A Practical Attribute-based Encryption with Silent Revocation and Exposure Resistance for Cloud Storage

Journal:	*Transactions on Dependable and Secure Computing*
Manuscript ID	TDSC-2023-05-0463
Manuscript Type:	Regular Paper
Keywords:	Attribute-based Encryption, Silent Revocation, Cloud Storage, Original Ciphertext Exposure, Decryption Key Exposure

A Practical Attribute-based Encryption with Silent Revocation and Exposure Resistance for Cloud Storage

Lukai Qin, Shuanggen Liu*, Xiaoqing Liu, Jindong Yu, Xu An Wang, Teng Wang

***Abstract*—Conventional revocable attribute-based encryption (ABE) generally requires the frequent generation and distribution of update material by the revocation list maintainer, long-term online reception and storage of key update materials by the unrevoked users, and periodic overall or instant individual ciphertext updates by the cloud server. These mechanisms pose a substantial communication, computation, and storage burden to all system entities, especially in large-scale file storage and frequent attribute update scenarios. Therefore, this paper proposes a CP-ABE scheme with silent revocation (CP-ABE-SR) which only requires the trusted authority to update the user attribute list silently in a revocation event by leveraging a user qualification check protocol in the ciphertext query phase. To improve the scheme's security, we considered decryption key exposure, original ciphertext exposure, and limited identity exposure and proved it IND-CPA secure in the random oracle model. Analysis and implementation indicate that the scheme makes a transition from an active re-encryption revocation mechanism centered on the invalidation of revoked users' decryption keys to a passive re-encryption mechanism centered on the invalidation of the exposed original ciphertexts, resulting in a significant reduction in computational cost from periodic revocation-related variables to minor request-related constants for both the cloud server and the trusted authority.**

***Index Terms*—Original Ciphertext Exposure, Decryption Key Exposure, Silent Revocation, Attribute-based Encryption, Cloud Storage**

I. Introduction

CLOUD storage has become a prevailing data hosting tendency among enterprises and individual users, thanks to its low management costs and convenient sharing features. However, as data is generally hosted by third-party, this inevitably raises concerns about security and privacy. Encryption is a crucial technique that helps safeguard data confidentiality in clouds. Recently, ABE [1] has gained significant popularity as a widely adopted encryption technique in cloud data sharing, enabling fine-grained sharing and ensuring data confidentiality.

In a common ABE, users' credentials are linked to attribute sets, and access policies are integrated into the ciphertext. The decryption of the target ciphertext is only possible for users meeting the access policy requirement. Furthermore, revocable ABE schemes have been proposed to support user or attribute-level revocation, which is critical in scenarios with dynamically changing user information. Revocable ABE can be categorized as either direct or indirect revocation. The latter requires a trusted authority rather than the data owner to manage user revocation, making it more suitable for cloud data-sharing scenarios. In indirect revocable ABE, an independent, trusted authority manages and maintains a revocation list and computes and distributes update materials periodically or instantaneously with the overhead linear to the frequency of revocation events. Similarly, cloud servers can perform periodically or instantaneously overall ciphertext update or individual ciphertext update upon request, with the overhead increasing linearly with the number of ciphertexts or user requests. Compared with basic ABE, the computational cost for the authority and the cloud server increase markedly but is generally acceptable.

However, conventional revocable schemes become infeasible in scenarios with large-scale file storage and frequent attribute updates, such as changes in location, speed, and time [2]. The overall ciphertext update mechanism in the cloud server is inappropriate for such scenarios owing to the overhead increasing linearly with the number of ciphertexts. Hence, the individual ciphertext update mechanism appears to be a solid choice. For the trusted authority, in periodic revocation, the update interval must be shorter than the minimum interval between revocation events to ensure that all revocation events are processed. A shorter update interval would lead to a significant increase in computation and communication overhead. Instantaneous revocation also suffers from similar excessive overhead. Therefore, to adapt to extreme scenarios, seeking a new revocation mechanism is necessary to make the computation overhead of the trusted authority's update materials independent of the frequency of revocation events.

In addition, various exposure attacks pose threats to revocable ABE schemes. The issue of decryption key exposure (DKE) [3] has received widespread attention. In general revocable ABE schemes, if an adversary obtains a user's decryption key for a non-challenge period, i.e., $dk_{ID^*,T}$, where $T \neq T^*$, and the public key update material ku_T, it may recover the (partial) long-term key sk_{ID^*} and subsequently compute the challenge decryption key dk_{ID^*,T^*}. Furthermore, we call attention to the potential threat of original ciphertext exposure (OCE) attacks. Specifically, suppose a revoked user retains the attribute key that satisfies the challenge access policy.

*Corresponding author.

Lukai Qin, Xiaoqing Liu, and Jindong Yu are with the School of Cyberspace Security, Xi'an University of Posts and Telecommunications. E-mail: {1061646401, liuxiaoqing, yujindong}@stu.xupt.edu.cn

Shuanggen Liu, Teng Wang are with Xi'an University of Posts and Telecommunications. E-mail: {liushuanggen201, wangteng}@xupt.edu.cn

Xu An Wang is with Engineering University of People's Armed Police. E-mail: wangxazjd@163.com

If the attacker compromises the cloud server by exploiting backdoors or vulnerabilities and directly obtains the original challenge ciphertext, it may decrypt it trivially. Alternatively, if the ciphertext update material is public or user-generated, the attacker can update it to the version that matches the retained decryption key and decrypt it. The original ciphertext exposure attack is a real threat to cloud servers using individual ciphertext update mechanisms, which may be feasible, especially for a revoked former employee of a cloud service provider. Moreover, authorized users' personal identity exposure is the most common information leakage. If an attacker acquires the authorized user's private identity identifier, they can impersonate the user. Therefore, providing as strong as possible resistance to limited identity exposure is also desirable.

A. Our contribution

Motivated by the practical issues that existing revocation schemes pose significant computation and transmission burden to all system entities compared to basic ABE schemes, it is necessary to innovate a new revocation mechanism to minimize revocation events' overhead. Therefore, this paper proposes a practical LSSS-based CP-ABE scheme with silent revocation and multiple exposure attack resistance. In summary, our contributions are as follows.

- **Silent revocation:** Conventional revocation mechanisms will bring considerable computation and transmission overhead to the cloud server and the trusted authority, especially in large-scale file storage and frequent user and attribute update scenarios. Therefore, we present the notion of silent revocation which only requires the trusted authority to update the user attribute list silently in a revocation event. The mechanism realizes user and attribute revocation simultaneously.
- **Exposure resistance:** Besides forward and backward secrecy and collusion resistance, revocable ABE has been threatened by various potential exposure attacks, which brings higher security requirements for revocable ABE. We made a tradeoff between computing efficiency and exposure resistance so that the scheme can resist decryption key exposure, original ciphertext exposure, and limited identity exposure.
- **Efficient construction:** We construct the CP-ABE-SR scheme based on an efficient LSSS-based CP-ABE scheme and prove it to be IND-CPA secure under the decisional Modified-BDHE assumption in the random oracle. We further conduct theoretical analysis and benchmark to evaluate the performance of our proposed scheme to demonstrate its applicability to the target scenario.

B. Related work

ABE. Since introduced by Sahai et al. [1], ABE has evolved into two distinct types, namely CP-ABE [4] and KP-ABE [5]. KP-ABE grants access control qualifications to the key issuer, making it suitable for fine-grained user permission management by resource providers. CP-ABE grants access control qualifications to data owners, making it suitable for constructing egalitarian fine-grained data-sharing platforms. ABE has found widespread use in cloud computing [6], e-health [7], and IoT [8], with several variants having been developed.

Revocable ABE. As a widely studied variant of ABE [2], revocable ABE aims to satisfy practical needs for user revocation and attribute changes. Alongside the proposal of IBE, Boldyreva et al. [9] incorporated an expiration date in the public key to address the issue of revocation. Later, Pirretti et al. [10] embedded the expiration date in attributes, necessitating periodic key redistribution for all users by the authority. Ibraimi et al. [11] first implemented an instantaneous revocation mechanism, and the key is split and stored separately by the user and a third-party server. Hur et al. [12] used a binary tree to manage user groups, achieving indirect user attribute revocation. Many revocable ABE schemes discussed applications in cloud storage [13]–[15], e-health [16], [17], and traitor tracing [18], [19]. However, the computation overhead for updating materials in previous indirect revocation schemes was linear to the frequency of revocation events, making them unsuitable for scenarios with frequent user and attribute revocations.

Decryption key exposure. Since captured by Seo et al. [3], decryption key exposure in IBE settings has received significant attention [20]–[22]. To address this issue in the ABE setting, Cui et al. [23] introduced an auxiliary server and proposed an outsourced revocable ABE scheme. Qin et al. [24] addressed the issue of server compromise by dividing the decryption key into long-term and short-term keys within the same framework. Recently, Wei et al. [16] integrated the delegation function into their revocable ABE scheme while addressing the DKE issue. Moreover, some works [25]–[27] addressed the decryption key exposure issue in the IoT environment. However, revocable ABE is actually threatened by various exposure attacks, such as original ciphertext exposure and identity exposure, which were not covered in previous literature.

II. Preliminaries

A. Bilinear pairing

Let $\mathbb{G}$ and $\mathbb{G}_T$ be two multiplicative groups of the same prime order p, and g is the generator of $\mathbb{G}$. A bilinear mapping is a mapping $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ with the following properties.

1) Bilinearity: $\forall u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p, e\left(u^a, v^b\right) = e(u, v)^{ab}$.
2) Non-Degeneracy: $e(g, g) \neq 1$, which means $e(g, g)$ is the generator of group $\mathbb{G}_T$
3) Computability: $\forall u, v \in \mathbb{G}$, there exist efficient algorithms to compute $e(u, v)$

B. Access structure

Let $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$ be a set of parties. A collection $\mathbb{A} \subseteq 2^{\mathcal{P}}$ is monotone if $B \in \mathbb{A}$ and $B \subseteq C$, then $C \in \mathbb{A}$. An access structure is a monotone collection $\mathbb{A}$ of all non-empty subsets of $\mathcal{P}$, that is, $\mathbb{A} \subseteq 2^{\mathcal{P}} \backslash \{\emptyset\}$. The sets in $\mathbb{A}$ are called the authorized sets, and the sets not in $\mathbb{A}$ are called the unauthorized sets.

