An Efficient Revocable Attribute-based Encryption Scheme Supporting Data Integrity*

Yue Zhang[1], Meijuan Huang[1], Bo Yang[2], and Haijun Liu[3†]

[1] Baoji University of Arts and Sciences, Baoji, Shaanxi, China.
huangmeijuan@bjwlxy.edu.cn, zhangzhang992024@163.com
[2] Shaanxi Normal University, Xi'an, Shaanxi, China
byang@snnu.edu.cn
[3] Shaanxi Aerospace Time Navigation Equipment Co.,Ltd, Baoji, Shaanxi, China.
875153075@qq.com

Abstract

Currently, cloud storage services offer great convenience for personal and business data. Especially in the mobile Internet environment, the demand for users to access cloud data anytime and anywhere through mobile phones, tablets and other terminals is increasing day by day. However, the openness of the mobile Internet and the portability of terminal devices have also intensified the security risks of sensitive data leakage and illegal access. To achieve the security protection and flexible sharing of sensitive data by cloud storage users in the mobile Internet scenario, dynamic access control has become the core demand, and Revocable Attribute-based Encryption (RABE) has become an effective solution. To reduce the computational load of attribute revocation, most RABE schemes have the revocation operation performed by cloud servers. Such an operation brings convenience to users and also brings data integrity issues -how to ensure the plaintext for the revoked ciphertext remains identical to the original plaintext. The existing RABE schemes that support data integrity are almost all based on the dual-attribute encryption form. The computational overhead of encryption and decryption in this encryption method is very high, and it is not practical in actual scenarios. To this end, we propose an efficient RABE scheme supporting data integrity (RABE-DI), and under the decision linear assumption and the discrete logarithm assumption, prove that our scheme achieves semantic security and data integrity. Performance evaluation shows that our scheme outperforms existing ones in efficiency, particularly during the decryption phase.

Keywords: Cloud Storage, Attribute-based Encryption, Attribute Revocation, Data Integrity.

1 Introduction

Nowadays,cloud computing, enabling individuals and organizations to outsource [1] and share data [2], has become increasingly prevalent. But there is often some sensitive information contained in the data. To maintain data confidentiality, data must be encrypted before being uploaded to the cloud. Since traditional public key encryption only enables one-to-one sharing, it is difficult to realize the share of data among multiple users. The attribute-based encryption (ABE) [3] technique effectively solves this problem by realizing one-to-many sharing mechanism, so this becomes a promising solution for access control of encrypted data in cloud storage. In ciphertext policy attribute-based encryption (CP-ABE) [4], the user's attribute set is stored in

*Proceedings of the 9th International Conference on Mobile Internet Security (MobiSec'25), Article No. 20, December 16-18, 2025, Sapporo, Japan.

†Corresponding Author

the key and the access structure is attached to the ciphertext. Access to data is permitted when the user's attribute set satisfies the predefined access structure. Waters et al. [5] constructed a practical CP-ABE scheme based on a linear secret sharing scheme. Subsequently, based on the literature [5], many scholars have researched CP-ABE in terms of functionality [6, 7], efficiencies [8, 9]. However, all the above-mentioned CP-ABE schemes are implemented on small attribute domains and are not practical in actual scenarios. Therefore, Agrawal et al. [10] constructed a fast attribute message encryption scheme in unbounded attribute space, which is more practical and has relatively constant decryption time.

In CP-ABE, the access structure is fixed during the encryption process, posing a key challenge: how to enable user access revocation through dynamic adjustment of the access structure. RABE [11] provides a flexible and fine-grained data access control solution by controlling user attributes. Revocation can be categorized into two forms: direct revocation and indirect revocation. In literature [12], a direct revocation RABE scheme was proposed. In their system, fine-grained access is controlled by embedding a user revocation list in the revocation ciphertext. In literature [13], direct revocation of a user is accomplished by binding a timestamp to the set of attributes, while a private key is periodically generated by KGC for the unrevoked user. However, this leads to a very inefficient scheme, as key updates are required at each new time period. To reduce the amount of computation, most RABE programs need to introduce a third party to perform outsourcing operations. However, after the third party performs the revocable, the integrity of the data cannot be verified by the user. So literature [14] and [15] proposed the RABE scheme that can verify the data integrity. However, their RABE scheme uses double attribute-based encryption to ensure data integrity, and this method has high computational cost. Therefore, how to improve the computational efficiency of the RABE scheme with integrity remains a problem that needs to be improved.

1.1 Related works

At present, a large amount of research has been conducted on revocable attribute-based encryption [16–20]. Among them, Wang et al. [18] achieved attribute-level user revocation by updating the corresponding ciphertext when the attribute was revoked. However, the update process will bring certain computing and communication costs. Hur et al. [19] proposed a RABE scheme using key cryptographic key trees, which achieves fine-grained attribute revocation by introducing attribute groups, but it cannot resist collusion attacks by revoked and unrevoked users. Li et al. [20] introduced a new RABE scheme, in which the data user must have both the private key and the attribute group key to successfully decrypt the ciphertext. In the event of revocation of an attribute, the system needs to update the attribute group key and the ciphertext, so the revoked data user will lose the decryption privilege. Tu et al. [21] proposed an ABE scheme that combines outsourcing techniques with multiple authorization authorities, which enables attribute revocation by introducing attribute group keys. These schemes need to update the user's private key when performing the revocation operation, which leads to a large amount of computation and low efficiency for the schemes.

To this end, literature [22] proposes a RABE scheme in which a cloud server performs the revocable operation. But with the introduction of third-party service providers, this can lead to potential risks to data security and difficulties in ensuring data integrity. Ge et al. [14] were the first to introduce the concept of data integrity, constructing a novel RABE scheme that supports data integrity through a user-verifiable approach. Huang et al. [23] improves the work in literature [14] to enhance efficiency while solving the key escrow problem. However, their scheme can only be applied in scenarios with small attribute domains, so the practicality of the

scheme is not high. Based on the literature [10], Chen et al. [15] proposed a RABE scheme with data integrity but in order to ensure data integrity, the scheme performs double attribute-based encryption, which makes the whole scheme less efficient.

Attribute-Based Proxy Re-Encryption (AB-PRE) [24] technology can provide efficient support for access permission revocation. In AB-PRE, the authorized agent can convert the ciphertext of the data owner into a form that can be decrypted by the authorized user without obtaining the plaintext and key. Yu et al. [25] proposed a RABE scheme that utilizes AB-PRE to perform attribute revocation ciphertext, the proxy converts the ciphertext under a specific access policy. Obviously, there are data integrity issues in AB-PRE as well.

1.2 Contributions

The specific contributions are as follows:

1)An efficient RABE scheme supporting data integrity (RABE- DI) is constructed based on the literature [10]. We introduce symmetric encryption so that only a fast attribute-based encryption is used in the encryption phase and only a constant number of logarithmic operations are required in the decryption operation. This improves the computational efficiency of encryption and decryption. In addition, it can also ensure that when the cloud incorrectly performs a undo, it can be detected by users.

2)Our proposed RABE-DI scheme is proven to be adaptive chosen plaintext secure based on the decision-based linear assumption. At the same time, it is proved that the scheme has data integrity based on the discrete logarithmic assumption.

3)We evaluate and compare the performance of the proposed scheme with the existing schemes in four aspects: key generation, encryption, decryption and revocation. The experimental results show that, the results show that our scheme is efficient and safe.

2 Preliminaries

Let G be a group generator, it inputs 1^{λ}, outputs three cyclic groups $\mathbb{G}$, $\mathbb{H}$, $\mathbb{G}_T$ with prime order p, bilinear map $e : \mathbb{G} \times \mathbb{H} \to \mathbb{G}_T$, g, h, where g and h are generators of $\mathbb{G}$ and $\mathbb{H}$ respectively.

2.1 Bilinear mapping

For $\forall a \in \mathbb{G}$, $b \in \mathbb{H}$, the bilinear mapping $e : \mathbb{G} \times \mathbb{H} \to \mathbb{G}_T$ meets the following properties:

1) Non-degeneracy: $e(a, b) \neq 1_{\mathbb{G}_T}$ (where, $1_{\mathbb{G}_T}$ is the identity element of $\mathbb{G}_T$);

2)Bilinear: $e(a^u, b^v) = e(a, b)^{uv}$, $\forall u, v \in \mathbb{Z}_p^*$;

3)Computability: $e(a, b)$can be efficiently calculated.

2.2 Complexity Assumption

Discrete Logarithm Assumption (DL) Given $(\mathbb{G}, g, p, \mu, g^{\mu})$, where $\mu \in \mathbb{Z}_p^*$, $G\left(1^{\lambda}\right) \to (\mathbb{G}, g, p)$.The DL assumption says, that for a probabilistic polynomial time (PPT) adversary $\mathcal{A}$ find the integer μ, the advantage of $\mathcal{A}$'s
$\Pr\left[\mathcal{A}\left(e, \mathbb{G}, g, p, g^{\mu}\right) = \mu\right]$ negligible in λ.

