Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.073553

ARTICLE

Evolve and Revoke: A Secure and Efficient Conditional Proxy Re-Encryption Scheme with Ciphertext Evolution

Han-Yu Lin, Tung-Tso Tsai* and Yi-Jia Ye

Department of Computer Science and Engineering, National Taiwan Ocean University, Keelung, 202, Taiwan

*Corresponding Author: Tung-Tso Tsai. Email: tttsai@mail.ntou.edu.tw

Received: 20 September 2025; Accepted: 28 November 2025; Published: 10 February 2026

ABSTRACT: Cloud data sharing is an important issue in modern times. To maintain the privacy and confidentiality of data stored in the cloud, encryption is an inevitable process before uploading the data. However, the centralized management and transmission latency of the cloud makes it difficult to support real-time processing and distributed access structures. As a result, fog computing and the Internet of Things (IoT) have emerged as crucial applications. Fog-assisted proxy re-encryption is a commonly adopted technique for sharing cloud ciphertexts. It allows a semi-trusted proxy to transform a data owner's ciphertext into another re-encrypted ciphertext intended for a data requester, without compromising any information about the original ciphertext. Yet, the user revocation and cloud ciphertext renewal problems still lack effective and secure mechanisms. Motivated by it, we propose a revocable conditional proxy re-encryption scheme offering ciphertext evolution (R-CPRE-CE). In particular, a periodically updated time key is used to revoke the user's access privileges while an access condition prevents a malicious proxy from re-encrypting unauthorized ciphertext. We also demonstrate that our scheme is provably secure under the notion of indistinguishability against adaptively chosen identity and chosen ciphertext attacks in the random oracle model. Performance analysis shows that our scheme reduces the computation time for a complete data access cycle from an initial query to the final decryption by approximately 47.05% compared to related schemes.

KEYWORDS: Revocable; proxy re-encryption; conditional access control; ciphertext evolution; fog computing

1 Introduction

The prevalence of cloud computing [1] and the emergence of various fog computing [2] applications have changed our ways in the utilization, sharing, and value-added creation of information. In diversified Internet of Things (IoT) applications such as intelligent transportation systems [3] and healthcare management systems, IoT devices play a crucial role in the circulation and sharing of gathered information. Specifically, in the healthcare applications, the widespread use of wearable devices and various wireless sensors enables healthcare professionals to remotely monitor patients' physiological data. The personal health data collected through IoT devices can serve as a valuable resource for medical research.

However, data owners might be concerned about data privacy and confidentiality when sharing their data in clouds. Another challenge is how to revoke user's access privileges, so as to achieve forward security. Proxy Re-Encryption (PRE) schemes [4–10] are commonly adopted techniques for sharing data in cloud environments. A PRE scheme allows an authorized proxy like a fog node to convert a ciphertext intended for some person into another one designated for the other person. The proxy is viewed as a semi-trusted party and should not be able to learn any information about shared ciphertexts.

In 2007, Canetti and Hohenberger [11] presented the first Chosen-Ciphertext Attack (CCA)-secure PRE mechanism with bidirectionality allowing ciphertext conversion between both parties. Green and Ateniese [12] came up with an identity-based proxy re-encryption (IB-PRE) in the same year. Nevertheless, Weng et al. [13] pointed out that although PRE schemes had been widely used for delegated decryption, traditional PRE mechanisms could not provide fine-grained access control. To cope with this problem, they addressed the concept of conditional proxy re-encryption (C-PRE) in which ciphertexts can only be re-encrypted if received re-encryption key satisfies the predefined condition specified by its data owner. In 2009, Chu et al. [14] presented the conditional proxy broadcast re-encryption (CPBRE) to support broadcast decryption. In 2011, Shao et al. [15] proposed an IB-CPRE system suitable for the fine-grained access control over encrypted emails. In 2018, Zeng and Choo [16] introduced a sender-specified PRE (SS-PRE) which could be viewed as a variant of C-PREs. Their SS-PRE scheme restricts that a proxy can only re-encrypt ciphertexts from a specific sender. This approach effectively addresses the over-delegation problem of PRE.