C. Linear secret sharing schemes

The linear secret sharing scheme (LSSS) consists of two algorithms:

- Share: Consider a vector $v = (s, y_2, \ldots, y_n)^T$, where $s \in Z_p$ is the secret and $y_2, \ldots, y_n \in Z_p$ are randomly chosen, then $\lambda_i = M_i \cdot v$ is a share of the secret s corresponding to the attribute name by $\rho(i)$, where M_i is the i-th row of M.
- Reconstruction: Let $S \in \mathbb{A}$ be any authorized set and $I = \{i : \rho(i) \in S\} \subseteq \{1, 2, \ldots, l\}$. Then there exist coefficients $\{\omega_i \mid i \in I\}$ such that $\sum_{i \in I} \omega_i M_i = (1, 0, \ldots, 0)$, and thus, we have $\sum_{i \in I} \omega_i \lambda_i = s$.

The security proof needs the following proposition.

Proposition 1 [28]**.** A vector $\vec{v}$ is independent of a set of vectors represented by a matrix M if and only if there exists a vector $\vec{w}$ such that $M\vec{w} = \overrightarrow{0}$ while $\vec{v} \cdot \vec{w} \neq 0$.

D. Modified-BDHE assumption

Let $(p, \mathbb{G}, \mathbb{G}_T, e)$ be a bilinear pairing. Choose a, $t, s, q \in \mathbb{Z}_p$, a generator $g \in \mathbb{G}$. Given $\vec{Y} =$

$$g, g^s, g^{s(at+a)}, g^a, \ldots, g^{a^q}, g^{a^{q+2}}, \ldots, g^{a^{2q}},$$
$$g^{at}, g^{a^2 t}, \ldots, g^{a^q t}, g^{a^{q+2} t}, \ldots, g^{a^{2q} t}$$

it must remain hard to distinguish between $T = e(g,g)^{a^{q+1}s} \in \mathbb{G}_T$ and $T = R \in \mathbb{G}_T$. Formally, the advantage of an adversary $\mathcal{A}$ in solving Modified-BDHE problem [29]

$$\left| \Pr\left[\mathcal{A}\left(\vec{Y}, T = e(g,g)^{a^{q+1}s}\right) = 0\right] - \Pr[\mathcal{A}(\vec{Y}, T = R) = 0] \right|$$

is negligible.

III. System Architecture and Definitions

A. System architecture

The CP-ABE with silent revocation (CP-ABE-SR) framework, as illustrated in Fig. 1, comprises the data owner, the trusted authority, the cloud server, and the data user.

- **Trusted Authority:** TA issues private keys for participants and generates public parameters. It controls the user attribute list, the encryption description list, and the external key generator. TA only updates the local user attribute list without external coordination in a user/attribute revocation event. When TA receives a ciphertext query request from DU, it verifies whether DU's latest attribute set satisfies the access policy requirements of the ciphertext. If the verification is successful, TA computes the external key of the ciphertext and returns a temporary decryption key to DU while transmitting the random factor to CS through a secret channel. Otherwise, TA returns invalid to CS.
- **Data Owner:** DO requests an external key from TA, encrypts the data under an access policy and the external key, then uploads the resulting ciphertext to CS, and the encryption description item to TA, respectively.
- **Cloud Server:** CS stores the ciphertext and forwards the user's ciphertext query request to TA for executing user authentication. If TA verifies that DU is a legitimate user, it will return a temporary random factor, which CS will use to update the original request ciphertext and return a temporary ciphertext to DU. Otherwise, CS will reject the user's query request. CS is assumed curious but honest.
- **Data User:** DU has the private attribute key and two semi-functional identities dedicated to KeyGen and CT-Query phases. Before decryption, DU must obtain the temporary decryption key and the temporary ciphertext if the user passes the qualification check as a legitimate user in the ciphertext query phase.

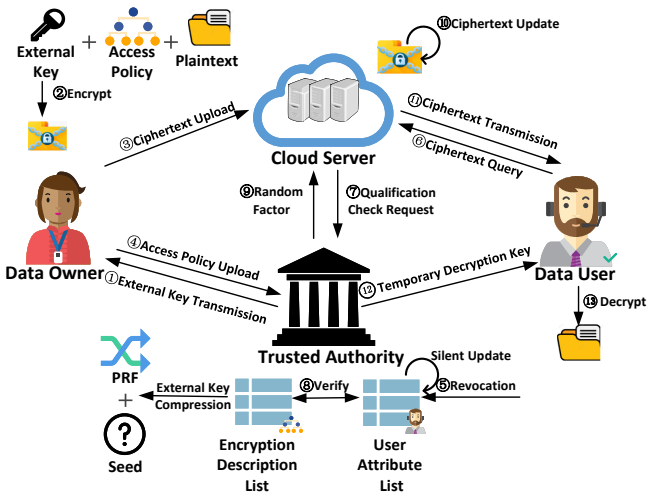


Fig. 1. System architecture of our scheme.

B. Threat model

We take into account the following threats:

- **Collusion attack:** Revoked users[1] and unauthorized users[2] may collude to retrieve and decrypt the ciphertext they can't individually access.
- **Forward and backward secrecy:** After a revocation, some revoked users may try to access the previously or subsequently published ciphertexts to which they no longer have access rights.
- **Decryption key exposure:** Users may leak their temporary decryption keys to revoked or unauthorized users, which they leverage to decrypt the target ciphertext.
- **Original ciphertext exposure:** Some malicious users may compromise the cloud server by exploiting vulnerabilities or backdoors and directly retrieving the target ciphertext.
- **Limited identity exposure:** Authorized users may leak their CTQuery ID or KeyGen ID (not both) to revoked users or unauthorized users.

We assume that the cloud server and the revoked user do not conclude, as they can trivially win if they both secretly retain and share the historical query results for the target ciphertext. It should be noted that the original ciphertext exposure attack is a

[1]Revoked users refer to users who lose access to a target ciphertext due to user or attribute revocation.

[2]Unauthorized users refer to users who have never obtained enough access privileges to a target ciphertext.

compromise attack, not a collusion attack, because the cloud server does not assist the attacker in obtaining the exposed ciphertext and would erase the temporary random factor and updated ciphertext.

C. CP-ABE with Silent Revocation (CP-ABE-SR)

Definition 1(CP-ABE-SR). The scheme comprises the following algorithms.

- $\mathrm{Setup}(\lambda, \mathcal{S}) \rightarrow \{\mathrm{MSK}, \mathrm{param}\}$: Using the security parameter λ and attribute universe $\mathcal{S}$ as input,TA executes Setup to generates the system's public parameter param and master secret key MSK.
- $\mathrm{KeyGen}(u_k, \mathcal{S}(u), \mathrm{MSK}, \mathrm{param}) \rightarrow \{d_u\}$: Using a KenGen identity u_k, an attribute set $\mathcal{S}(u)$, master secret key MSK and param as input, TA executes KeyGen to generates the attribute key d_u.
- $\mathrm{Encrypt}(\mathcal{M}, u_k, u_q, c, \beta, \mathrm{MSK}, \mathrm{param}) \rightarrow \{ct, \mathcal{I}_c\}$: Using a message $\mathcal{M}$, a KeyGen identity u_k, a CTQuery identity u_q, a ciphertext identifier c, an access policy β, the master secret key MSK and param as input, DO executes Encrypt to generate the ciphertext ct and its encryption description item $\mathcal{I}_c$.
- $\mathrm{CTQuery}(c, u_q, ct, \mathcal{S}(u), \mathcal{I}_c, \mathrm{MSK}) \rightarrow \{(ct', k')/\perp\}$: When a user u sends a query (c, u_q) to CS for the ciphertext ct, CS and TA execute a qualification check protocol jointly. If TA verifies that u's latest attributes $\mathcal{S}(u)$ satisfy c's access policy $\mathbb{A}(c)$, then CS and TA separately return the temporary ciphertext ct' and the temporary decryption key k' to user u. Otherwise, CS returns $\perp$.
- $\mathrm{Decrypt}\,(ct', d_u, k', \mathrm{param}) \rightarrow \{\mathcal{M}/\perp\}$: Using the temporary ciphertext ct', the temporary decryption key k', the attribute key d_u, and param as input, if $\mathcal{S}(u) \models \mathbb{A}$, DU executes Decrypt to obtain the message $\mathcal{M}$.

D. IND-CPA security model

As the same probability distribution of the original ciphertext and the temporary ciphertext which is unavailable for revoked users and unauthorized users, we only capture the security of the original ciphertext in CP-ABE-SR.

Init. $\mathcal{A}$ outputs a challenge ciphertext identifier c^*, a challenge access policy $\mathbb{A}^*$, a challenge KeyGen identity u_k^*, and a challenge CTQuery identity u_q^*.

Setup. $\mathcal{C}$ executes $\mathrm{Setup}\left(1^\lambda, \mathcal{S}\right)$ to obtain a master key MSK and param. Return param to $\mathcal{A}$. Set the revoked user list $\mathcal{R}$ to empty.

Query phase 1.

- Key extraction query $\mathcal{O}_{sk}(u_k, \mathcal{S}(u))$: For an KeyGen identity u_k and an attribute set $\mathcal{S}(u)$, the challenger $\mathcal{C}$ runs algorithm $\mathrm{KeyGen}(u_k, \mathcal{S}(u), \mathrm{MSK}, \mathrm{param})$ and returns d_u to adversary $\mathcal{A}$. If $\mathcal{S}(u) \models \mathbb{A}^*$, append the user u to the revoked user list $\mathcal{R}$.
- External key query $\mathcal{O}_k(u_k, u_q, c)$: For a KeyGen identity u_k and a ciphertext identifier c. $\mathcal{C}$ checks whether $u_k = u_k^*$ and $u_q = u_q^*$ and $c = c^*$. If so, $\mathcal{C}$ returns $\perp$. Else, $\mathcal{C}$ computes $k = H(u_k||u_q||c||\gamma)$ and returns k to adversary $\mathcal{A}$.

Challenge. $\mathcal{A}$ outputs two message $\mathcal{M}_0^*, \mathcal{M}_1^*$ with equal length. Next, $\mathcal{C}$ randomly samples $b \in \{0, 1\}$ and executes $\mathrm{Encrypt}\left(\mathcal{M}_b^*, u_k^*, u_q^*, c^*, \mathbb{A}^*, \mathrm{MSK}, \mathrm{param}\right)$ to obtain ct^*. Finally, $\mathcal{C}$ outputs ct^*.

