ABE with partially hidden access structures," in *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security - ASIACCS '12*, ACM: New York, 2012.

[46] Y. Zhang, X. Chen, J. Li, D. S. Wong, and H. Li, "Anonymous attributebased encryption supporting efficient decryption test," in *Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security - ASIA CCS '13*, pp. 511–516, ACM: New York, 2013.

[47] H. Cui, R. H. Deng, G. Wu, and J. Lai, "An efficient and expressive ciphertext-policy attribute-based encryption scheme with partially hidden access structures," in *Provable Security, vol. 10005*, pp. 19–38, Springer, Cham, 2016.

[48] H. Cui, R. H. Deng, J. Lai, X. Yi, and S. Nepal, "An efficient and expressive ciphertext-policy attribute-based encryption scheme with partially hidden access structures, revisited," *Computer Networks*, vol. 133, pp. 157–165, 2018.

[49] Z. Zhou, D. Huang, and Z. Wang, "Efficient privacy-preserving ciphertext-policy attribute based-encryption and broadcast encryption," *IEEE Transactions on Computers*, vol. 64, no. 1, pp. 126–138, 2015.

[50] T. V. X. Phuong, G. Yang, and W. Susilo, "Hidden ciphertext policy attribute-based encryption under standard assumptions," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 1, pp. 35–45, 2016.

[51] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Advances in Cryptology — CRYPTO 2001*, pp. 213–229, Springer, 2001.

[52] A. Kate, G. M. Zaverucha, and I. Goldberg, "Pairing-based onion routing," in *Privacy Enhancing Technologies*, pp. 95–112, Springer, Berlin, Heidelberg, 2007.

[53] A. De Caro and V. Iovino, "JPBC: Java pairing based cryptography," in *2011 IEEE Symposium on Computers and Communications (ISCC)*, pp. 850–855, Kerkyra, Greece, 2011.