In 2021, Yao et al. [17] presented a revocable proxy re-encryption scheme providing ciphertext evolution. Their scheme allows a data owner to delegate decryption rights to authorized users and periodically evolve ciphertexts to reach the goal of revoking user. In 2023, Yao et al. [18] came up with an IB-PRE system with the property of single-hop conditional delegation and supports multi-hop ciphertext evolution. Yet, both schemes fail to consider the key-escrow issue in identity-based setting. Combined with ciphertext-policy attribute-based encryption (CP-ABE), Worapaluk and Fugkeaw [19] proposed a blockchain-assisted PRE in 2023. Their scheme supports both user and attribute revocation with auditability. Nevertheless, the blockchain only stores the revoked attribute list, and thus does not achieve genuine ciphertext evolution. Moreover, their scheme lacks both formal security proofs and integrated security analyses of potential information leakage in blockchain-assisted PRE.

In 2025, Singh et al. [20] proposed a lightweight and revocable certificateless public key encryption without bilinear pairings. Although their scheme also offers the characteristic ciphertext evolution, it essentially is not a PRE mechanism and would be not well-suited for flexible data sharing in clouds. Considering the merits of certificateless public systems, Eltayieb et al. [21] proposed a certificateless PRE (CL-PRE) for data sharing in clouds. Nevertheless, their scheme only satisfies the security requirement of chosen plaintext attacks (CPA). Zhou et al. [22] further addressed a certificateless conditional PRE (CL-CPRE) scheme for data sharing in IoT clouds. Although their scheme provides the functionality of ciphertext evolution and dynamic user key-update, it achieves only CPA security and fails to support the user revocation.

Motivation and Contributions

In recent years, fog-assisted PRE schemes [23–26] have gained popularity in data sharing. It enables a data owner to encrypt data and then allows a semi-honest fog node (proxy) to re-encrypt it for authorized recipients. However, existing schemes still present several limitations and research gaps:

- **Incomplete functionality in C-PRE schemes:** Prior C-PRE schemes [13–16] have provided fine-grained access control and hence solved the problem of over-delegation. However, they fail to consider the revocation problem and cannot support the functionality of ciphertext evolution, which is important to cloud ciphertext renewal.
- **Unresolved security issues in revocable schemes:** Although the scheme introduced by Yao et al. [17] further supports the characteristics of revocability, their identity-based framework still suffers from the key-escrow problem. In a related work, Lin and Chen [23] proposed a revocable IB-PRE scheme for fog computing environments. Nevertheless, we observe that the scheme still has the over-delegation and key-escrow problems and fails to support ciphertext evolution.

To overcome these limitations and address the gaps in existing research, we propose a fog-assisted revocable C-PRE scheme with ciphertext evolution (R-CPRE-CE). Our work is motivated by the need for an integrated and secure mechanism that simultaneously solves the problem of over-delegation, revocability, ciphertext evolution, and key-escrow. The primary contributions of this paper are as follows:

- **Conditional re-encryption with user revocation:** The proposed R-CPRE-CE scheme not only incorporates a predefined access condition to prevent over-delegation, but also uses a periodically updated time-key to efficiently revoke a user's access privileges.
- **Ciphertext evolution for forward security:** With ciphertext evolution, our R-CPRE-CE scheme allows cloud ciphertexts to be shared across time periods and fulfills forward security.
- **Elimination of the key-escrow problem:** By employing an anonymous key generation technique, our R-CPRE-CE scheme effectively solves the inherent key-escrow problem in identity-based systems.

Although our R-CPRE-CE scheme builds upon similar concepts from previous PRE systems, it offers many advantages over the schemes of Yao et al. [17,18] and the Lin and Chen [23]. Specifically, the identity-based frameworks of Yao et al. [17,18] inherently suffer from the key-escrow problem. In contrast, the proposed R-CPRE-CE scheme eliminates this risk by employing an anonymous key generation technique. Furthermore, compared to the Lin-Chen system [23], our work provides better functionality by incorporating both conditional PRE to prevent over-delegation and a ciphertext evolution mechanism to ensure forward security. Computationally, the proposed system also demonstrates better performance, particularly in terms of the one-time access metric compared to Yao et al. [17], as detailed in Section 5.2.

The remaining parts of this paper are organized below. Section 2 briefly reviews some preliminaries in relation to the proposed scheme. We give the formal security model and definition for the proposed system in Section 3. A concrete R-CPRE-CE scheme is presented in Section 4. We analyze the security and performance of our scheme in Section 5. Finally, a conclusion is made in Section 7.