Query phase 2.

- Key extraction query $\mathcal{O}_{sk}(u_k, \mathcal{S}(u))$
- External key query $\mathcal{O}_k(u_k, u_q, c)$
- Decryption key query $\mathcal{O}_{k'}(u)$: For any identity u exposed to $\mathcal{A}$, $\mathcal{C}$ runs $\mathrm{CTQuery}(u_q, c^*, ct^*, \mathcal{S}(u), \mathcal{I}_{c^*}, \mathrm{MSK})$. If the user u passes the qualification check, $\mathcal{C}$ outputs the temporary decryption key k', else outputs $\perp$.

Guess. $\mathcal{A}$ outputs $b' \in \{0, 1\}$.

Denote the advantage of $\mathcal{A}$ by

$$\mathrm{Adv}_{\mathcal{A}}^{\text{CP-ABE-SR}} = |\Pr[b = b'] - \frac{1}{2}|.$$

Definition 2 (sIND-CPA security). A CP-ABE-SR scheme is indistinguishable from chosen-plaintext attacks under a selective access policy if all poly-time adversaries have at most negligible advantage in the above game.

Remark. In the above security game, in addition to obtaining the attribute keys and the full identities (u_q, u_k) of colluding users, $\mathcal{A}$ can directly obtain the original challenge ciphertext, thereby defining the original ciphertext exposure. In the decryption key query, $\mathcal{A}$ can get a temporary decryption key k' of $ct^{*\prime}$ using an exposed identity u, thus defining the (temporary) decryption key exposure.

IV. Construction

A. Overview

The CP-ABE-SR scheme adopts the efficient ABE scheme [29]. In brief, the underlying ABE is built upon [30, Sec. 3.5], which optimizes the decryption algorithm of the CP-ABE proposed by Hohenberger et al. [31, Sec. 5], and incorporates new conditional encryption and decryption algorithm. In Encrypt algorithm, a boolean formula is first transformed into Disjunctive Normal Form (DNF). The clauses' number of DNF m is then compared with the size of the original boolean formula l. If m is smaller, proceed with the extended encryption and decryption algorithm. Otherwise, revert to the algorithm in [31].

In order to minimize the computation and transmission overhead associated with revocation, our scheme employs a silent, passive malice prevention strategy. Specifically, the silent revocation mechanism comprises ciphertext access control set up on CS and user qualification checks executed by TA. During system initialization, TA establishes the user attribute list and the encryption description list. In a revocation event, TA only silently updates the user attribute list without external coordination. After encryption, the data user uploads the encryption description item to the encryption description list. To obtain the ciphertext, all users must pass the user qualification check in the CTQuery phase. Therefore, revoked and unauthorized users can never successfully request ciphertext from CS.

We assume that CTQuery is a vulnerable algorithm. That is, the malicious revoked user may circumvent the normal

ciphertext query procedure and directly obtain the original target ciphertext. To resist the original ciphertext exposure, we introduce external encryption on top of the basic ABE, and only the data owner and TA share the external key k. Moreover, to avoid the massive actual storage of external keys for all ciphertexts, we leverage a pseudo-random function (PRF) $H(u_k||u_q||c||\gamma)$ built upon a cryptographic hash function as the external key generator so that TA no need to store any external key but only regenerate it when being legally queried.

Meanwhile, to address the decryption key exposure issue, we further append a randomization process for ciphertext and decryption key after each successful qualification check. A random factor r ensures that each (ct', k') temporary pair is independent. Thus, the exposed temporary decryption key is a one-off and cannot be used to decrypt other temporary ciphertexts.

Moreover, to achieve limited identity exposure resistance, users have two semi-functional identities: the KeyGen ID u_k and the CTQuery identity u_q. In this regard, unauthorized users can only decrypt the target ciphertext by acquiring a legitimate user's CTQuery identity and KeyGen identity. Revoked users can only decrypt the target ciphertext by acquiring the CTQuery identity of a legitimate user. In the event of original ciphertext exposure, a malicious user can only query from TA the original external key k for the target ciphertext by acquiring both identities of the data owner.

B. Detailed construction

- $\text{Setup}(\lambda, \mathcal{S})$: Let $(p, \mathbb{G}, \mathbb{G}_T, e)$ be a bilinear group. Let $N = |\mathcal{S}|$ be the size of the attribute universe. TA proceeds as follows:
  1) Choose random value $g, h_1, \ldots, h_N \in \mathbb{G}$, $a, \alpha \in \mathbb{Z}_p$, and then computes g^a, g^α.
  2) Select a hash function $H : \{0,1\}^* \to \mathbb{Z}_p$ and randomly select a binary string $\gamma \in \{0,1\}^*$.
  3) Set $\text{MSK} = (g^\alpha, H, \gamma)$ and publish $\text{param} = (p, e, \mathbb{G}, \mathbb{G}_T, g, g^a, e(g,g)^\alpha, h_1, \ldots, h_N)$.
- $\text{KeyGen}(u_k, \mathcal{S}(u), \text{MSK}, \text{param})$: Let u_k, $\mathcal{S}(u)$ be the private KeyGen id and the latest attribute set of user u. TA randomly picks $s_u \in \mathbb{Z}_p$, and computes

$$d_u = \left(d_{u_0} = g^\alpha \cdot g^{a \cdot s_u}, d'_{u_0} = g^{s_u}, \{d_{u_i} = h_i^{s_u}\}_{i \in \mathcal{S}(u)}\right).$$

- $\text{Encrypt}(\mathcal{M}, u_k, u_q, c, \beta, \text{MSK}, \text{param})$: Let β, $\mathcal{M}$, param be an access boolean formula, a message, and public parameters. DO proceeds as follows:
  1) Transform β to DNF as $\beta = (\beta_1 \vee \cdots \vee \beta_m)$, where β_i is a set of attributes, $i \in [1, m]$.
  2) Query the external key k from TA as follows:
     a) Send encryption discription item $\mathcal{I}_c = (u_k, u_q, c, \beta)$ to TA.
     b) TA verifies the validity of u_k and u_q. If u_k or u_q is invalid, then TA returns $\perp$; If u_k and u_q are valid, TA computes $k = H(u_k||u_q||c||\gamma)$, and returns k. Thereafter, TA stores $\mathcal{I}_c$ in the encryption description list.
  3) Randomly pick $s \in \mathbb{Z}_p$, then compute

$$C = \mathcal{M} \cdot e(g,g)^{\alpha \cdot s}, C_0 = g^{s \cdot k}.$$

  4) Assume that the size of β is l. Compare m with l.
     - If $m \le l$, compute

$$C_1 = \left(g^a \prod_{i \in \beta_1} h_i\right)^s, \ldots, C_m = \left(g^a \prod_{i \in \beta_m} h_i\right)^s.$$

     - If $m > l$, construct the LSSS matrix $(M, \rho) \in (\mathbb{Z}_p^{l \times n}, \mathcal{F}([1,l] \to [1,N]))$representing the original boolean formula β. Randomly select $\vec{v} = (s, y_2, \ldots, y_n) \in \mathbb{Z}_p^n$. For each row M_i, compute $\lambda_i = \vec{v} \cdot M_i,\ i \in [1, l]$. Then compute:

$$C_1 = g^{a \cdot \lambda_1} h_{\rho(1)}^{-s}, \ldots, C_l = g^{a \cdot \lambda_l} h_{\rho(l)}^{-s}.$$

  5) Output the ciphertext as $ct = (c, (M, \rho), C, C_0, \ldots, C_m)$ or $ct = (c, \beta, C, C_0, \ldots, C_l)$.
- $\text{CTQuery}(u_q, c, ct, \mathcal{S}(u), \mathcal{I}_c, \text{MSK})$: Let u_q be the CTQuery id of user u. To acquire the ciphertext, u proceeds as follows:
  1) Send $Q_{c,u_q} = (c, u_q)$ to CS.
  2) CS forwards Q_{c,u_q} to TA for qualification check.
  3) TA verifies whether u_q is valid. If not, TA returns $\perp$. Else, TA retrieves encryption description item $\mathcal{I}_c = (u_k^*, u_q^*, c, \beta)$ and the user attribute set $\mathcal{S}(u)$, then verifies whether $\mathbb{A}(c) \models \mathcal{S}(u)$.
     - If $\mathbb{A}(c) \models \mathcal{S}(u)$, TA randomly picks $r \in \mathbb{Z}_p$, and computes $k' = rk = rH(u_{k^*}||u_{q^*}||c||\gamma)$.Then, TA returns r to CS and k' to user u, respectively.
     - If $\mathbb{A}(c) \not\models \mathcal{S}(u)$, TA only returns $\perp$ to CS.
  4) If CS receives r from TA, it computes and returns $ct' = (c, (M, \rho), C, C_0^r, \ldots, C_l)$ or $ct' = (c, \beta, C, C_0^r, \ldots, C_m)$ to user u. Thereafter, CS immediately erases ct' and r. Otherwise, CS returns $\perp$ to user u.
- $\text{Decrypt}\,(ct', d_u, k', \text{param})$: Let (ct', k') be the temporary ciphertext and decryption key pair. Let m be the clauses' number in the access policy. User u compares the number of ciphertext elements $|ct'|$ with $m+1$.
  - If $|ct'| = m+1$, parse ct' as $(C_0', C_1\ , \ldots C_m)$. Find j such that $\beta_j \subset \mathcal{S}(u)$, compute

$$\begin{aligned}\mathcal{M} &= C / \frac{e\left(C_0'^{\frac{1}{k'}}, d_{u_0} \prod_{i \in \beta_j} d_{u_i}\right)}{e\left(d'_{u_0}, C_j\right)} \\ &= C / \frac{e\left(g^s, g^\alpha \left(g^a \prod_{i \in \beta_j} h_i\right)^{s_u}\right)}{e\left(g^{s_u}, \left(g^a \prod_{i \in \beta_j} h_i\right)^s\right)} \\ &= C / e(g,g)^{\alpha \cdot s}.\end{aligned}$$

  - If $|ct| \neq m+1$, parse ct as $(C_0', C_1, \ldots C_l)$. Define the set $I \subset \{1, 2, \ldots, l\}$ such that $I = \{i : \rho(i) \in \mathcal{S}(u)\}$. There exist coefficients $\{\omega_i \in \mathbb{Z}_p\}_{i \in I}$ such that $\sum_{i \in I} \omega_i M_i = (1, 0, \ldots, 0)$, and thus, $\sum_{i \in I} \omega_i \lambda_i = s$. Then compute:

$$\mathcal{M}=\frac{C}{e\left(\prod_{i\in I}C_i^{-\omega_i},d'_{u_0}\right)\cdot e\left(C_0'^{\frac{1}{k'}},d_{u_0}\prod_{i\in I}d_{u_{\rho(i)}}^{-\omega_i}\right)}.$$

C. Security proof

Theorem 1. Let hash function H be a random oracle. Let β^* and $\left(M'_{l'\times n'},\rho'\right)$ be the challenge access policy and the corresponding challenge LSSS matrix. Let $\beta^*=\beta_1^*\vee\cdots\vee\beta_m^*$ where $\beta_i^*,\ i\in[1,m]$ are disjoint sets. Let $\left(M^*_{l^*\times n^*},\rho^*\right)$ be the corresponding challenge LSSS matrix. If the decisional Modified-BDHE Assumption holds, then no PPT adversary can break the selective security of the proposed CP-ABE-SR scheme with a challenge access policy β^*, where $l',n',l^*,n^*\le q$.