Decision Linear Assumption (DLIN) Given a tuple
$D = (\mathbb{H}, \mathbb{G}, h^{s_1 a_1}, h^{a_1}, h, h^{s_2 a_2}, h^{a_2}, g^{s_1 a_1}, g^{a_1}, g, g^{s_2 a_2}, g^{a_2})$, where $\mu \in \mathbb{Z}_p^*$, $G\left(1^{\lambda}\right) \to (\mathbb{H}, \mathbb{G}, h, g, p)$,

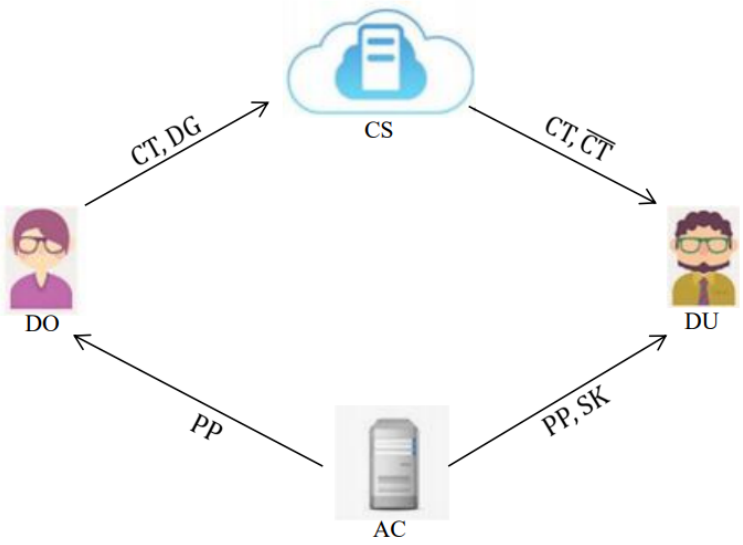

Figure 1: System structure for RABE-DI

$a_1, a_2 \in \mathbb{Z}_p^*$, $s_1, s_2, s \in \mathbb{Z}_p$. The DLIN assumption says, for the PPT adversary $\mathcal{A}$, the advantage of $\mathcal{A}$'s $|\Pr[\mathcal{A}(D, V_0) = 1] - \Pr[\mathcal{A}(D, V_1) = 1]|$ is negligible in λ, where $V_1 = (g^s, h^s)$, $V_0 = (g^{s_1+s_2}, h^{s_1+s_2})$.

3 System Modeling

3.1 System architecture

We will give the roles, formal definitions and security models for each entity in the RABE-DI scheme. Our RABE-DI system requires four entities: Data Owner (DO), Data User (DU), Authority Center (AC), and Cloud Service (CS), which is illustrated in Figure 1.
AC: Responsible for setting up the entire system and generating all public parameters based on security.
DO: Develops the access structure to determine the users who can access the data. File data is encrypted and transferred to the cloud under the specified access structure.
CS: stores the ciphertext uploaded by DO and performs a revocation operation.
DU: Recovers the plaintext using its own key and at the same time can check the integrity of the data.

3.2 Formalized definitions

Our RABE-DI scheme consists of the following algorithms:

1)$Setup(\lambda) \rightarrow (PP, MSK)$: This algorithm generates the public key PP and private key MSK of the AC according to the security parameter λ.

2)$Keygen(MSK, \mathcal{S}) \rightarrow SK$: AC takes as input the master private key MSK and the set of attributes $\mathcal{S}$ corresponding to the DU and outputs the DU's private key SK.

3)$Encrypt(PP, F, \mathbb{A}) \rightarrow CT$: This algorithm is executed by DO, which takes as input the public parameters PP, access structure $\mathbb{A}$ and message F, and outputs the ciphertext CT.

4)$Decrypt_{or}(SK, CT) \rightarrow F$: This algorithm is executed by the DU. the DU takes as input its private key SK and the original ciphertext CT, where the private key corresponds to the set of attributes $\mathcal{S}$, and the ciphertext CT contains the access structure $\mathbb{A}$.

5)$Delegate(\tilde{\mathbb{A}}) \rightarrow DG$: This algorithm is executed by the DO. the DO takes the delegate access structure $\tilde{\mathbb{A}}$ as input, computes based on the new attributes involved in $\tilde{\mathbb{A}}$ and gets the delegate as output.

6)$Revoke(CT, DG) \rightarrow \overline{CT}$: This algorithm is executed by CS, which takes as input the original ciphertext CT and the delegated DG, and takes as output the revoked ciphertext $\overline{CT}$ with the corresponding access structure $\overline{\mathbb{A}}$.

7)$Decrypt_{re}(\overline{SK}, \overline{CT}, csum) \rightarrow F$: This algorithm is executed by a DU. DU inputs a private key $\overline{SK}$ different from SK and a checksum $csum$, and outputs a message F or $\perp$.

3.3 Security model

The RABE-DI scheme is required to ensure both IND-CPA (Indistinguishability under Chosen-Plaintext Attack) security and data integrity.

IND-CPA Security: RABE-DI scheme resists the adaptively IND-CPA if the advantage of $\mathcal{A}$ in the following game is negligible.

Setup: The challenger $\mathcal{B}$ generates the system parameter PP and the master key MSK, and sends PP to $\mathcal{A}$.

Query 1: $\mathcal{A}$ sends a set of attribute $\mathcal{S}$. $\mathcal{B}$ runs key generation algorithm to retrieve a key $SK_{\mathcal{S}}$, and sends $SK_{\mathcal{S}}$ to $\mathcal{A}$.

Challenge: $\mathcal{A}$ submits two messages F_0, F_1 and access structure $\mathbb{A}^*$.It is required that for all $\mathcal{S}$($\mathcal{S}$ represents the set of attributes that were inquired about during the private key inquiry stage) is unsatisfiable for $\mathbb{A}^*$. $\mathcal{B}$ randomly chooses $\theta \in \{0, 1\}$ and encrypts F_θ to get the challenge ciphertext CT^*, which is returned to $\mathcal{A}$.

Query 2: The same as query 1.

Guess: $\mathcal{A}$ outputs its guess $\theta' \in \{0, 1\}$ for θ. If $\theta' = \theta$, the adversary $\mathcal{A}$ wins.

Integrity: RABE-DI scheme resis $\mathcal{A}$ in the following game is negligible.

Setup: The challenger $\mathcal{B}$ generates the public parameter PP and the master key MSK, and sends PP to $\mathcal{A}$.

Query 1: $\mathcal{A}$ performs a private key query based on the user attribute set $\mathcal{S}$. $\mathcal{B}$ runs key generation algorithm to obtain a key $SK_{\mathcal{S}}$, and sends $SK_{\mathcal{S}}$ to $\mathcal{A}$.

Challenge: $\mathcal{A}$ submits a message F and an access structure $\mathbb{A}^*$. $\mathcal{B}$ encypts F use $\mathbb{A}^*$ to obtain CT, and returns to $\mathcal{A}$.

Query 2: The same as query 1.

Output: $\mathcal{A}$ outputs $\mathcal{S}'$ and CT'. $\mathcal{A}$ wins if $Dec_{re}\left(SK_{\mathcal{S}'}, CT, CT'\right) \notin \{F, \perp\}$.

4 Specific construction of our scheme

4.1 Algorithm design

1) $Setup(\lambda) \rightarrow (PP, MSK)$: AC Runs group generator G, outputs $(\mathbb{G}, \mathbb{H}, \mathbb{G}_T, h, g, e, p)$.
AC selects φ,$\phi \in \mathbb{G}$, $a_1, a_2, d_1, d_2 \in \mathbb{Z}_p^*$, $b_1, b_2, b_3 \in \mathbb{Z}_p$, two hash functions $\mathcal{H}_1 : \{0,1\}^* \rightarrow \mathbb{G}$ and

$\mathcal{H}_2 : \mathbb{G}_T \to \mathbb{Z}_p^*$. AC calculates $T_1 = e(g,h)^{b_1 a_1 + b_3}$, $H_1 = h^{a_1}$, $T_2 = e(g,h)^{b_2 a_2 + b_3}$, $H_2 = h^{a_2}$, and outputs public parameter and master private keys $PP = (T_2, \mathcal{H}_2, H_2, g, \varphi, T_1, \mathcal{H}_1, H_1, h, \phi)$, $MSK = (a_1, a_2, d_1, d_2, g^{b_1}, g^{b_2}, g^{b_3})$.

2)$Keygen(MSK, \mathcal{S}) \to SK$: AC inputs an attribute set $\mathcal{S}$ and MSK. AC random selects $r_1, r_2, \sigma_y, \sigma' \in \mathbb{Z}_p$, $y \in \mathcal{S}$, $t = 1, 2$, calculates $sk_y = (sk_{y,1}, sk_{y,2}, g^{-\sigma_y})$, $sk_{y,t} = \mathcal{H}_1(y1t)^{\frac{d_1 r_1}{a_t}} \mathcal{H}_1(y2t)^{\frac{d_2 r_2}{a_t}} \mathcal{H}_1(y3t)^{\frac{r_1 + r_2}{a_t}} g^{\frac{\sigma_y}{a_t}}$, $sk'_t = g^{b_t} \cdot \mathcal{H}_1(011t)^{\frac{d_1 r_1}{a_t}} \cdot \mathcal{H}_1(012t)^{\frac{d_2 r_2}{a_t}} \cdot \mathcal{H}_1(013t)^{\frac{r_1 + r_2}{a_t}} \cdot g^{\frac{\sigma'}{a_t}}$, $sk' = \left(sk'_1, sk'_2, g^{b_3 - \sigma'}\right)$, $sk_0 = \left(h^{d_1 r_1}, h^{d_2 r_2}, h^{r_1 + r_2}\right) = (sk_{0,1}, sk_{0,2}, sk_{0,3})$. AC outputs private key $SK = \left(\mathcal{S}, sk_0, \{sk_y\}_{y \in \mathcal{S}}, sk'\right)$, and sends SK to the DU.