2 Preliminaries

This section introduces the essential cryptographic backgrounds and computational assumptions.

2.1 Bilinear Pairing

Let G_1 and G_2 be two cyclic groups of prime order p. Two generators g and g_1 belong to G_1. A symmetric bilinear pairing $e: G_1 \times G_1 \rightarrow G_2$ satisfies the following three properties:

- ***Bilinearity:*** If $g \in G_1$ and $a, b \in Z_p$, the equality $e\left(g^a, g^b\right) = e(g, g)^{ab}$ holds.
- ***Non-degeneracy:*** For generators $g, g_1 \in G_1$, we have that $e(g, g_1) \neq 1$.
- ***Computability:*** Let $g, g_1 \in G_1$ and there exists a polynomial-time algorithm could calculate $e(g, g_1)$ efficiently.

2.2 The Problem of Decisional Bilinear Diffie-Hellman (DBDH)

Given $(g, g^f, g^s, g^k, \gamma)$, where $g, g^f, g^s, g^k \in G_1$, $\gamma \in G_2$, and $f, s, k \in Z_p$, the problem of DBDH is to determine if $\gamma = e(g, g)^{fsk}$ in which $e: G_1 \times G_1 \rightarrow G_2$ is a symmetric bilinear pairing.

2.3 The Assumption of Decisional Bilinear Diffie-Hellman (DBDH)

The assumption of DBDH holds if any probabilistic polynomial-time adversary $\mathcal{A}$ has a negligible advantage in solving the DBDH problem. The advantage of adversary $\mathcal{A}$ is defined as:

$$Adv\left(\mathcal{A}\right) = \left|\Pr\left[\mathcal{A}\left(g, g^f, g^s, g^k, e(g, g)^{fsk}\right) = 1\right] - \Pr\left[\mathcal{A}\left(g, g^f, g^s, g^k, \gamma\right) = 1\right]\right| \leq \varepsilon$$

where ε is negligible.

3 The Proposed Scheme

This section outlines the system model, core algorithms, and security model of the proposed scheme.

3.1 System Model

The proposed R-CPRE-CE system includes three layers of participated parties and an additional Private Key Generator (PKG) for key generation. As shown in Fig. 1, the User Layer transmits the demand for ciphertext via the Proxy Layer to the Cloud Layer. Furthermore, the proxy also obtains re-encryption keys from the data owner, transforms cloud evolved ciphertexts, and forwards the result to users. In the fog computing environment, the proxy is usually composed of a group of distributed fog nodes.

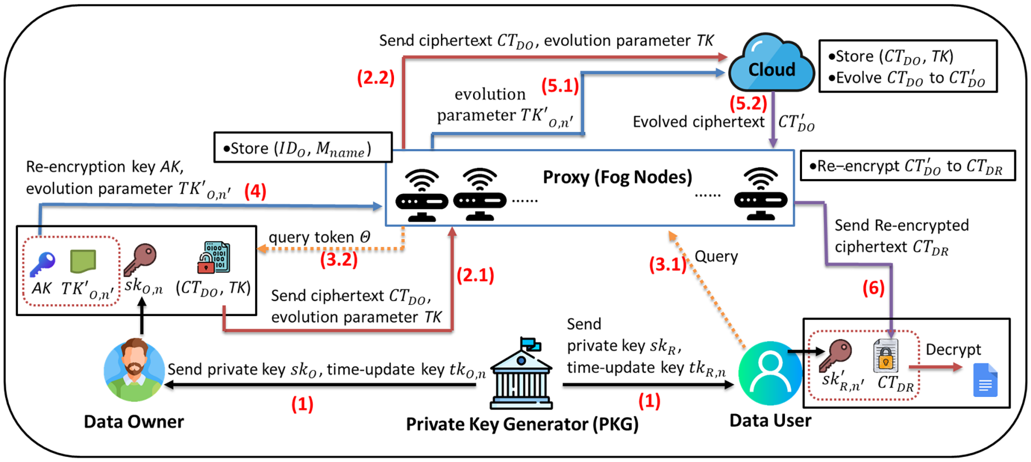
Send ciphertext CT_{DO}, evolution parameter TK
(2.2)
Cloud
•Store (CT_{DO}, TK)
•Evolve CT_{DO} to CT'_{DO}
evolution parameter $TK'_{O,n'}$
(5.1)
(5.2)
Evolved ciphertext CT'_{DO}
•Store (ID_O, M_{name})
Re-encryption key AK, evolution parameter $TK'_{O,n'}$
(4)
Proxy (Fog Nodes)
•Re-encrypt CT'_{DO} to CT_{DR}
query token Θ
(3.2)
AK
$TK'_{O,n'}$
$sk_{O,n}$
(CT_{DO}, TK)
(2.1)
Send ciphertext CT_{DO}, evolution parameter TK
(3.1)
Query
Send Re-encrypted ciphertext CT_{DR}
(6)
Send private key sk_O, time-update key $tk_{O,n}$
(1)
Send private key sk_R, time-update key $tk_{R,n}$
(1)
$sk'_{R,n'}$
CT_{DR}
Decrypt
Data Owner
Private Key Generator (PKG)
Data User