Proof. Suppose $\mathcal{A}$ that can selectively break the security of the proposed scheme with non-negligible advantage $\mathrm{Adv}_{\mathcal{A}}^{\mathrm{CP-ABE-SR}}(\lambda)$, then $\mathcal{B}$ can solve the decisional Modified-BDHE problem. Let $(p,\mathbb{G},\mathbb{G}_T,e)$ be a bilinear pairing system, randomly choose $a,t,s,q\in\mathbb{Z}_p,\ g\in\mathbb{G}$. Given a decisional Modified-BDHE problem instance $\vec{Y}=\left(\begin{array}{c}p,e,\mathbb{G},\mathbb{G}_T,g,g^s,g^a,\ldots,g^{a^q},g^{a^{q+2}},\ldots,g^{a^{2q}},\\ g^{s(at+a)},g^{at},g^{a^2t},\ldots,g^{a^qt},g^{a^{q+2}t},\ldots,g^{a^{2q}t}\end{array}\right)$ and T, $\mathcal{B}$ is to ouput the guess of T.

Init. $\mathcal{A}$ sends a challenge access policy β^*, a challenge ciphertext identifier c^*, a challenge KeyGen identity u_k^*, and a challenge CTQuery identity u_q^* to $\mathcal{B}$.

Setup. $\mathcal{B}$ first transforms β^* to DNF as $\beta^*=\beta_1^*\vee\cdots\vee\beta_m^*$ where $\beta_i^*,i\in[1,m]$ are disjoint sets. $\mathcal{B}$ compares m and l'. There are two cases:

Case 1. $m>l'$. $\mathcal{B}$ constructs LSSS matrix $\left(M'_{l'\times n'},\rho'\right)$ from the original challenge access policy β^* where both $l',n'\le q$.

$\mathcal{B}$ proceeds as follows:

1) Randomly pick $\alpha'\in\mathbb{Z}_p$ and implicitly set $\alpha=\alpha'+a^{q+1}$.Compute $e(g,g)^{\alpha}=e\left(g^a,g^{a^q}\right)e(g,g)^{\alpha'}$.
2) Randomly choose $z_j\in\mathbb{Z}_p,\ \forall j\in[1,N]$. If $i\in[1,l']$ such that $j=\rho(i)$, then set $h_j=g^{z_j}\cdot g^{\sum_{i=1}^{n'}M'_{i,k}a^k}$; otherwise, set $h_j=g^{z_j}$.
3) Output the public parameter $\mathrm{param}=(p,e,\mathbb{G},\mathbb{G}_T,g,g^a,e(g,g)^{\alpha},h_1,\ldots,h_N)$.

Phase I. $\mathcal{B}$ initializes a revoked user list $\mathcal{R}$ to empty and answers queries.

$\mathcal{O}_{sk}(u_k,\mathcal{S}(u))$: On input the KeyGen identity u_k and the corresponding attribute set $\mathcal{S}(u)$, $\mathcal{B}$ first checks that:

- If $\mathcal{S}(u)\not\models\beta^*$, u_q is considered an unrevoked user. $\mathcal{B}$ proceeeds as follows:
  1) Construct a vector $\vec{x}=(x_1,\ldots,x_{n'})\in\mathbb{Z}_p^{n'}$ such that $x_1=-1$ and for all i where $\rho'(i)\in S$, the product $\vec{x}\cdot M_i'=0$. Based on proposition 1, the vector exists. Randomly choose $r\in\mathbb{Z}_p$ and implicitly set
  $$s_u=r+x_1a^q+x_2a^{q-1}+\cdots+x_{n'}a^{q-n'+1}.$$
  2) Compute
  $$\begin{aligned}d_{u_0}&=g^{\alpha}\cdot g^{a\cdot s_u}\\&=g^{\alpha'+a^{q+1}}\cdot g^{ar+\sum_{i=1}^{n'}x_i\cdot a^{q+2-i}}\\&=g^{\alpha'}g^{ar}\prod_{i=2,\ldots,n'}\left(g^{a^{q+1-i}}\right)^{x_i},\\d'_{u_0}&=g^{s_u}=g^r\prod_{i=1,\ldots,n'}\left(g^{a^{q+1-i}}\right)^{x_i}.\end{aligned}$$
  3) For $j\in S$ such that there is no $i\in[1,l']$ satisfying $\rho'(i)=j$, compute $h_j^{s_u}=\left(g^{s_u}\right)^{z_j}$; otherwise, compute
  $$h_j^{s_u}=g^{s_uz_j}\cdot g^{\left(r+x_1a^q+\cdots+x_{n'}a^{q-n'+1}\right)\sum_{k=1}^{n'}M'_{i,k}a^k}.$$
  Note that the product $\vec{x}\cdot M_i'=0$ thus $\mathcal{B}$ doesn't need to know the term $g^{a^{q+1}}$ to compute $h_j^{s_u}$.
  4) Finally, $\mathcal{B}$ sends $d_u=\left(d_{u_0},d'_{u_0},\{d_{u_i}\}_{i\in\mathcal{B}(u)}\right)$ to $\mathcal{A}$.
- If $\mathcal{S}(u)\models\beta^*$, u is considered a revoked user. $\mathcal{B}$ proceeeds as follows:
  1) Randomly choose $r\in\mathbb{Z}_p$ and implicitly set $s_u=r-a^q+a^{q-1}\cdot\frac{M'_{i,1}}{M'_{i,2}}$.
  2) Compute
  $$\begin{aligned}d_{u_0}&=g^{\alpha}\cdot g^{a\cdot s_u}=g^{\alpha'}\cdot g^{ar+a^q\cdot\frac{M'_{i,1}}{M'_{i,2}}},\\d'_{u_0}&=g^{s_u}=g^{r-a^q+a^{q-1}\cdot\frac{M'_{i,1}}{M'_{i,2}}}.\end{aligned}$$
  3) For $j\in S$ such that there is no $i\in[1,l']$ satisfying $\rho'(i)=j$, compute $h_j^{s_u}=\left(g^{s_u}\right)^{z_j}$; otherwise, compute
  $$h_j^{s_u}=g^{s_uz_j}\cdot g^{\left(r-a^q+a^{q-1}\cdot\frac{M'_{i,1}}{M'_{i,2}}\right)\sum_{k=1}^{n'}M'_{i,k}a^k}.$$
  Note that the terms $g^{a^{q+1}}$ generated by s_u cancel out each other thus $\mathcal{B}$ doesn't need to know the term $g^{a^{q+1}}$ to compute $h_j^{s_u}$.
  4) Finally, $\mathcal{B}$ sends $d_u=\left(d_{u_0},d'_{u_0},\{d_{u_i}\}_{i\in\mathcal{S}(u)}\right)$ to $\mathcal{A}$ and adds u to $\mathcal{R}$.

$\mathcal{O}_k(u_k,u_q,c)$: Before receiving a query on (u_k,u_q,c) from $\mathcal{A}$, prepare a empty hash list as follows.

- If $u_k=u_k^*$ and $u_q=u_q^*$ and $c=c^*$, $\mathcal{C}$ returns $\perp$.
- If (u_k,u_q,c) already exists, $\mathcal{B}$ responds according to the record.
- Otherwise, randomly select $k\in\mathbb{Z}_p$ and set $H(u_k||u_q||c||\gamma)=k$. Then, $\mathcal{B}$ responds with $H(u_k||u_q||c||\gamma)$ and adds $(u_k,u_q,c,H(u_k||u_q||c||\gamma))$ to the hash list.

Challenge. $\mathcal{A}$ gives two message $\mathcal{M}_0^*,\mathcal{M}_1^*$ with equal length to $\mathcal{B}$. $\mathcal{B}$ proceeds as follows:

1) Randomly pick $b\in\{0,1\}$, $k\in\mathbb{Z}_p$ and compute $C=\mathcal{M}_b^*\cdot T\cdot e\left(g^s,g^{\alpha'}\right),C_0=g^{s\cdot k}$.

2) Randomly choose $y'_2, \ldots, y'_{n'} \in \mathbb{Z}_p$ and set $\vec{v} = \left(s, sa + y'_2, sa^2 + y'_3, \ldots, sa^{n'-1} + y'_{n'}\right) \in \mathbb{Z}_p^{n'}$. For $i \in [1, n']$, compute

$$\begin{aligned} C_i =& g^{a\lambda_i} h_{\rho(i)}^{-s} \\ =& g^{a\left(sM'_{i,1} + \left(sa+y'_2\right)M'_{i,2} + \cdots + \left(sa^{n'-1}+y'_{n'}\right)M'_{i,n'}\right)} \\ & \cdot g^{-z_j} \cdot g^{-s\left(M'_{i,1}a + M'_{i,2}a^2 + \cdots + M'_{i,n'}a^{n'}\right)} \\ =& \left(\prod_{j=1,\cdots,n'} \left(g^a\right)^{M'_{i,j}y'_j}\right) \left(g^s\right)^{-z_j}. \end{aligned}$$

3) Finally, $\mathcal{B}$ sets $ct^* = (c, (M, \rho), C, C_0, \ldots, C_{l'})$.