3)$Encrypt(PP, F, \mathbb{A}) \to CT$: DO inputs shared data files $F \in \mathbb{G}_T$, public keys PP and an access structure $\mathbb{A} = (M, \tau)$. M is a matrix with n_1 row and n_2 columns, and M_{ij} is the element of the i-th row and j-th column of M. $\tau : \{1, \cdots, n_1\} \to \mathcal{S}$ is maps the rows of $M_{n_1 \times n_2}$ to the attributes using linear secret sharing [13]. First DO compute $CF = Enc_{ck}(F)$, where the encryption algorithm is AES, $ck \in \mathbb{G}_T$ is the symmetric key. DO random selects $s_1, s_2 \in \mathbb{Z}_p, k = 1, 2, 3, i = 1, \ldots, n_1$ and calculates $csum = \varphi^{\mathcal{H}_2(F)} \phi^{\mathcal{H}_2(ck)}$, $ct' = T_1^{s_1} \cdot T_2^{s_2} \cdot ck$, $c_0 = (H_1^{s_1}, H_2^{s_2}, h^{s_1 + s_2}) = (c_{0,1}, c_{0,2}, c_{0,3})$, $c_i = (c_{i,1}, c_{i,2}, c_{i,3})$, $c_{i,k} = \prod_{t=1}^{2} \mathcal{H}_1(\tau(i)\, kt)^{s_t} \cdot \prod_{j=1}^{n_2} \left[\prod_{t=1}^{2} \mathcal{H}_1(0jkt)^{s_t}\right]^{M_{i,j}}$, then sign $csum$ to get $sign(csum)$. DO outputs ciphertext $CT = (\mathbb{A}, CF, c_0, c_1, \ldots, c_{n_1}, ct', csum || sign(csum))$, and sends CT to the CS.

4)$Decrypt_{or}(SK, CT) \to F$: DU first verifies whether $sign(csum)$ is the signature of the DO. If the validation passes, determine if the attribute set $\mathcal{S}$ meets the access structure $\mathbb{A}$. If it satisfied then the DU finds a set of constants $\{\theta_i\}_{i \in I}$ and the set $I = \{j : \tau(j) \in \mathcal{S}\} \subset \{1, 2, \ldots, n_1\}$ such that $\sum_{i \in I} \theta_i \cdot M_i = (1, 0, \ldots, 0)$, where the j-th component of $\sum_{i \in I} \theta_i \cdot M_i$ is expressed as $\sum_{i \in I} \theta_i \cdot M_{ij}$. DU inputs private key SK, ciphertext CT and calculates $ck = Q/P$, $Q = ct' \cdot \prod_{k=1}^{3} e\left(\prod_{i \in I} c_{i,k}^{\theta_i}, sk_{0,k}\right)$, $P = \prod_{k=1}^{3} e\left(ct_{0,k}, sk'_k \cdot \prod_{i \in I} sk_{\tau(i),k}^{\theta_i}\right)$. DU outputs ck and further decrypts CF to get the shared file F, and then verify whether $csum = \varphi^{\mathcal{H}_2(F)} \phi^{\mathcal{H}_2(ck)}$ holds, if so outputs F.

5)$Delegate(\tilde{\mathbb{A}}) \to DG$: DO inputs specified access structure $\tilde{\mathbb{A}} = \left(\tilde{M}_{\widetilde{n_1} \times \widetilde{n_2}}, \tilde{\tau}\right)$ and calculates delegate $dt_k = \prod_{i=1}^{\widetilde{n_1}} \mathcal{H}_1(\widetilde{\tau}(i)\, k1)^{s_1} \cdot \mathcal{H}_1(\widetilde{\tau}(i)\, k1)^{s_2}$, $dt = (dt_1, dt_2, dt_3)$, outputs delegate: $DG = \left(dt, \tilde{\mathbb{A}}\right)$.

6)$Revoke(CT, DG) \to \overline{CT}$: CS inputs ciphertext CT and delegate DG generates revoked ciphertext $\overline{CT}$ under access structure $\overline{\mathbb{A}} = (M', \tau')$. (Here, $\mathbb{A}$ correspond (M, τ), $\tilde{\mathbb{A}}$ correspond $\left(\tilde{M}, \tilde{\tau}\right)$. $\overline{\mathbb{A}} = \left(\mathbb{A} AND \tilde{\mathbb{A}}\right)$ corresponds to (M', τ'). Chen et al. [14] proved that for the LSSS scheme, access structure (M', τ') is a valid. Therefore, $\overline{CT}$ is a valid revocation ciphertext.) CS random selects $s'_1, s'_2 \in \mathbb{Z}_p$, calculates the revoked ciphertext: $\overline{ct'} = ct' \cdot T_1^{s'_1} \cdot T_2^{s'_2}$, $\overline{c_0} = \left(H_1^{s'_1} \cdot c_{0,1}, H_2^{s'_2} \cdot c_{0,2}, h^{s'_1 + s'_2} \cdot c_{0,3}\right)$, $\overline{c_i} = (\overline{c}_{i,1}, \overline{c}_{i,2}, \overline{c}_{i,3}), i \in [1, n'_1]$, $\overline{c_{i,k}} = \prod_{j=1}^{n'_2} \left[\prod_{t=1}^{2} \mathcal{H}_1(0jkt)^{s'_t}\right]^{M'_{i,j}} \cdot c_{i,k} \cdot \prod_{t=1}^{2} \mathcal{H}_1(\tau(i)\, kt)^{s'_t}, i \in [1, n_1]$, $\overline{c_{i,k}} = dt_k \cdot \prod_{t=1}^{2} \mathcal{H}_1(\tau(i)\, kt)^{s'_t} \cdot$

$\prod_{j=1}^{n'_2}\left[\prod_{t=1}^{2}\mathcal{H}_1(0jkt)^{s'_t}\right]^{M'_{i,j}}, i\in[n_1+1,n'_1]$. CS calculates the checksum: $\overline{csum}=csum$, outputs revoked ciphertext: $\overline{CT}=\left(\overline{\mathbb{A}},CF,\overline{c}_0,\overline{c}_1,\ldots,\overline{c}_{n'_1},\overline{ct'},\overline{csum}\right)$.

7)$Decrypt_{re}(SK',\overline{CT},csum||sign\,(csum))\rightarrow F$: DU inputs private key SK', attribute set $\mathcal{S}'$. DU first verifies whether $sign\,(csum)$ is the signature of the DO. If the validation passes, then verifies that $\overline{csum}=csum$ holds and outputs $\perp$ if it does not. Otherwise, if satisfied then finds the set $I'=\{j:\tau'(j)\in\mathcal{S}'\}\subset\{1,2,\ldots,n_1{}'\}$ and a set of constants $\{\theta'_i\}_{i\in I'}$ that $\sum_{i\in I'}\theta'_i\cdot M'=(1,0,0,\ldots,0)$, DU computes $ck=Q'/P'$, $Q'=\overline{ct'}\cdot\prod_{k=1}^{3}e\left(sk_{0,k},\prod_{i\in I}c_{i,k}^{\theta'_i}\right)$, $P'=\prod_{k=1}^{3}e\left(ct_{0,k},sk'_k\cdot\prod_{i\in I}sk_{\tau'(i),k}^{\theta_i{}'}\right)$, DU by ck can be further decrypted to get the shared file F, and then verify whether $\overline{csum}=\varphi^{\mathcal{H}_2(F)}\phi^{\mathcal{H}_2(ck)}$, if holds then outputs F.

Note: We use a secure digital signature $sign\,(csum)$ to ensure the integrity of $csum$.

4.2 Correctness analysis

In the $Decrypt_{or}$ algorithm:

$$\begin{aligned}\prod_{i\in I}c_{i,k}^{\theta_i} &= \prod_{i\in I}\left(\prod_{t=1}^{2}\mathcal{H}_1(\tau\,(i)\,kt)^{\theta_i s_t}\cdot\prod_{j=1}^{n_2}\left[\prod_{t=1}^{2}\mathcal{H}_1(0jkt)^{s_t}\right]^{\theta_i M_{i,j}}\right) \\ &= \prod_{i\in I}\left(\prod_{t=1}^{2}\mathcal{H}_1(\tau\,(i)\,kt)^{\theta_i s_t}\right)\cdot\prod_{j=1}^{n_2}\left[\prod_{t=1}^{2}\mathcal{H}_1(0jkt)^{s_t}\right]^{\sum_{i\in I}\theta_i M_{i,j}} \\ &= \prod_{i\in I}\left(\prod_{t=1}^{2}\mathcal{H}_1(\tau\,(i)\,kt)^{\theta_i s_t}\right)\cdot\left[\prod_{t=1}^{2}\mathcal{H}_1(0jkt)^{s_t}\right]^{\sum_{i\in I}\theta_i M_{i,j}}\end{aligned} \tag{1}$$

$$\begin{aligned}\frac{Q}{ct'} &= \prod_{k=1}^{3}e\left(\prod_{i\in I}c_{i,k}^{\theta_i},sk_{0,k}\right) \\ &= \prod_{k=1}^{3}e\left(\prod_{i\in I}\left(\prod_{t=1}^{2}\mathcal{H}_1(\tau\,(i)\,kt)^{\theta_i s_t}\right)\cdot\left[\prod_{t=1}^{2}\mathcal{H}_1(01kt)^{s_t}\right]^{\sum_{i\in I}\theta_i M_{i,j}},sk_{0,k}\right) \\ &= \prod_{t=1}^{2}\Big[e(h,\mathcal{H}_1\,(013t))^{(r_1+r_2)s_t}\cdot e(h,\mathcal{H}_1\,(011t))^{d_1r_1s_t}\cdot e(h,\mathcal{H}_1\,(012t))^{d_2r_2s_t} \\ &\cdot \prod_{i\in I}e\Big(h,\mathcal{H}_1(\tau\,(i)\,3t)^{\theta_i}\Big)^{(r_1+r_2)s_t}\cdot e\Big(h,\mathcal{H}_1(\tau\,(i)\,1t)^{\theta_i}\Big)^{d_1r_1s_t} \\ &\cdot e\Big(h,\mathcal{H}_1(\tau\,(i)\,2t)^{\theta_i}\Big)^{d_2r_2s_t}\end{aligned} \tag{2}$$