Phase II. $\mathcal{B}$ answers the following queries.

$\mathcal{O}_{sk}(u_k, \mathcal{S}(u))$: Same as phase I.

$\mathcal{O}_k(uk, c)$: Same as phase I.

$\mathcal{O}_{k'}(u)$: On input any user 's public identity u, randomly select $t \in \mathbb{Z}_p$ and compute $k' = t \cdot r$. $\mathcal{B}$ sends k' to $\mathcal{A}$.

Guess. $\mathcal{A}$ outputs a guess b' of b.

- If $b = b'$, $\mathcal{B}$ outputs 0 to guess that $T = e(g, g)^{a^{q+1}s}$.
- If $b \neq b'$, $\mathcal{B}$ outputs 1 to guess that $T = R \in \mathbb{G}_T$.

Case 2. $m \leq l'$. $\mathcal{B}$ constructs LSSS matrix $\left(M^*_{l^* \times n^*}, \rho^*\right)$ from $\beta^* = \beta^*_1 \vee \cdots \vee \beta^*_m$ where both $l^*, n^* \leq q$.

$\mathcal{B}$ proceeds as follows:

1) Randomly pick $\alpha' \in \mathbb{Z}_p$ and implicitly set $\alpha = \alpha' + a^{q+1}$. Compute $e(g, g)^{\alpha} = e\left(g^a, g^{a^q}\right) e(g, g)^{\alpha'}$.
2) Define set of disjoint sets of rows of matrix M^* $\{I_j, j \in [1, m] : I_p \cap I_q = \varnothing, \forall p \neq q$ and $\{\rho(i), i \in I_j\} = \beta^*_j\}$ and describe β^* as: $(\wedge\rho(i))_{i\in I_1} \vee (\wedge\rho(i))_{i\in I_2} \vee \cdots \vee (\wedge\rho(i))_{i\in I_m}$.
3) Implicitly set $\vec{y} = \left(t, ta, ta^2, \ldots, ta^{n^*-1}\right)^{\perp} \in \mathbb{Z}_p^{n^*}$.Let $\vec{\lambda} = M^* \cdot \vec{y}$ be the vector shares, thus $\lambda_j = \sum_{i=1}^{n^*} M^*_{j,i} t a^{i-1}, j \in [1, l^*]$.Finds the set $\{\omega_i\}_{1\leq i\leq l^*}$ such that $\sum_{i\in I_j} \omega_i \cdot \lambda_i = t, j \in [1, m]$.
4) Randomly choose $z_j \in \mathbb{Z}_p$, $\forall j \in [1, N]$. If $i \in [1, l']$ such that $j = \rho(i)$, then set $h_j = g^{z_j} \cdot g^{a\omega_i\lambda_i} = g^{z_j} \cdot g^{\omega_i \sum_{k\in[n*]} M^*_{i,k} t a^k}$; otherwise, set $h_j = g^{z_j}$.
5) Publish the public parameter as $\text{param} = (p, e, \mathbb{G}, \mathbb{G}_T, g, g^a, e(g, g)^{\alpha}, h_1, \ldots, h_N)$.

Query phase 1. $\mathcal{B}$ initializes an revoked user list $\mathcal{R}$ to empty and answers queries.

$\mathcal{O}_{sk}(u_k, \mathcal{S}(u))$: On input the KeyGen identity u_k and the corresponding attribute set $\mathcal{S}(u)$, $\mathcal{B}$ first checks that:

- If $\mathcal{S}(u) \nvDash \beta^*$, u_q is considered an unrevoked user. $\mathcal{B}$ proceeds as follows:
  1) Find a vector $\vec{x} = (x_1, \ldots, x_{n^*}) \in \mathbb{Z}_p^{n*}$ such that $x_1 = -1$ and for all i where $\rho^*(i) \in S$ the product $\vec{x} \cdot M^*_i = 0$. Based on proposition 1, such vector $\vec{x}$ exists. $\mathcal{B}$ Randomly choose $r \in \mathbb{Z}_p$ and implicitly set

$$s_u = r + x_1 a^q + x_2 a^{q-1} + \cdots + x_{n^*} a^{q-n^*+1}.$$

  2) Compute

$$\begin{aligned} d_{u_0} &= g^{\alpha} \cdot g^{a \cdot s_u} \\ &= g^{\alpha' + a^{q+1}} \cdot g^{ar + \sum_{i=1}^{n^*} x_i \cdot a^{q+2-i}} \\ &= g^{\alpha'} g^{ar} \prod_{i=2,\ldots,n^*} \left(g^{a^{q+1-i}}\right)^{x_i}, \\ d'_{u_0} &= g^{s_u} = g^r \prod_{i=1,\ldots,n^*} \left(g^{a^{q+1-i}}\right)^{x_i}. \end{aligned}$$

  3) For $j \in S$ such that there is no $i \in [1, l^*]$ satisfying $\rho^*(i) = j$, compute $h_j^{s_u} = \left(g^{s_u}\right)^{z_j}$; otherwise, compute

$$h_j^{s_u} = g^{s_u z_j} g^{\left(r + x_1 a^q + \cdots + x_{n^*} a^{q-n^*+1}\right)\omega_i \sum_{k=1}^{n^*} M^*_{i,k} t a^k}.$$

  Note that the product $\vec{x} \cdot M^*_i = 0$ thus $\mathcal{B}$ doesn't need to know the term $g^{a^{q+1}t}$ to compute $h_j^{s_u}$.
  4) Finally, $\mathcal{B}$ sends $d_u = \left(d_{u_0}, d'_{u_0}, \{d_{u_i}\}_{i\in\mathcal{B}(u)}\right)$ to $\mathcal{A}$.
- If $\mathcal{S}(u) \vDash \beta^*$, u_q is considered a revoked user. $\mathcal{B}$ proceeeds as follows:
  1) Randomly choose $r \in \mathbb{Z}_p$ and implicitly set $s_u = r - a^q + a^{q-1} \cdot \frac{M^*_{i,1}}{M^*_{i,2}}$.
  2) Compute

$$\begin{aligned} d_{u_0} &= g^{\alpha} \cdot g^{a \cdot s_u} = g^{\alpha'} g^{ar + a^q \cdot \frac{M^*_{i,1}}{M^*_{i,2}}}, \\ d'_{u_0} &= g^{s_u} = g^{r - a^q + a^{q-1} \cdot \frac{M^*_{i,1}}{M^*_{i,2}}}. \end{aligned}$$

  3) For $j \in S$ such that there is no $i \in [1, l^*]$ satisfying $\rho^*(i) = j$, compute $h_j^{s_u} = \left(g^{s_u}\right)^{z_j}$; otherwise, compute

$$h_j^{s_u} = g^{s_u z_j} g^{\left(r - a^q + a^{q-1} \cdot \frac{M^*_{i,1}}{M^*_{i,2}}\right)\omega_i \sum_{k=1}^{n^*} M^*_{i,k} t a^k}.$$

  Note that the terms $g^{a^{q+1}}$ generated by s_u cancel out each other thus $\mathcal{B}$ doesn't need to know term $g^{a^{q+1}}t$ to compute $h_j^{s_u}$.
  4) Finally, $\mathcal{B}$ sends $d_u = \left(d_{u_0}, d'_{u_0}, \{d_{u_i}\}_{i\in\mathcal{S}(u)}\right)$ to $\mathcal{A}$ and adds u to $\mathcal{R}$.

$\mathcal{O}_k(u_k, u_q, c)$: Before receiving queries from $\mathcal{A}$, prepare a empty hash list as follows.

- If $u_k = u^*_k$ and $u_q = u^*_q$ and $c = c^*$, $\mathcal{C}$ returns $\perp$.
- If (u_k, u_q, c) already exists, $\mathcal{B}$ responds according to the record.
- Otherwise, randomly select $k \in \mathbb{Z}_p$ and set $H(u_k||u_q||c||\gamma) = k$. Then, $\mathcal{B}$ responds with $H(u_k||u_q||c||\gamma)$ and adds $(u_k, u_q, c, H(u_k||u_q||c||\gamma))$ to the hash list.

Challenge. $\mathcal{A}$ gives two message $\mathcal{M}^*_0, \mathcal{M}^*_1$ with equal length to $\mathcal{B}$. $\mathcal{B}$ proceeds as follows:

1) Randomly pick $b \in \{0, 1\}$, $k \in \mathbb{Z}_p$ and compute $C^* = \mathcal{M}^*_b \cdot T \cdot e\left(g^s, g^{\alpha'}\right), C^*_0 = g^{s \cdot k}$.

2) For $j \in [1, m]$, compute

$$C_j = \left(g^a \prod_{i \in \beta_j^*} h_i \right)^s = \left(g^a \prod_{i \in I_j} h_{\rho(i)} \right)^s$$
$$= \left(g^a \cdot \prod_{i \in I_j} g^{z_{\rho(i)}} \cdot g^{a\omega_i \lambda_i} \right)^s$$
$$= g^{s(a+at)} g^{\sum_{i \in I_j} s z_{\rho(i)}}.$$

Note that the disjoint set $\beta_j^* = \{\rho(i), i \in I_j\}$.

3) Finally, $\mathcal{B}$ sets $ct^* = (c, \beta^*, C, C_0, \ldots, C_{l^*})$.

Query phase 2.

$\mathcal{O}_{sk}(u_k, \mathcal{S}(u))$: Same as phase I.

$\mathcal{O}_k(uk, c)$: Same as phase I.

$\mathcal{O}_{k'}(u)$: On input any user 's public identity u, randomly select $t \in \mathbb{Z}_p$ and compute $k' = t \cdot r$. $\mathcal{B}$ sends k' to $\mathcal{A}$.

Guess. $\mathcal{A}$ outputs a guess b'.

- If $b = b'$, $\mathcal{B}$ guesses that $T = e(g, g)^{a^{q+1}s}$ with output 0.
- If $b \neq b'$, $\mathcal{B}$ guesses that $T = R \in \mathbb{G}_T$ with output 1.