$$\prod_{i\in I}sk_{\tau(i),k}^{\theta_i}=\prod_{i\in I}\left(\mathcal{H}_1(\tau\,(i)\,1k)^{\frac{d_1r_1}{a_k}}\cdot\mathcal{H}_1(\tau\,(i)\,2k)^{\frac{d_2r_2}{a_k}}\cdot\mathcal{H}_1(y3t)^{\frac{r_1+r_2}{a_k}}\cdot g^{\frac{\sigma_y}{a_k}}\right)^{\theta_i} \tag{3}$$

$$
\begin{aligned}
P/\frac{Q}{ct'} &= \prod_{k=1}^{3} e\left(ct_{0,k}, sk'_k \cdot \prod_{i\in I} sk_{\tau(i),k}^{\theta_i}\right) / \prod_{t=1}^{2}\Big[e(h,\mathcal{H}_1(013t))^{(r_1+r_2)s_t} \\
&\cdot e(h,\mathcal{H}_1(011t))^{d_1 r_1 s_t} \cdot e(h,\mathcal{H}_1(012t))^{d_2 r_2 s_t} \cdot \prod_{i\in I} e\Big(h,\mathcal{H}_1(\tau(i)\,3t)^{\theta_i}\Big)^{(r_1+r_2)s_t} \\
&\cdot e\Big(h,\mathcal{H}_1(\tau(i)\,1t)^{\theta_i}\Big)^{d_1 r_1 s_t} \cdot e\Big(h,\mathcal{H}_1(\tau(i)\,2t)^{\theta_i}\Big)^{d_2 r_2 s_t}\Big] \\
&= \left(\prod_{t=1}^{2} e\left(h^{a_t s_t}, g^{b_t} \cdot g^{\frac{\sigma'}{a_t}} \cdot \prod_{i\in I} g^{\frac{\theta_i \sigma_{\tau(i)}}{a_t}}\right)\right) \cdot e\left(h^{s_1+s_2}, g^{b_3} g^{-\sigma'} \cdot \prod_{i\in I} g^{-\theta_i \sigma_{\tau(i)}}\right) \\
&= e(g,h)^{b_1 a_1 s_1 + b_2 a_2 s_2 + b_3(s_1+s_2)} \\
&= T_1^{s_1} \cdot T_2^{s_2}.
\end{aligned} \tag{4}
$$

Thus the symmetric key ck can be computed: $ck = \frac{ct'}{T_1^{s_1} \cdot T_2^{s_2}} = Q/P$.

5 Performance evaluation

5.1 Security analysis

Theorem 1: Under the DLIN assumption, the proposed RABE-DI scheme is adaptive chosen plaintext attack secure.

Proof: The security of RABE-DI is proven through a sequence of indistinguishable games. This proof framework constructs a series of game transitions. Each game modifies the private key or challenges the ciphertext in a controllable way, thereby proving to demonstrate that any adversary's advantage in distinguishing between real encrypted messages and simulated ones remains negligible.

Game_0 is the security game defined on page 4. The ciphertext and key are in normal form. We use a sampling algorithm(see the appendix for details) to modify the system parameters to get Game_1. After that, one by one, we modify the form of the ciphertext and the private key to pseudo-normal (P-normal) or semi-functional (SF*) by adding or subtracting random numbers. Eventually, in Game_4, both the key and the ciphertext are transformed into a SF state. These changes can be shown to be indistinguishable to the adversary under the DLIN assumption (See the Lemma in the appendix for details). Since in a dual system, a SF ciphertext cannot be decrypted by a SF* key [26], the challenge ciphertext cannot be decrypted all keys in Game_4. Finally in Game_5 the challenge ciphertext is modified to be the ciphertext Rnd* of a random message, which is still indistinguishable from the adversary. Thus, it is shown that the advantage of adversary's success is negligible. Let the adversary perform a total of $q = 1, 2, \cdots, Q$ key queries. The game procedure is shown in Figure 2.

Theorem 2: Under the DL assumption, the proposed RABE-DI scheme satisfies data integrity.

Proof: Suppose there exists a PPT adversary $\mathcal{A}$ with non-negligible advantage in violating the data integrity of our RABE-DI scheme. Then, we can construct a simulator that breaks the DL assumption. Specific proof is shown in Figure 3.

Game	Challenge Ciphertext	Key Query					
		1	2	⋯	i	⋯	Q
$Game_0$	Normal						
$Game_1$	Normal*	Normal					
$Game_{2,1,1}$	Normal*	P-normal	Normal				
$Game_{2,1,2}$	Normal*	P-normal*	Normal				
$Game_{2,1,3}$	Normal*	Normal*	Normal				
$Game_{2,2,1}$	Normal*	Normal*	P-normal	Normal			
$Game_{2,2,2}$	Normal*	Normal*	P-normal*	Normal			
$Game_{2,2,3}$	Normal*	Normal*		Normal			
⋯	⋯						
$Game_{2,i,1}$	Normal*	Normal*			P-normal	Normal	
$Game_{2,i,2}$	Normal*	Normal*			P-normal*	Normal	
$Game_{2,i,3}$	Normal*	Normal*				Normal	
⋯	⋯						
$Game_{2,Q,1}$	Normal*	Normal*					P-normal
$Game_{2,Q,2}$	Normal*	Normal*					P-normal*
$Game_{2,Q,3}$	Normal*	Normal*					
$Game_3$	SF*	Normal*					
$Game_{4,1,1}$	SF*	P-normal*	Normal*				
$Game_{4,1,2}$	SF*	P-SF*	Normal*				
$Game_{4,1,3}$	SF*	SF*	Normal*				
$Game_{4,2,1}$	SF*	SF*	P-normal*	Normal*			
$Game_{4,2,2}$	SF*	SF*	P-SF*	Normal*			
$Game_{4,2,3}$	SF*	SF*		Normal*			
⋯		⋯					
$Game_{4,i,1}$	SF*	SF*			P-normal*	Normal*	
$Game_{4,i,2}$	SF*	SF*			P-SF*	Normal*	
$Game_{4,i,3}$	SF*	SF*				Normal*	
⋯		⋯					
$Game_{4,Q,1}$	SF*	SF*					P-normal*
$Game_{4,Q,2}$	SF*	SF*					P-SF*
$Game_{4,Q,3}$	SF*	SF*					
$Game_5$	Rnd*	SF*					

Figure 2: Changes in ciphertexts and keys.

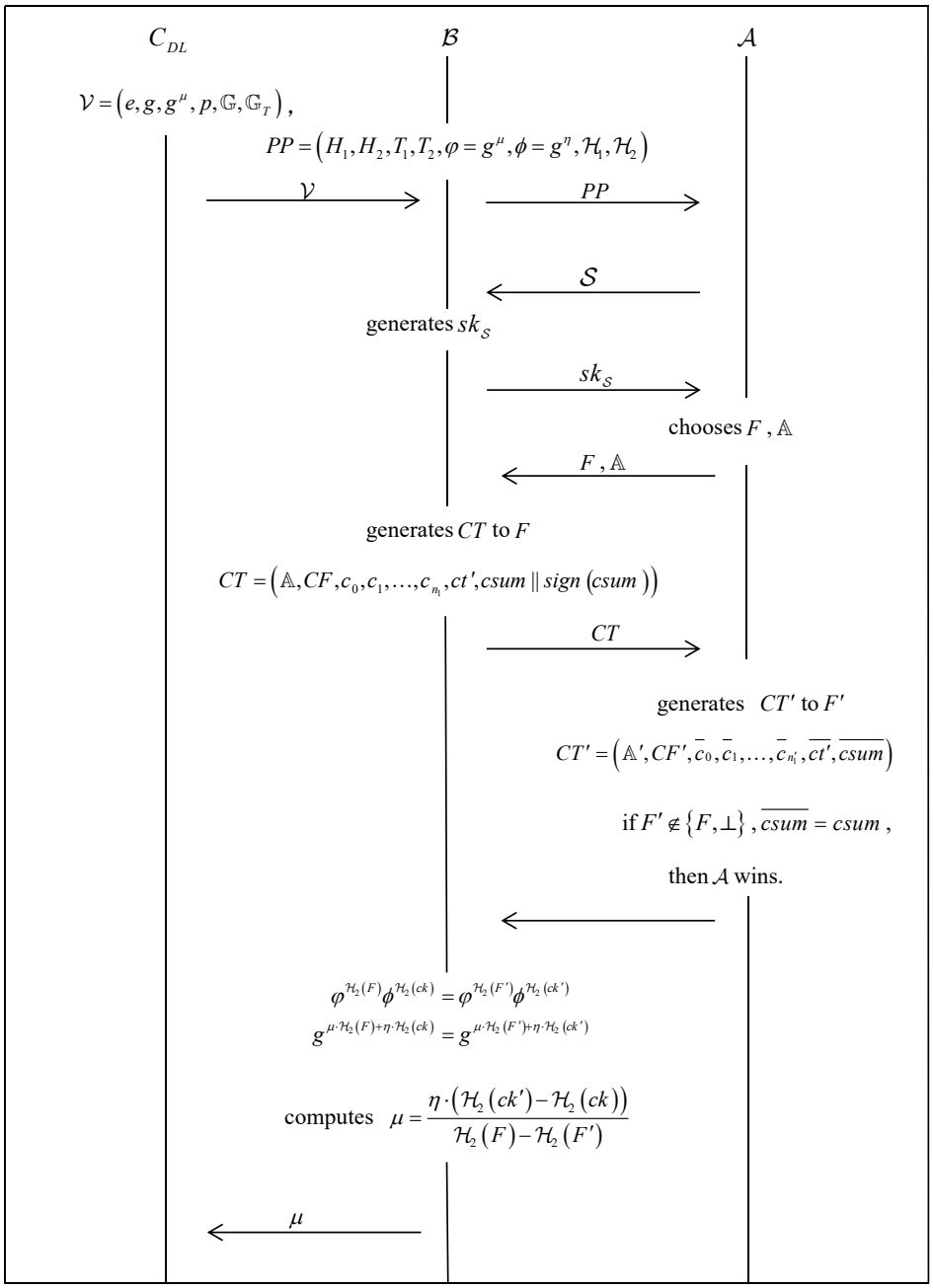

Figure 3: Ciphertext indistinguishability reduction.