Analisis. When $T = e(g, g)^{a^{q+1}s}$, $\mathcal{B}$ gives a perfect simulation in both cases, and thus we have

$$\Pr\left[\mathcal{B}\left(\vec{Y}, T = e(g, g)^{a^{q+1}s}\right) = 0\right] = \frac{1}{2} + \mathrm{Adv}_{\mathcal{A}}^{\mathrm{CP-ABE-SR}}(\lambda).$$

When $T = R \in \mathbb{G}_T$, the message $\mathcal{M}_b^*$ is completely randomized in both cases and thus we have

$$\Pr[\mathcal{B}(\vec{Y}, T = R) = 0] = \frac{1}{2}.$$

Therefore, the advantage of $\mathcal{B}$ solving the underlying problem is captured as

$$\mathrm{Adv}_{\mathcal{B}}^{\mathrm{Modified-BDHE}}(\lambda)$$
$$= \left| \Pr[\mathcal{B}(\vec{Y}, T = e(g, g)^{a^{q+1}s}) = 0] - \Pr[\mathcal{B}(\vec{Y}, T = R) = 0] \right|$$
$$= \mathrm{Adv}_{\mathcal{A}}^{\mathrm{CP-ABE-SR}}(\lambda),$$

which completes the proof.

V. Performance Evaluation

We compare the CP-ABE-SR scheme with recently proposed LSSS-based revocable ABE schemes [14], [16], [25]–[27] in functionality, computation overhead, storage, and transmission overhead. We proceed to implement the proposed scheme to demonstrate its efficiency and practicality.

A. Functionality

Table I compares the functionalities of the listed scheme. Notably, the ciphertext in [26] includes a fixed timestamp and can never be updated. As a result, only users that meet policy requirements within the specified time interval can decrypt it, rendering forward and backward secrecy and original ciphertext exposure attacks inapplicable. In [14], [25], [27] that use individual ciphertext update mechanisms, a vulnerability exists against original ciphertext exposure. In [25], the ciphertext update material is user-generated; in [14], the key and ciphertext update materials are publicly broadcast; in [27], the ciphertext update uses publicly available parameters. Therefore, a revoked user can update the original ciphertext to a desired version in these schemes. While in scheme [16] using overall ciphertext update mechanism, only the latest version ciphertext is stored, and the version can only be incrementally updated, hence OCE resistance. Nevertheless, the overall update mechanism is not suitable for large-scale file storage and frequent attribute update scenarios. Moreover, [16], [25], and [27] also support DKE resistance.

B. Computation

Table II compares the computational overheads of the listed schemes. For generality, we excluded the verification algorithm in [14], renamed the Revoke algorithm in [25] as DKeyGen, and partitioned the CTQuery algorithm in our scheme into CTUpdate and Decrypt.

In [16] and our scheme, the Setup algorithm can be achieved by randomly selecting elements rather than by exponentiation operations since the corresponding exponents are not required. In contrast, the exponentiation operation in [14] increases linearly with the number of attributes to support attribute-level revocation. Surprisingly, the KeyGen algorithm in [14] is unrelated to $|\mathcal{S}|$, because the attribute key is generated in the DKeyGen phase to enable attribute revocation. In terms of the UKeyGen algorithm, the scheme [14], [16], [26], [27] combine binary trees and KUNode algorithms to achieve user revocation, with complexity related to $\log(N/r_u)$, while in [25], the ciphertext update key (transformation key) is generated by users. In addition, [14] adds a complexity of $N - r_u$ to achieve user revocation, while [16] adds an ℓ factor to enable scalable key delegation. For the DKeyGen algorithm, [26] only involves multiplication operation, while other schemes is mainly linear with $|\mathcal{S}|$. The complexity of Encrypt algorithm is generally linear with l. For the CTUpdate algorithm, [25], [27], and our scheme require ciphertext updates for each request, while [14] and [16] adopt a periodic overall update mechanism. It is worth emphasizing that our scheme "mutes" the revocation process, with the cost of users needing to obtain a temporary decryption key each time, which is exposure-resistant. Overall, each algorithm in our proposed scheme demonstrates preferable computational efficiency.

C. Storage and transmission

Table III compares the storage and transmission overheads of the listed scheme, which exhibit similarity to computational overheads.

The public parameter size in [25], [26] is constant due to their unique revocation mechanisms. Ciphertext size generally increases linearly with the number of rows in the LSSS matrix or the timestamp length. For secret key size, [14] transfers the computation of private attribute keys to key updates achieving the minimal size. [16], [26], [27] use a user binary tree with the size linear to $|\mathcal{S}| \log N$, while [25] and ours are only linear with $|\mathcal{S}|$ .Furthermore, the update key size in our scheme remains constant, which maximally compensates for the frequent transmission overhead between the Trusted Authority and the Cloud Server caused by qualification checks.

TABLE I
FUNCTIONALITY COMPARISON

Schemes	User/Attribute revocation	Forward/Backward secrecy	Ciphertext update machenism	Silent revocation	DKE resistance	OCE resistance
[14]	U/A	F/B	Individual	✗	✗	✗
[16]	U	F/B	Overall	✗	✓	✓
[25]	U	F/B	Individual	✗	✓	✗
[26]	U	N/A	N/A	✗	✗	N/A
[27]	U	F/B	Individual	✗	✓	✗
Ours.	U/A	F/B	Individual	✓	✓	✓

TABLE II
COMPUTATION COMPARISON

Schemes	KeyGen	UKeyGen	DKeyGen	Encrypt	CTUpdate	Decrypt
[14]	$\mathcal{O}(\log N)\mathrm{E}$	$\mathcal{O}(\lvert\mathcal{U}\rvert \log(N/r_a) + N - r_u)\mathrm{E}$	$\mathcal{O}(\lvert\mathcal{S}\rvert)\mathrm{E}$	$\mathcal{O}(l+t)\mathrm{E}$	$\mathcal{O}(l)\mathrm{E}$	$\mathcal{O}(\lvert I\rvert)\mathrm{E} + \mathcal{O}(\lvert I\rvert)\mathrm{P}$
[16]	$\mathcal{O}(\ell\lvert\mathcal{S}\rvert \log N)\mathrm{E}$	$\mathcal{O}(r_u \log(N/r_u) + \ell\lvert\mathcal{S}\rvert)\mathrm{E}$	$\mathcal{O}(\ell\lvert\mathcal{S}\rvert)\mathrm{E}$	$\mathcal{O}(t(\ell l + t))\mathrm{E}$	$\mathcal{O}(t(\ell l + t))\mathrm{E}$	$\mathcal{O}(\lvert I\rvert)\mathrm{E} + \mathcal{O}(\lvert I\rvert)\mathrm{P}$
[25]	$\mathcal{O}(\lvert\mathcal{S}\rvert)\mathrm{E} + \mathcal{O}(1)P$	$\mathcal{O}(\lvert\mathcal{S}\rvert)\mathrm{E}$	$\mathcal{O}(\lvert\mathcal{S}\rvert)\mathrm{E} + \mathcal{O}(1)P$	$\mathcal{O}(l)\mathrm{E}$	$\mathcal{O}(\lvert I\rvert)\mathrm{E} + \mathcal{O}(\lvert I\rvert)\mathrm{P}$	$\mathcal{O}(1)\mathrm{E}$
[26]	$\mathcal{O}(\lvert\mathcal{S}\rvert \log N)\mathrm{E}$	$\mathcal{O}(\lvert\mathcal{S}\rvert \log(N/r_u))\mathrm{E}$	$\mathcal{O}(\lvert\mathcal{S}\rvert)\mathrm{M}$	$\mathcal{O}(nl)\mathrm{E}$	N/A	$\mathcal{O}(\lvert I\rvert)\mathrm{E} + \mathcal{O}(1)\mathrm{P}$
[27]	$\mathcal{O}(\lvert\mathcal{S}\rvert \log N)\mathrm{E}$	$\mathcal{O}(\log(N/r_u))\mathrm{E}$	$\mathcal{O}(\lvert\mathcal{S}\rvert)\mathrm{E}$	$\mathcal{O}(l+t)\mathrm{E}$	$\mathcal{O}(l)\mathrm{E}$	$\mathcal{O}(\lvert I\rvert)\mathrm{E} + \mathcal{O}(\lvert I\rvert)\mathrm{P}$
Ours.	$\mathcal{O}(\lvert\mathcal{S}\rvert)\mathrm{E}$	N/A	N/A	$\mathcal{O}(m')\mathrm{E}$	$\mathcal{O}(1)\mathrm{E}$	$\mathcal{O}(\{1, \lvert I\rvert\})\mathrm{E} + \mathcal{O}(1)\mathrm{P}$

** l = the number of rows in the LSSS matrix. ℓ = the maximum length of attribute vectors. $m' = \min\{l, m\}$, where m = the number of clause in a DNF. r_a = the number of the revoked users of attributes to be revoked. r_u = the number of users to be revoked. t = the length of the binary representation of the maximum version number. N = the number of users. $|I|$ = the number of attributes involved in the decryption algorithm. $|\mathcal{U}|$ = the attribute universe size. $|\mathcal{S}|$ = the size of a user's attribute set. E = exponentiation. M = multiplication. P = pairing.*