Scheme	Key Generation	Encryption	Decryption	Revocation
Ge et al. [14]	M_1+ $(3+u)E_1$	M_1+ $(3+u)E_1$	$(2u+8)\,E_T$ $+(6+2u)\,P$ $+4H$	$(6n_1+4)\,M_1+$ $(12n_1+8)E_1+$ $4E_T+4P$
Chen et al. [15]	$(6u+8)M_1$ $+(8u+12)\,E_1$ $+(6u+6)H$ $+3E_2$	$(12n_1n_2+6n_1)M_1$ $+(12n_1)\,E_1+6E_2+$ $(12n_1+12n_2+4)H$	$(12u+6)M_1$ $+12P$	$(6n_1n_2+12n_1$ $+10)M_1+6E_2+$ $(12n_1+12n_2)\,E_1+$ $(12n_1+12n_2)H$
Huang et al. [23]	M_1+ $(3+u)E_1$	$(n_1+2)\,M_1$ $+(3n_1+1)E_1$ $+2E_T+P$ $+2H$	$(u+4)\,E_T$ $+(3+u)\,P$ $+2H$	$(3n_1+2)\,M_1+$ $(6n_1+4)E_1+$ $2E_T+2P$
Our scheme	$(6u+8)M_1$ $+(8u+12)\,E_1$ $+(6u+6)H$ $+3E_2$	$(6n_1n_2+3n_1)M_1+$ $(6n_1)\,E_1+3E_2+$ $(6n_1+6n_2+2)H$	$(6u+3)M_1$ $+6P$	$(3n_1n_2+6n_1$ $+5)M_1+3E_2+$ $(6n_1+6n_2)\,E_1+$ $(6n_1+6n_2)H$

Figure 4: Comparison of computational efficiency.

5.2 Performance analysis

As shown in Figure 4, we compare the computational overhead of encryption, decryption, key generation, and revocation processes between our scheme and those in literature [14], [15], [23]. To facilitate the comparison, some notations are defined below: u expresses the number of attributes, n_1 expresses the number of rows in the access matrix in LSSS, n_2 expresses the number of columns in the access matrix in LSSS, E_1 expresses the exponential operation in the group $\mathbb{G}$, E_2 expresses the exponential operation in the group $\mathbb{H}$, E_T expresses the exponential operation in the group $\mathbb{G}_T$, P expresses a bilinear pair operation, M_1 expresses a multiplication operation in the group $\mathbb{G}$, and M_2 expresses a multiplication operation in the group $\mathbb{G}_T$.

As illustrated in Figure 4, during the key generation phase, the computational overhead of our scheme and comparative approaches correlates with the number of attributes. The key generation time of our scheme is the same as that in [15]. Although the time is slower than that of [14] and [23], their scheme only achieves selective security. Moreover, their scheme can only be implemented on a very small attribute domain and is not as practical as ours. In the encryption phase, our scheme demonstrates lower computational overhead than the approaches in literature [14] and [15], while being comparable to the scheme in [23]. In the decryption process, our scheme eliminates the need for exponential operations. Meanwhile, the number of bilinear pairing operations is only six (fewer than that in literature [15]), which does not increase with the number of attributes. However, the schemes in literature [14] and [23] require bilinear pairing operations dependent on the number of attributes, which significantly reduces computational efficiency. In the attribute revocable phase, the computational overhead of our scheme is lower compared to literature [14], [15], [23].

5.3 Comparison of experiments

In this section, in order to better evaluate the performance of our scheme, we use JAVA language for programming, and conduct simulation experiments with our scheme and the literature [14], [15], [23]. The experimental environment is configured with Intel (R) Core (TM) i5-8265U, CPU 1.60 GHz, 8.0 GB RAM, and Windows 10 operating system. We use A-type elliptic curves,

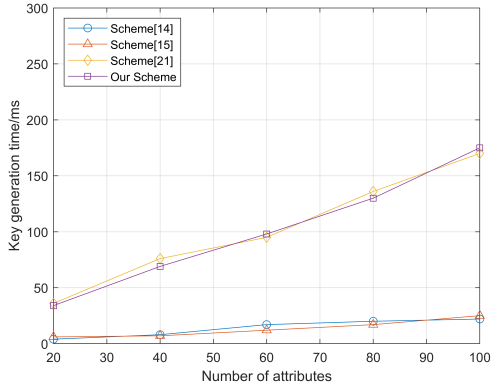

Figure 5: Key generation time

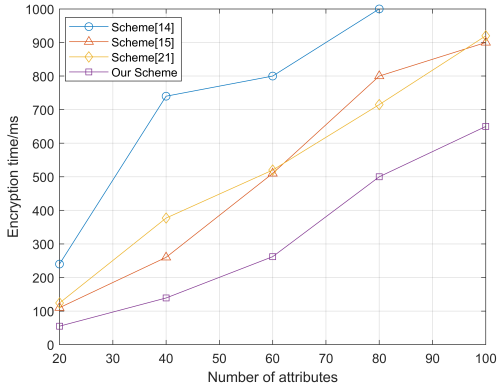

Figure 6: Encryption time

IntelliJIDEA2024 tool and JPBC 2.0 library for simulation during the experiments.

We conducted simulation experiments on four metrics: key generation, encryption, decryption, and revocation times. With access structure attribute counts set to 20, 40, 60, 80, and 100 to model varying complexity scenarios, each experiment group was run 100 times. Averaging the results ensured reliable and accurate conclusions. The generation time of the system key is shown in Figure 5. Compared with the literatures [14] and [23], our computational cost is lower, but our scheme has achieved adaptive IND-CPA security. It is basically consistent with the scheme in reference [15]. Figure 6 shows the system's encryption time. Our scheme outperforms literature [14], [15], and [23] in computational cost, with significantly lower values. Thus, our scheme reduces the computational burden on the user.

The system decryption time is shown in Figure 7. It can be seen that our proposed scheme has a clear advantage over the literature [14] and [23] in terms of computational cost, which is significantly lower. Although the cost of the literature [15] looks similar to our computational cost, the computational cost of the literature [15] is actually double that of our. Figure 8 depicts the system's revocation time. Our scheme outperforms literature [14], [15], and [23] in computational cost, with notably lower values, demonstrating higher efficiency.

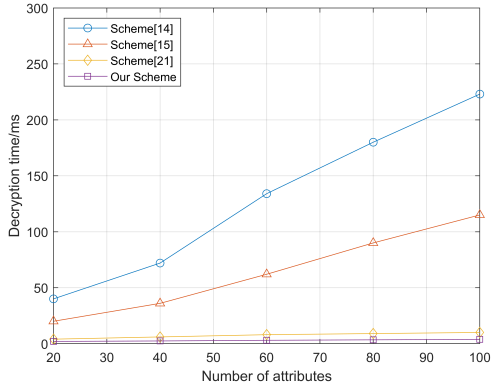

Figure 7: Decryption time

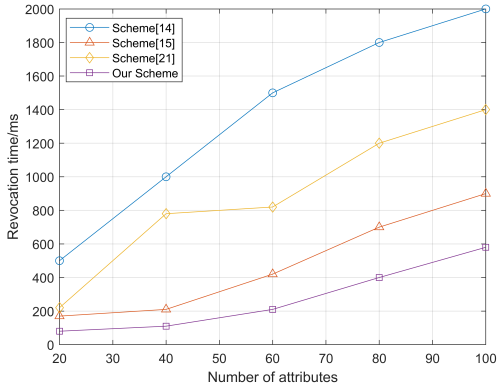

Figure 8: Revocation time

6 Conclusions

Our scheme proposes an efficient RABE scheme supporting data integrity. Leveraging a ciphertext delegation algorithm for revocation, it enables users to verify plaintext consistency between updated ciphertexts and their original counterparts. While achieving efficient revocable, the scheme also guarantees data integrity. Based on the fast attribute message encryption in [10], the computational efficiency of the existing RABE-DI has been improved by introducing symmetric encryption. The final simulation experiment shows the efficiency of our scheme.

7 Acknowledgments

This work was supported by the "Scientist + Engineer" Team Construction Project of Qin Chuang Yuan in Shaanxi Province under Grant No.2025QCY-KXJ-168, the Natural Science Basic Research Program of Shaanxi Province under Grant No.2024JC-YBMS-537.

References

[1] Zhang Yinghui, Chen Xiaofeng, Li Jin, Wong Duncan S., Li Hui, and Li Ilsun. Ensuring attribute privacy protection and fast decryption for outsourced data security in mobile cloud computing. https://www.sciencedirect.com/, 2017.

[2] Dayananda R B, Someswar G Manoj, and Rao T P Suryachandra. Secure data sharing with abe in wireless sensor networks. https://www.compusoftgroup.com/, 2015.

[3] Sahai Amit and Waters Brent. Fuzzy identity-based encryption. https://idp.springer.com/, 2005.

[4] Goyal Vipul, Pandey Omkant, Sahai Amit, and Waters Brent. Attribute-based encryption for fine-grained access control of encrypted data. https://archive.org/, 2006.

[5] Waters Brent. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. https://idp.springer.com/, 2011.

[6] Frikken Keith B., Atallah M. J., and Li Jin. Attribute-based access control with hidden policies and hidden credentials. https://www.computer.org/, 2006.

[7] Sahai Amit, Seyalioglu Hussein, and Waters Brent. Dynamic credentials and ciphertext delegation for attribute-based encryption. https://idp.springer.com/, 2012.

[8] Li Jin, Huang Xinyi, Li Jingwei, Chen Xiaofeng, and Xiang Yan. Securely outsourcing attribute-based encryption with check ability. https://cn.ieee.org/, 2006.

[9] Ambrona Marc, Barthe Gilles, Gay Roger, and Wee Hoeteck. Attribute-based encryption in the generic group model: Automated proofs and new constructions. https://www.sigsac.org/, 2017.

[10] Agrawal Shweta and Chase Melissa. Fame: Fast attribute-based message encryption. https://www.sigsac.org/, 2017.

[11] Wu Zijian, Cui Zhenhua, and Wang Chunyan. Access control scheme with attribute revocation for swim. https://www.sciencedirect.com/, 2017.

[12] Wang Panpan, Feng Deguang, and Zhang Linwei. Cp-abe scheme supporting fully fine-grained attribute revocation. https://www.jsoftware.us/, 2012.

[13] Pirretti Misty, Traynor Patrick, McDaniel Patrick, and Waters Brent. Secure attribute-based systems. https://www.sigsac.org/, 2006.

[14] Ge Chunpeng, Susilo Willy, Baek Joonsang, Liu Zhe, Xia Jinyue, and Fang Liming. Revocable attribute-based encryption with data integrity in clouds. https://cn.ieee.org/, 2022.

[15] Chen Shaobo, Li Jiguo, Zhang Yichen, and Han Jinguang. Efficient revocable attribute-based encryption with verifiable data integrity. https://cn.ieee.org/, 2024.

[16] Pratish Datta, Ratna Dutta, and Sourav Mukhopadhyay. General circuit realizing compact revocable attribute-based encryption from multilinear maps. https://link.springer.com/, 2015.

[17] Kotoko YAMADA, Nuttapong ATTRAPADUNG, Keita EMURA, Goichiro HANAOKA, and Keisuke TANAKA. Generic constructions for fully secure revocable attribute-based encryption. https://www.jstage.jst.go.jp/, 2018.

[18] Wang Guobin and Wang Jianhua. Research on ciphertext-policy attribute-based encryption with attribute level user revocation in cloud storage. https://www.hindawi.com/, 2017.

[19] Hur Junbeom and Noh Donghoon. Attribute-based access control with efficient revocation in data outsourcing systems. https://cn.ieee.org/, 2011.

[20] Li Jiguo, Lin Xiaonan, Zhang Yichen, and Han Jinguang. User collusion avoidance cp-abe with efficient attribute revocation for cloud storage. https://cn.ieee.org/, 2018.

[21] Tu Shanshan, Waqas Muhammad, Huang Fengming, Abbas Ghulam, and Abbas Ziaul Haq. A revocable and outsourced multi-authority attribute-based encryption scheme in fog computing. https://www.sciencedirect.com/, 2021.

[22] Qin Baodong, Zhao Qinglan, Zheng Dong, and Cui Hui. (dual) server-aided revocable attribute-basedencryption with decryption key exposure resistance. https://link.springer.com/, 2019.

[23] Huang Meijuan, Liu Yutian, Yang Bo, Zhao Yanqi, and Zhang Mingrui. Efficient revocable attribute-based encryption with data integrity and key escrow-free. https://www.mdpi.com/, 2024.

[24] Shao Jun, Lu Rongxing, Lin Xiaodong, and Liang Kaitai. Secure bidirectional proxy reencryption for cryptographic cloud storage. https://www.sciencedirect.com/, 2016.

[25] Yu Shucheng, Wang Cong, and Lou Wenjing. Attribute based data sharing with attribute revocation. https://cn.ieee.org/, 2010.

[26] Waters Brent. Realizing fully secure ibe and hibe under simple assumptions. https://link.springer.com/, 2009.

8 Appendix

8.1 Dual system group

Let $a, b, c \in \mathbb{Z}_P$, then $[a]_1$ expressed as g^a, $[b]_2$ expressed asexpressed as h^b, $[c]_T$ expressed as $e(g,h)^c$. Take the vector $\vec{d} = (d_1, d_2, \cdots, d_n)^\top$, $\left[\vec{d}\right]_1$ expressed as $\left(g^{d_1}, g^{d_2}, \cdots, g^{d_n}\right)^\top$. Take the matrix $\mathbf{E} = \begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{bmatrix}$, $[\mathbf{E}]_1$ expressed as $\begin{bmatrix} g^{e_{11}} & g^{e_{12}} \\ g^{e_{21}} & g^{e_{22}} \end{bmatrix}$, and define $e\left([\mathbf{X}]_1, [\mathbf{Y}]_2\right) = \left[\mathbf{X}^\top \mathbf{Y}\right]_T$.

8.2 Sampling Algorithm

Let $v_1, v_2 \in \mathbb{Z}_P^*$, input p is a prime, outputs

$$\mathbf{C} = \begin{bmatrix} v_1 & 0 \\ 0 & v_2 \\ 1 & 1 \end{bmatrix}, \quad c^\perp = \begin{bmatrix} {v_1}^{-1} \\ {v_2}^{-1} \\ -1 \end{bmatrix},$$

$[\mathbf{A}||\mathbf{B}]$ expresses the column concatenation of matrices $\mathbf{A}$ and $\mathbf{B}$, then $\left[\mathbf{C}||c^\perp\right]$ is a full-rank matrix. The matrix $\mathbf{C}$ has the same distribution as $\mathbf{A}$ in the DLIN, so that there are $\mathbf{C}^{\mathrm{T}} c^\perp = 0$.

Lemma 1: $\left(\mathbf{C}_1, c_1^\perp\right), \left(\mathbf{C}_2, c_2^\perp\right) \in Samp(p)$. Then with probability $1 - 1/p$, it holds that $\left[\mathbf{C}_1||c_1^\perp\right]$ and $\left[\mathbf{C}_2||c_2^\perp\right]$ are full-rank matrices as well as $\left\langle c_1^\perp, c_2^\perp \right\rangle \neq 0$.

Game_1 as follows:

Setup: Given a prime order bilinear system group $(p, \mathbb{G}, \mathbb{H}, \mathbb{G}_T, g, h, e)$. $\mathcal{B}$ randomly chosen $b_1, b_2, b_3 \in \mathbb{Z}_p$, $\left(\mathbf{X}, x^\perp\right), \left(\mathbf{Y}, y^\perp\right) \in Samp\left(p\right)$, there are $\mathrm{b} = (b_1, b_2, b_3)^{\mathrm{T}}$, so $MSK = \left(h, g, \mathbf{X}, \mathbf{Y}, [\mathrm{b}]_1\right), PP = \left([\mathbf{X}]_2, \left[\mathrm{b}^{\mathrm{T}}\mathbf{X}\right]_T\right)$.

$\mathcal{B}$ simulates the behavior of the hash function $\mathcal{H}_1$ by maintaining two lists L and Q. List L stores items of the form $\left(x, \mathbf{W}_{x_{3\times 3}}\right)$ or $\left(j, \mathbf{U}_{j_{3\times 3}}\right)$, and list Q stores items of the form xkr or $0jkt$ or some other form, where x is an arbitrary binary string, j is a positive integer, and $k \in \{1,2,3\}$, $t \in \{1,2\}$, $r \in \mathbb{G}$. $\mathcal{A}$ The following three types of hash queries can be performed:

(1) The query for input xkt: $\mathcal{B}$ check if there is an item in list Q for (xkt, r), and if so returns r; otherwise check if there is an item in list L for $(x, \mathbf{W}_x)$, and if so computes $r = \left[\left(\mathbf{W}_x^{\mathrm{T}}\mathbf{X}\right)_{k,t}\right]_1$ and add (xkt, r) to list Q and returns r; otherwise select $\mathbf{W}_x \in Z_p$, calculates $r = \left[\left(\mathbf{W}_x^{\mathrm{T}}\mathbf{X}\right)_{k,t}\right]_1$, then add (xkt, r) to list Q, $(x, \mathbf{W}_x)$ to list L, and returns r.

(2) The query for input $0jkt$: $\mathcal{B}$ Check if there is an item in list Q for $(0jkt, r)$, and if so returns r; otherwise check if there is an item in list L for $(j, \mathbf{U}_j)$, and if so computes

form	Private key
Normal	$sk_0 = [\mathbf{Yr}]_2$, $sk_y = \left[\mathbf{W}_y\mathbf{Yr} + \sigma_y\mathbf{a}^{\perp}\right]_1$, $sk' = \left[\mathbf{b} + \mathbf{U}_1\mathbf{Yr} + \sigma'\mathbf{a}^{\perp}\right]_1$
P-normal	$sk_0 = \left[\mathbf{Yr} + \hat{\mathbf{r}}\mathbf{a}^{\perp}\right]_2$, $sk_y = \left[\mathbf{W}_y\mathbf{Yr} + \sigma_y\mathbf{a}^{\perp}\right]_1$, $sk' = \left[\mathbf{b} + \mathbf{U}_1\mathbf{Yr} + \sigma'\mathbf{a}^{\perp}\right]_1$
P-normal*	$sk_0 = \left[\mathbf{Yr} + \hat{\mathbf{r}}\mathbf{a}^{\perp}\right]_2$, $sk_y = [\mathbf{W}_y\mathbf{Yr}]_1$, $sk' = [\mathbf{b} + \mathbf{U}_1\mathbf{Yr}]_1$
Normal*	$sk_0 = [\mathbf{Yr}]_2$, $sk_y = [\mathbf{W}_y\mathbf{Yr}]_1$, $sk' = [\mathbf{b} + \mathbf{U}_1\mathbf{Yr}]_1$
P-SF*	$sk_0 = \left[\mathbf{Yr} + \hat{\mathbf{r}}\mathbf{a}^{\perp}\right]_2$, $sk_y = [\mathbf{W}_y\mathbf{Yr}]_1$, $sk' = \left[\mathbf{b} + \mathbf{U}_1\mathbf{Yr} + \alpha\mathbf{a}^{\perp}\right]_1$
SF*	$sk_0 = [\mathbf{Yr}]_2$, $sk_y = [\mathbf{W}_y\mathbf{Yr}]_1$, $sk' = \left[\mathbf{b} + \mathbf{U}_1\mathbf{Yr} + \alpha\mathbf{a}^{\perp}\right]_1$