TABLE III
TRANSMISSION AND STORAGE COMPARISON

Schemes	Public Parameter Size	Ciphertext Size	Secret Key Size	Update Key Size
[14]	$\mathcal{O}(\lvert\mathcal{U}\rvert + t)\lvert\mathbb{G}\rvert + \mathcal{O}(1)\lvert\mathbb{G}_T\rvert$	$\mathcal{O}\left(t+l\right)\lvert\mathbb{G}\rvert + \mathcal{O}(1)\lvert\mathbb{G}_T\rvert$	$\mathcal{O}(\log N)\lvert\mathbb{G}\rvert$	$\mathcal{O}\left(\lvert\mathcal{U}\rvert \log\left(N/r_u\right) + N - r_u\right)\lvert\mathbb{G}\rvert$
[16]	$\mathcal{O}(\lvert\mathcal{U}\rvert + t + \ell)\lvert\mathbb{G}\rvert + \mathcal{O}(1)\lvert\mathbb{G}_T\rvert$	$\mathcal{O}(t(t+l))\lvert\mathbb{G}\rvert + \mathcal{O}(t)\lvert\mathbb{G}_T\rvert$	$\mathcal{O}(\ell\lvert\mathcal{S}\rvert \log N)\lvert\mathbb{G}\rvert$	$\mathcal{O}\left(r_u \log\left(N/r_u\right) + \ell\lvert\mathcal{S}\rvert\right)\lvert\mathbb{G}\rvert$
[25]	$\mathcal{O}(1)\lvert\mathbb{G}\rvert + \mathcal{O}(1)\lvert\mathbb{G}_T\rvert$	$\mathcal{O}\left(l\right)\lvert\mathbb{G}\rvert + \mathcal{O}(1)\lvert\mathbb{G}_T\rvert$	$\mathcal{O}(\lvert\mathcal{S}\rvert)\lvert\mathbb{G}\rvert + \mathcal{O}(1)\lvert\mathbb{G}_T\rvert$	$\mathcal{O}(\lvert\mathcal{S}\rvert)\lvert\mathbb{G}\rvert$
[26] [3]	$\mathcal{O}(1)\lvert\mathbb{G}\rvert + \mathcal{O}(1)\lvert\mathbb{G}_T\rvert$	$\mathcal{O}(l)\lvert\mathbb{G}\rvert + \mathcal{O}(1)\lvert\mathbb{G}_T\rvert$	$\mathcal{O}(\lvert\mathcal{S}\rvert \log N)\lvert\mathbb{G}\rvert$	$\mathcal{O}(\lvert\mathcal{S}\rvert \log(N/r_u))\lvert\mathbb{G}\rvert$
[27]	$\mathcal{O}(t)\lvert\mathbb{G}\rvert + \mathcal{O}(1)\lvert\mathbb{G}_T\rvert$	$\mathcal{O}\left(l+t\right)\lvert\mathbb{G}\rvert + \mathcal{O}(1)\lvert\mathbb{G}_T\rvert$	$\mathcal{O}(\lvert\mathcal{S}\rvert \cdot \log N)\lvert\mathbb{G}\rvert$	$\mathcal{O}\left(\log\left(N/r_u\right)\right)\lvert\mathbb{G}\rvert$
Ours.	$\mathcal{O}(\lvert\mathcal{U}\rvert)\lvert\mathbb{G}\rvert + \mathcal{O}(1)\lvert\mathbb{G}_T\rvert$	$\mathcal{O}(m')\lvert\mathbb{G}\rvert + \mathcal{O}(1)\lvert\mathbb{G}_T\rvert$	$\mathcal{O}(\lvert\mathcal{S}\rvert)\lvert\mathbb{G}\rvert$	$\mathcal{O}(1)\lvert\mathbb{Z}_p\rvert$

** We denote a symmetric-pairing group by $(\mathbb{G}, \mathbb{G}_T)$.*

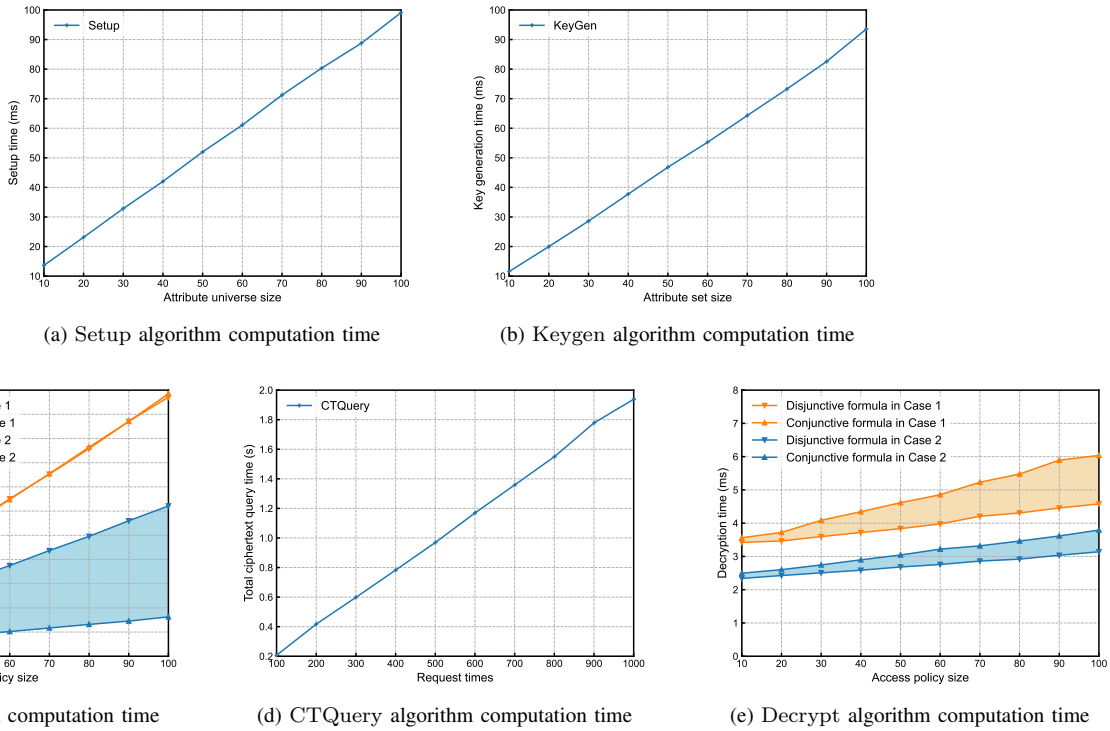
Overall, the proposed schemes show favorable performance on the storage and transmission overheads.

D. Implementation

We implemented the proposed scheme using the Charm [32] 0.50 package built upon PBC [33] and GMP [34] libraries in Python 3.7. We selected SS512, the 512-bit base field super-singular symmetric elliptic curve, and SHA-512 to instantiate the proposed scheme. All benchmarks were conducted on a computer with Intel Core i5-10200H CPU @ 2.40GHz CPU and 16.0 GB memory running on Ubuntu 20.04.

The user attribute set and the attribute universe are identical, with a size of 100. We used the conjunction and disjunction formulas in the Encrypt and Decrypt algorithm benchmark to fully demonstrate the efficiency gap between the Case 1 and Case 2 algorithms. Specifically, the conjunction formula ranges from $(x_1 \wedge x_2 \wedge \ldots \wedge x_{10})$ to $(x_1 \wedge x_2 \wedge \ldots \wedge x_{100})$ with a step of 10, while the disjunction formula simultaneously ranges from $(x_1 \vee x_2 \vee \ldots \vee x_{10})$ to $(x_1 \vee x_2 \vee \ldots \vee x_{100})$with the same step. This setting has many advantages. Firstly, both conjunction and disjunction formulas are in the simplest form and can simultaneously satisfy the access policy requirements of Case 1 and Case 2. Therefore, we can exclude the unstable and subtle calculations caused by the simplification and transformation of boolean formulas. Furthermore, for an access policy of a given size, the logical connectives vary, making it difficult to accurately predict the computational cost of the algorithm for each boolean formula. However, we can take the computational cost of the conjunction and disjunction formulas as the upper or lower bound of the algorithm's performance. Using the conjunction and disjunction formulas, we can visually reflect the overhead boundaries of Case 1 and Case 2 algorithms and the efficiency gap between the two cases. In addition, we take the average of 100 executions for all benchmark results.

[3]The scheme involves an asymmetric-pairing group, but we still use the symmetric-pairing group notation for brevity.



(a) Setup algorithm computation time

(b) Keygen algorithm computation time

(c) Encrypt algorithm computation time

(d) CTQuery algorithm computation time

(e) Decrypt algorithm computation time

Fig. 2. Computation time of our proposed CP-ABE-SR scheme.

In Fig. 2(a), the setup time increases linearly with the attribute universe size, which is reasonable because the significant overhead of the setup algorithm lies in the random element selection for each attribute. Fig. 2(b) shows that the key generation algorithm of the proposed scheme increases linearly with the user attribute set size, and KeyGen time for an attribute set of 100 only takes about 91 ms. Overall, the Setup and KeyGen algorithm performs better than the listed revocable ABE schemes because, basically, we do not modify the two algorithms of the underlying efficient ABE scheme.

Fig 2(c) illustrates the relation between access policies and the Encrypt algorithm performance. As mentioned earlier, given the access policy size, the performance of Case 1 and Case 2 algorithms for the conjunction and disjunction formulas can be regarded as the upper or lower bound. Firstly, it is evident that regardless of the conjunction or disjunction formula, the Case 2 algorithm overhead is more efficient than Case 1. Secondly, it can be noted that for the Case 1 algorithm, the cost of the conjunction and disjunction formulas with the same size is almost the same because the column number of the LSSS matrix is not the decisive factor for the overhead of the Case 1 algorithm. On the contrary, the Case 2 algorithm shows outstanding advantages in the computation of the conjunction formula because the pairing operation reduces to only one for the disjunction formula.

In Fig. 2(d), we illustrate how the request time impacts the total running time of the CTQuery algorithm under serial execution mode. Based on the benchmark results, we can determine that the average response time of the CTQuery algorithm is 2 ms. The result implies that a transition from active re-encryption revocation mechanisms centered around the invalidation of revoked users' keys to passive re-encryption revocation mechanisms centered around the invalidation of exposed ciphertexts will lead to a significant reduction of computational overhead from periodic revocation-related variables to minor request-related constants for both TA and CS.

Fig. 2(e) shows the effect of the access policy size on decryption time. Like Figure 4, the Case 2 algorithm consistently outperforms the Case 1 algorithm. The most surprising thing is that the decryption operation under an access policy with a size of 100 takes at most 6 ms for Case 1 and less than 4 ms for Case 2. In fact, for the Case 1 algorithm, if the access policy is a bool formula, the constant $\omega_i \in \{0, 1\}$, thus we can eliminate the exponentiation operation related to ω_i. Finally, both cases in the Decrypt algorithm only require one exponentiation operation and two pairing operations.

VI. Conclusion

As current revocable ABE schemes are not suitable for large-scale cloud storage and dynamic attribute updating scenarios, this paper introduced CP-ABE with silent revocation (CP-ABE-SR) which only requires the trusted authority to execute user/attribute revocation silently in a revocation event by leveraging a user qualification check protocol in the ciphertext query phase, resulting in a significant reduction in computational cost from periodic revocation-related variables to minor request-related constants for both the cloud server and the trusted authority. Furthermore, the scheme resists potential exposure attacks: decryption key exposure, original ciphertext exposure, and user identity exposure. We gave the concrete construction of the CP-ABE-SR scheme based on an efficient CP-ABE scheme and proved it IND-CPA secure in

the ROM. Theoretical analysis has shown that our proposed scheme exhibits significant functionality, computation, and storage advantages. We also implemented proposed CP-ABE-SR scheme and demonstrated its practicality and efficiency.