Figure 9: A private key can be of the forms.

form	ciphertext
Normal*	$c_0 = [\mathbf{Xs}]_2$, $c_i = \left[\mathbf{W}_{\tau(i)}^{\mathrm{T}}\mathbf{Xs} + \sum_{j=1}^{n_2}(\mathbf{M})_{i,j}\mathbf{U}_J^{\mathrm{T}}\mathbf{Xs}\right]_1$, $ct' = \left[\mathbf{b}^{\mathrm{T}}\mathbf{Xs}\right]_T \cdot ck$
SF*	$c_0 = \left[\mathbf{Xs} + \hat{\mathbf{s}}\mathbf{b}^{\perp}\right]_2$, $c_i = \left[\mathbf{W}_{\tau(i)}^{\mathrm{T}}\mathbf{Xs} + \sum_{j=1}^{n_2}(\mathbf{M})_{i,j}\mathbf{U}_J^{\mathrm{T}}\mathbf{Xs}\right]_1$, $ct' = \left[\mathbf{b}^{\mathrm{T}}\mathbf{Xs}\right]_T \cdot ck$
Rnd*	$c_0 = \left[\mathbf{Xs} + \hat{\mathbf{s}}\mathbf{b}^{\perp}\right]_2$, $c_i = \left[\mathbf{W}_{\tau(i)}^{\mathrm{T}}\mathbf{Xs} + \sum_{j=1}^{n_2}(\mathbf{M})_{i,j}\mathbf{U}_J^{\mathrm{T}}\mathbf{Xs}\right]_1$, $ct' = \left[\mathbf{b}^{\mathrm{T}}\mathbf{Xs}\right]_T \cdot ck^*$

Figure 10: A ciphertext can be of the forms.

$r = \left[\left(\mathbf{U}_j^{\mathrm{T}}\mathbf{X}\right)_{k,t}\right]_1$ and add $(0jkt, r)$ to list Q and returns r; otherwise selects $\mathbf{U}_j \in \mathbb{Z}_p$ and add $(0jkt, r)$ to list L, computes $r = \left[\left(\mathbf{U}_j^{\mathrm{T}}\mathbf{X}\right)_{k,t}\right]_1$, and add $(0jkt, r)$ to list Q and returns r.

(3) Anything else, as a: $\mathcal{B}$ Check if there is an item in list Q for (a, r), and if so returns r; otherwise selects $r' \in \mathbb{G}$ and add $(a, r')$to list Q and returns r'.

Key generation: $\mathcal{A}$perform a private key query, $\mathcal{B}$ retrieves $\mathbf{W}_y$ and $\mathbf{U}_1$ from the list L. $\mathcal{A}$ selects $r_1, r_2, \sigma_y, \sigma' \in \mathbb{Z}_p, \mathrm{r} = (r_1, r_2)^{\mathrm{T}}$, $\mathcal{B}$ computes $sk_0 = [\mathbf{Yr}]_2$, $sk_y = \left[\mathbf{W}_y\mathbf{Yr} + \sigma_y\mathbf{a}^{\perp}\right]_1$, $sk' = \left[\mathbf{b} + \mathbf{U}_1\mathbf{Yr} + \sigma'\mathbf{a}^{\perp}\right]_1$, $SK = \left(sk', \{sk_y\}_{y\in\mathcal{S}}, \mathcal{S}, sk_0\right)$. Then returns SK.

Encryption: $\mathcal{A}$ sends messages F_0, F_1, andaccess structure $\mathbb{A}^*$, $\mathcal{B}$ randomly selects $\theta \in \{0, 1\}$ and computes $CF = Enc_{ck}(F)$, where the encryption algorithm is AES, F_θ is the plaintext data, and $ck \in \mathbb{G}_T$ is the private key. $\mathcal{B}$ retrieves $\left[\left(\mathbf{W}_{\tau(i)}^{\mathrm{T}}\mathbf{X}\right)_{k,t}\right]_1$ and $\left[\left(\mathbf{U}_j^{\mathrm{T}}\mathbf{X}\right)_{k,t}\right]_1$ from the list Q. $\mathcal{A}$ select $s_1, s_2 \in \mathbb{Z}_p$, $\mathrm{r} = (s_1, s_2)^{\mathrm{T}}$, $\mathcal{B}$ computes $c_0 = [\mathbf{Xs}]_2$, $c_i = \left[\mathbf{W}_{\tau(i)}^{\mathrm{T}}\mathbf{Xs} + \sum_{j=1}^{n_2}(\mathbf{M})_{i,j}\mathbf{U}_J^{\mathrm{T}}\mathbf{Xs}\right]_1$, $ct' = \left[\mathbf{b}^{\mathrm{T}}\mathbf{Xs}\right]_T \cdot ck$, $CT = (CF, c_0, c_1, \ldots, c_{n_1}, ct')$.

The form of obtaining the private key in the game is shown in Figure 9. The form of the ciphertext is shown in Figure 10.

Lemma 2: For any PPT adversary $\mathcal{A}$, the advantages of distinguishing between Game_0 and Game_1 are: $\mathbf{Adv}_{0,1}^{\mathcal{A}}(\lambda) = 0$.

Proof: In Game_0 and Game_1, due to the fact that the initial output of S possesses a distribution congruent with the one postulated by DLIN. Therefore the process of generating public and private keys is the same. For a randomly selected $(\mathbf{W}_x)_{t,k}$ and $(\mathbf{W}_x)_{3,k}$, $\mathcal{B}$ performs an oracle query in the form of xkt in Game_1 being $\left[\left(\mathbf{W}_x^{\mathrm{T}}\mathbf{X}\right)_{k,t}\right]_1$, where the exponent of xkt

is $a_t(\mathbf{W}_x)_{t,k} + (\mathbf{W}_x)_{3,k}$. Consequently, $\left[\left(\mathbf{W}_x^{\mathrm{T}}\mathbf{X}\right)_{k,t}\right]_1$ is independent of x, k, t and follows a uniform distribution. Similarly, queries to $0jkt$ are independent and uniformly distributed in $\mathbb{G}$.

We set the response to the randomized prediction in Game_0 to the value generated in Game_1. Then the challenge ciphertext of Game_0 is:

$c_i = \left[\mathbf{W}_{\tau(i)}^{\mathrm{T}}\mathbf{X}\mathbf{s} + (\mathbf{M})_{i,j}\mathbf{U}_1^{\mathrm{T}}\mathbf{X}\mathbf{s} + \cdots + (\mathbf{M})_{i,n_2}\mathbf{U}_{n_2}^{\mathrm{T}}\mathbf{X}\mathbf{s}\right]_1$, where $\mathbf{s} = (\mathbf{s}_1, \mathbf{s}_2)^{\mathrm{T}}$, $k \in \{1, 2, 3\}$. Set $c_0 = [\mathbf{X}\mathbf{s}]_2$, $ct' = \left[\mathbf{b}^{\top}\mathbf{X}\mathbf{s}\right]_T \cdot ck$. Thus getting the same ciphertext as Game_1. $w_{i,j}$ expresses the element in row i and column j of $\mathbf{W}_{\mathbf{y}}$.

Then the key of Game_0 is: $sk_{y,t} = (a_t w_{t,1} + w_{3,1})\frac{d_1 r_1}{a_t} + (a_t w_{t,2} + w_{3,2})\frac{d_2 r_2}{a_t} + (a_t w_{t,3} + w_{3,3})\frac{r_1 + r_2}{a_t} + \frac{\sigma_y}{a_t} = (\mathbf{W}_y\mathbf{Y}\mathbf{r})_t + a_t^{-1}\left[(\mathbf{W}_y\mathbf{Y}\mathbf{r})_3 + \sigma_y\right]$, where $\mathrm{r} = (r_1, r_2)^{\mathrm{T}}$, The exponent of sk_y is $(\mathbf{W}_y\mathbf{Y}\mathbf{r})_3 - \left[(\mathbf{W}_y\mathbf{Y}\mathbf{r})_3 + \sigma_y\right]$. Then if σ_y is uniformly random, then $(\mathbf{W}_y\mathbf{Y}\mathbf{r})_3 + \sigma_y$ have the same distribution. Similarly, sk' and $\left[\mathbf{b} + \mathbf{U}_1\mathbf{Y}\mathbf{r} + \sigma'\mathbf{a}^{\perp}\right]_1$ have the same distribution. Denote $sk_0 = [\mathbf{Y}\mathbf{r}]_2$. Thus getting the same key as Game_1.

Lemma 3: For all PPT adversaries $\mathcal{A}$, where $q = 1, 2, \cdots, Q$, there exists a challenger $\mathcal{B}$ such that $\mathbf{Adv}^{\mathcal{A}}_{(2,q-1,3),(2,q,1)}(\lambda) \leq \mathbf{Adv}^{\mathcal{B}}_{\mathbf{DLIN}}(\lambda) + 1/p$

Proof: For the challenger, the difference between $\mathrm{Game}_{2,q-1,3}$ and $\mathrm{Game}_{2,q,1}$ is that i-th key. We generalize the advantage of the adversary $\mathcal{A}$ in distinguishing between two mixes to the DLIN assumption. The process is shown in Figure 11. If the DLIN assumption holds, the private key indistinguishability of the scheme is negligible for any PPT adversary, thus showing that our scheme is IND-CPA secure.

Similarly, for adversary $\mathcal{A}$ it is difficult to distinguish between the $\mathrm{Game}_{2,Q,3}$ and Game_3, and $\mathrm{Game}_{4,q,1}$ and $\mathrm{Game}_{4,q-1,3}$. See appendix for remaining certifications.

Lemma 4: For all PPT adversaries $\mathcal{A}$, where $q = 1, 2, \cdots, Q$, there there is $\mathbf{Adv}^{\mathcal{A}}_{(2,q,1),(2,1,2)}(\lambda) \leq 2/P$

Proof: Let the matrix $\mathbf{V} = \mathbf{x}^{\perp}{\mathbf{y}^{\perp}}^{\mathrm{T}}$, such that $\varsigma = {\mathbf{y}^{\perp}}^{\mathrm{T}}\mathbf{x}^{\perp}$, then $\mathbf{V}^{\mathrm{T}}\mathbf{X} = \mathbf{V}\mathbf{Y} = 0$, $\mathbf{V}\mathbf{x}^{\perp} = \left(\mathbf{x}^{\perp}\mathbf{2}{y^{\perp}}^{\mathrm{T}}\right)\mathbf{x}^{\perp} = \left({\mathbf{x}^{\perp}}^{\mathrm{T}}\mathbf{y}^{\perp}\right)\mathbf{x}^{\perp}$. In $\mathrm{Game}_{2,q,1}$, take $\sigma_x, \hat{r}, \sigma' \in \mathbb{Z}_p$ and implicitly set $\mathbf{W}_x$ and $\mathbf{U}_j$ to $\mathbf{W}_x^* = \mathbf{W}_x - \sigma_x(\varsigma\hat{r})^{-1}\mathbf{V}$, $\mathbf{U}_j^* = \mathbf{U}_j - \sigma'(\varsigma\hat{r})^{-1}\mathbf{V}$.

Since the random numbers do not affect the distribution of the matrix, the ciphertext is also unaffected. Since only the i -th key is changed and none of the rest of the key forms are changed, the i -th key is: $\mathbf{W}_y^*\left(\mathbf{Y}\mathbf{r} + \hat{r}\mathbf{x}^{\perp}\right) + \sigma_y\mathbf{x}^{\perp} = \mathbf{W}_y\left(\mathbf{Y}\mathbf{r} + \hat{r}\mathbf{x}^{\perp}\right)$. Similarly the i -th key in $\mathrm{Game}_{2,q,2}$ is distributed as $\mathbf{2}b + \mathbf{U}_1^*\left(\mathbf{Y}\mathbf{r} + \hat{r}\mathbf{x}^{\perp}\right) + \sigma'\mathbf{x}^{\perp} = \mathbf{b} + \mathbf{U}_1\left(\mathbf{Y}\mathbf{r} + \hat{r}\mathbf{x}^{\perp}\right)$.

Next, we show that for $\mathcal{A}$ it is difficult to distinguish between the two distributions $\mathrm{Game}_{2,Q,3}$ and Game_3.

Similarly, we show that it is difficult to distinguish between the two distributions $\mathrm{Game}_{4,q,1}$ and $\mathrm{Game}_{4,q-1,3}$ for any PPT adversary $\mathcal{A}$.

Lemma 5: For all PPT adversaries $\mathcal{A}$, where $q = 1, 2, \cdots, Q$, then there is $\mathbf{Adv}^{\mathcal{A}}_{(4,Q,3),5}(\lambda) \leq 2/p$.

Proof: For the challenger, the difference between $\mathrm{Game}_{4,Q,3}$ and Game_5 is that $\mathrm{Game}_{4,Q,3}$ is an encryption of ck and Game_5 is encrypting a random message. For $\in \mathbb{Z}_p$, we implicitly set $\mathbf{b}$ in $\mathrm{Game}_{4,Q,3}$ to $\mathbf{b} - \delta mathbf{f}x^{\perp}$. Obviously, it is embedded in the public key, and the key maintains an identical distribution. We replace the last ciphertext component $\left[\mathbf{b}^{\top}\left(\mathbf{X}\mathbf{s} + \hat{s}y^{\perp}\right)\right]_T \cdot ck$ by

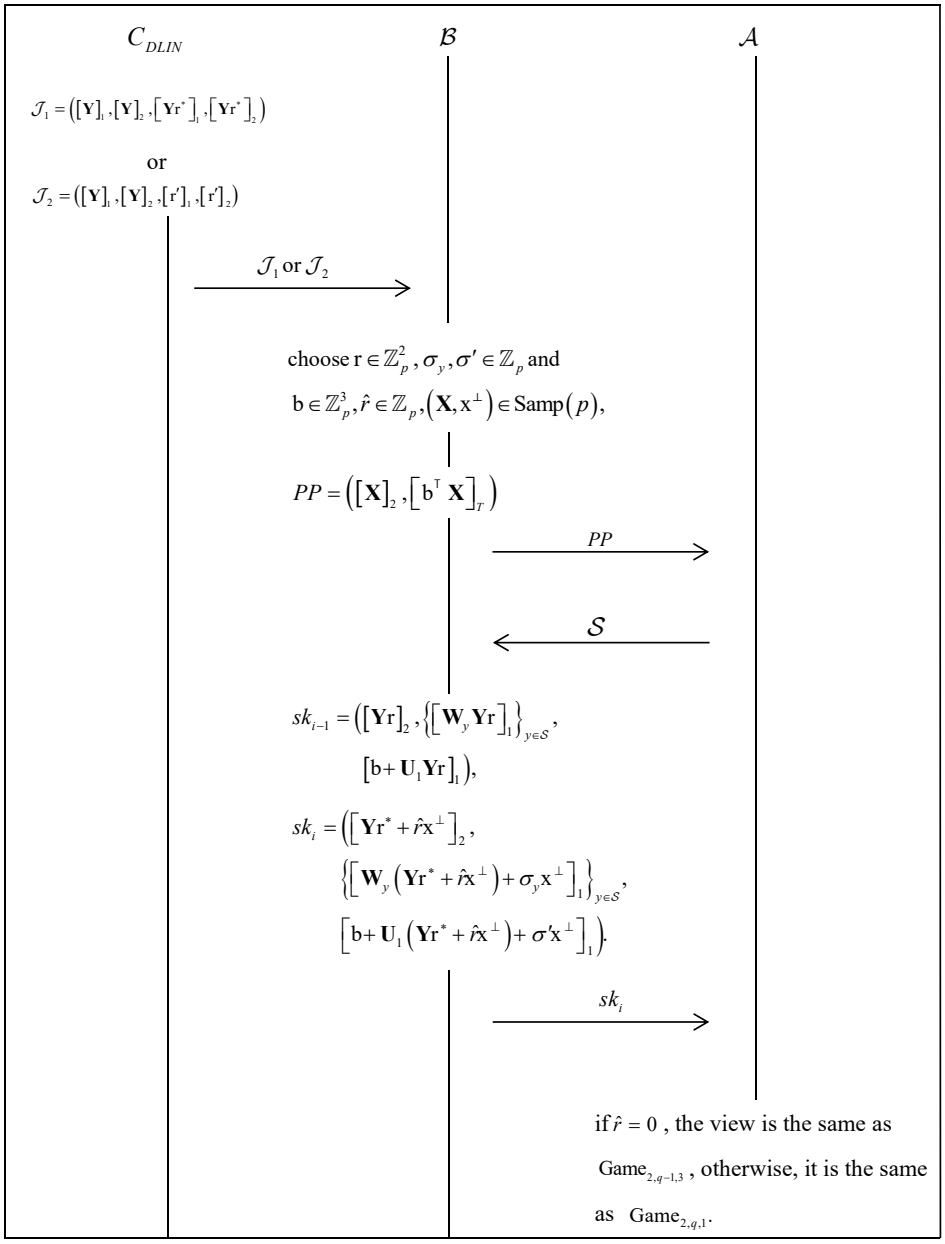

Figure 11: To distinguish the reductions in games.

$\left[\left(\mathbf{b}-\delta\mathbf{x}^{\perp}\right)^{\top}\left(\mathbf{X}\mathbf{s}+\hat{\mathbf{s}}\mathbf{y}^{\perp}\right)\right]_T \cdot ck$, we have

$$\begin{aligned}\left[\left(\mathrm{b}-\delta\mathrm{x}^{\perp}\right)^{\top}\left(\mathbf{X}\mathbf{s}+\hat{\mathbf{s}}\mathbf{y}^{\perp}\right)\right]_T \cdot ck &= \left[\mathbf{b}^{\top}\left(\mathbf{X}\mathbf{s}+\hat{\mathbf{s}}\mathbf{y}^{\perp}\right)+\delta\hat{\mathbf{s}}\langle\mathbf{x}^{\perp},\mathbf{y}^{\perp}\rangle\right]_T \cdot ck \\ &= \left[\mathbf{b}^{\top}\left(\mathbf{X}\mathbf{s}+\hat{\mathbf{s}}\mathbf{y}^{\perp}\right)\right]_T \cdot e(g,h)^{\delta\hat{\mathbf{s}}\langle\mathbf{x}^{\perp},\mathbf{y}^{\perp}\rangle} \cdot ck\end{aligned} \tag{5}$$

δ will not appear in any other part of the ciphertext, nor in any key or master public key. With probability $1-1/p$, $\left\langle\mathbf{x}^{\perp},\mathbf{y}^{\perp}\right\rangle \neq 0$. Thus, if $\hat{s}\neq 0$ occurs with probability $1-1/p$, $\delta\hat{s}\left\langle\mathbf{x}^{\perp},\mathbf{y}^{\perp}\right\rangle$ is uniformly distributed over $\mathbb{Z}_p$. Therefore, the ciphertext at this point is encrypt a random message.