While our proposed solution addresses many of the challenges in the current system, we have identified an interesting open problem regarding the collusion resistance between revoked users and CS, which deserves further investigation for a higher security level.

ACKNOWLEDGMENTS

This work was partly supported by the National Natural Science Foundation of China (No. 62102311) and partly by the Natural Science Basic Research Program of Shaanxi (Program No. 2022JQ-600).

REFERENCES

[1] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Adv. Cryptology–EUROCRYPT 2005 24th Annu. Int. Conf. Theory Appl. Cryptogr. Tech. Aarhus Den. May 22-26 2005 Proc. 24*. Springer, 2005, pp. 457–473.

[2] R. R. Al-Dahhan, Q. Shi, G. M. Lee, and K. Kifayat, "Survey on revocation in ciphertext-policy attribute-based encryption," *Sensors*, vol. 19, no. 7, p. 1695, 2019.

[3] J. H. Seo and K. Emura, "Revocable identity-based encryption revisited: Security model and construction," in *Public-Key Cryptogr. 2013 16th Int. Conf. Pract. Theory Public-Key Cryptogr. Nara Jpn. Febr. 26–March 1 2013 Proc. 16*. Springer, 2013, pp. 216–234.

[4] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *2007 IEEE Symp. Secur. Priv. SP07*. IEEE, 2007, pp. 321–334.

[5] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. 13th ACM Conf. Comput. Commun. Secur.*, 2006, pp. 89–98.

[6] Y. Zhang, R. H. Deng, S. Xu, J. Sun, Q. Li, and D. Zheng, "Attribute-based encryption for cloud computing access control: A survey," *ACM Comput. Surv. CSUR*, vol. 53, no. 4, pp. 1–41, 2020.

[7] R. Imam, K. Kumar, S. M. Raza, R. Sadaf, F. Anwer, N. Fatima, M. Nadeem, M. Abbas, and O. Rahman, "A systematic literature review of attribute based encryption in health services," *J. King Saud Univ.-Comput. Inf. Sci.*, 2022.

[8] M. Rasori, M. La Manna, P. Perazzo, and G. Dini, "A survey on attribute-based encryption schemes suitable for the internet of things," *IEEE Internet Things J.*, 2022.

[9] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Adv. Cryptology—CRYPTO 2001 21st Annu. Int. Cryptol. Conf. St. Barbara Calif. USA August 19–23 2001 Proc.* Springer, 2001, pp. 213–229.

[10] M. Pirretti, P. Traynor, P. McDaniel, and B. Waters, "Secure attribute-based systems," in *Proc. 13th ACM Conf. Comput. Commun. Secur.*, 2006, pp. 99–112.

[11] L. Ibraimi, M. Petkovic, S. Nikova, P. Hartel, and W. Jonker, "Mediated ciphertext-policy attribute-based encryption and its application," in *Inf. Secur. Appl. 10th Int. Workshop WISA 2009 Busan Korea August 25-27 2009 Revis. Sel. Pap. 10*. Springer, 2009, pp. 309–323.

[12] J. Hur and D. K. Noh, "Attribute-based access control with efficient revocation in data outsourcing systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 7, pp. 1214–1221, 2010.

[13] C. Ge, W. Susilo, J. Baek, Z. Liu, J. Xia, and L. Fang, "Revocable attribute-based encryption with data integrity in clouds," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 5, pp. 2864–2872, 2021.

[14] S. Deng, G. Yang, W. Dong, and M. Xia, "Flexible revocation in ciphertext-policy attribute-based encryption with verifiable ciphertext delegation," *Multimed. Tools Appl.*, pp. 1–24, 2022.

[15] L. Zhang, J. Su, and Y. Mu, "Outsourcing attributed-based ranked searchable encryption with revocation for cloud storage," *IEEE Access*, vol. 8, pp. 104 344–104 356, 2020.

[16] J. Wei, X. Chen, X. Huang, X. Hu, and W. Susilo, "RS-HABE: Revocable-storage and hierarchical attribute-based access scheme for secure sharing of e-health records in public cloud," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 5, pp. 2301–2315, 2019.

[17] M. Bouchaala, C. Ghazel, and L. A. Saidane, "Trak-cpabe: A novel traceable, revocable and accountable ciphertext-policy attribute-based encryption scheme in cloud computing," *J. Inf. Secur. Appl.*, vol. 61, p. 102914, 2021.

[18] D. Han, N. Pan, and K.-C. Li, "A traceable and revocable ciphertext-policy attribute-based encryption scheme based on privacy protection," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 1, pp. 316–327, 2020.

[19] Z. Liu, S. Duan, P. Zhou, and B. Wang, "Traceable-then-revocable ciphertext-policy attribute-based encryption scheme," *Future Gener. Comput. Syst.*, vol. 93, pp. 903–913, 2019.

[20] J. Yu, R. Hao, H. Zhao, M. Shu, and J. Fan, "IRIBE: Intrusion-resilient identity-based encryption," *Inf. Sci.*, vol. 329, pp. 90–104, 2016.

[21] Q. Xing, B. Wang, X. Wang, and J. Tao, "Unbounded and revocable hierarchical identity-based encryption with adaptive security, decryption key exposure resistant, and short public parameters," *PloS one*, vol. 13, no. 4, p. e0195204, 2018.

[22] A. Takayasu and Y. Watanabe, "Revocable identity-based encryption with bounded decryption key exposure resistance: Lattice-based construction and more," *Theor. Comput. Sci.*, vol. 849, pp. 64–98, 2021.

[23] H. Cui, R. H. Deng, Y. Li, and B. Qin, "Server-aided revocable attribute-based encryption," in *Comput. Secur. 2016 21st Eur. Symp. Res. Comput. Secur. Heraklion Greece Sept. 26-30 2016 Proc. Part II 21*. Springer, 2016, pp. 570–587.

[24] B. Qin, Q. Zhao, D. Zheng, and H. Cui, "(Dual) server-aided revocable attribute-based encryption with decryption key exposure resistance," *Inf. Sci.*, vol. 490, pp. 74–92, 2019.

[25] R. Guo, G. Yang, H. Shi, Y. Zhang, and D. Zheng, "O3-R-CP-ABE: An efficient and revocable attribute-based encryption scheme in the cloud-assisted IoMT system," *IEEE Internet Things J.*, vol. 8, no. 11, pp. 8949–8963, 2021.

[26] H. Xiong, X. Huang, M. Yang, L. Wang, and S. Yu, "Unbounded and efficient revocable attribute-based encryption with adaptive security for cloud-assisted internet of things," *IEEE Internet Things J.*, vol. 9, no. 4, pp. 3097–3111, 2021.

[27] S. Xu, G. Yang, Y. Mu, and X. Liu, "A secure IoT cloud storage system with fine-grained access control and decryption key exposure resistance," *Future Gener. Comput. Syst.*, vol. 97, pp. 284–294, 2019.

[28] A. Beimel, "Secure schemes for secret sharing and key distribution," 1996.

[29] Q. M. Malluhi, A. Shikfa, and V. C. Trinh, "A ciphertext-policy attribute-based encryption scheme with optimized ciphertext size and fast decryption," in *Proc. 2017 ACM Asia Conf. Comput. Commun. Secur.*, 2017, pp. 230–240.

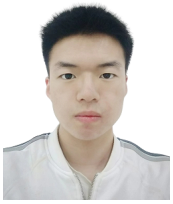
[30] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *Public Key Cryptogr. 2011 14th Int. Conf. Pract. Theory Public Key Cryptogr. Taormina Italy March 6-9 2011 Proc. 14*. Springer, 2011, pp. 53–70.

[31] S. Hohenberger and B. Waters, "Attribute-based encryption with fast decryption," in *Public-Key Cryptogr. 2013 16th Int. Conf. Pract. Theory Public-Key Cryptogr. Nara Jpn. Febr. 26–March 1 2013 Proc. 16*. Springer, 2013, pp. 162–179.

[32] J. A. Akinyele, C. Garman, I. Miers, M. W. Pagano, M. Rushanan, M. Green, and A. D. Rubin, "Charm: A framework for rapidly prototyping cryptosystems," *J. Cryptogr. Eng.*, vol. 3, pp. 111–128, 2013.

[33] B. Lynn. (2006) PBC library. [Online]. Available: http://crypto.stanford.edu/pbc

[34] T. Granlund. (1996) The GNU multiple precision arithmetic library. [Online]. Available: https://gmplib.org/



Lukai Qin is pursuing a BS Degree in Cyberspace Security from Xi'an University of Posts and Telecommunications. He has developed several cryptographic systems and cross-chain systems. His recent research interests include cryptography and blockchain.

Shuanggen Liu received a Ph.D. degree in cryptography from Xidian University in 2008. He is currently a Professor at the School of Cyber Security, Xi'an University of Posts and Telecommunications, Xi'an, China. His recent research interests include cryptography and information security. He is a member of the China Computer Federation and the Chinese Association for Cryptologic Research.

Xiaoqing Liu is pursuing a BS Degree in Cyberspace Security from Xi'an University of Posts and Telecommunications. She has experience in cryptographic system design and development. Her recent research interests include cryptography, and blockchain.

Jindong Yu is pursuing a BS Degree in Cyberspace Security from Xi'an University of Posts and Telecommunications. He has experience in cryptographic system design and development. His recent research interests include blockchain and network attack and defense technology.

Xu An Wang is now a professor in Engineering University of People's Armed Police. His main research interests include cloud computing, cryptography, security and privacy for IoT, and information security. He has published about 100 papers in cloud computing, cryptography, information security, and computer science.

Teng Wang received a Ph.D. degree in the School of Computer Science and Technology from Xi'an Jiaotong University, China, in 2020 and a BS degree in the School of Software from XiDian University, China, in 2015. She is currently a lecturer in the School of Cyberspace Security at Xi'an University of Posts and Telecommunications. She has been a visiting Ph.D. student at the School of Computer Science and Engineering at Nanyang Technological University in Singapore from 2018 to 2019. Her research interests include mobile crowdsourcing systems (MCS), privacy-preserving data collection and analysis, privacy-preserving machine learning, and privacy computing. She won the support of Young Outstanding Talents of Shaanxi Province in 2022.