Figure 1: The proposed R-CPRE-CE system model

3.2 Algorithm Definition

The algorithms utilized in the proposed scheme are defined as follows:

1. **Setup**$(1^l) \rightarrow (\boldsymbol{PP}, \boldsymbol{Msk})$: Given a security parameter l, it generates the system's public parameters PP and a master secret key Msk.
2. **InitialKeyGen**$(\boldsymbol{PP}, \boldsymbol{Msk}, \boldsymbol{ID_u}) \rightarrow \boldsymbol{sk_u}$: Given PP, Msk, and a user identity ID_u, this algorithm generates the user's initial private key sk_u.
3. **TimeUpdateKeyGen**$(\boldsymbol{PP}, \boldsymbol{Msk}, \boldsymbol{ID_u}, \boldsymbol{n}) \rightarrow \boldsymbol{tk_{u,n}}$: Given PP, Msk, ID_u, and a time period n, this algorithm generates a corresponding time-update key $tk_{u,n}$.
4. **Encryption**$(\boldsymbol{PP}, \boldsymbol{Y}, \boldsymbol{ID_O}, \boldsymbol{sk_O}, \boldsymbol{n}, \boldsymbol{c}, \boldsymbol{M}) \rightarrow (\boldsymbol{CT_{DO}}, \boldsymbol{TK})$: Given PP, a symmetric key Y, a data owner's identity ID_O, his initial private key sk_O, a time period n, an access condition c, and a message M, this algorithm outputs the original ciphertext CT_{DO} and an evolution parameter TK. The condition c represents a meaningful bit-string such as a keyword, a timestamp or an attribute.
5. **Query**$(\boldsymbol{PP}, \boldsymbol{ID_R}, \boldsymbol{M_{name}}) \rightarrow \boldsymbol{\Theta}$: Given PP, a data user's identity ID_R, and the index of requested file M_{name}, this algorithm generates the corresponding query token Θ.
6. **ReKeyGen**$\left(\boldsymbol{PP}, \boldsymbol{ID_R}, \boldsymbol{n'}, \boldsymbol{sk_O}, \boldsymbol{sk'_{O,n'}}, \boldsymbol{c}, \boldsymbol{\Theta}\right) \rightarrow (\boldsymbol{AK}, \boldsymbol{TK'_{O,n'}}$ **):** Given PP, an identity of data user ID_R, the current time period n', the data owner's initial private key sk_O and the one $sk'_{O,n'}$ for the time period n', an access condition c, and the query token Θ, it generates the re-encryption key AK and a corresponding evolution parameter $TK'_{O,n'}$.

7. **Evolve**$(\boldsymbol{PP}, \boldsymbol{CT_{DO}}, \boldsymbol{TK}, \boldsymbol{TK'_{O,n'}}) \rightarrow \boldsymbol{CT^{'}_{DO}}$: Given PP, an original ciphertext CT_{DO}, and two evolution parameters (TK, $TK'_{O,n'}$), it calculates the evolved ciphertext (CT'_{DO}).
8. **Re-Encryption**$(\boldsymbol{PP}, \boldsymbol{ID_R}, \boldsymbol{AK}, \boldsymbol{CT^{'}_{DO}}) \rightarrow \boldsymbol{CT_{DR}}$: Given PP, a data user's identity ID_R, a re-encryption key AK, and the evolved ciphertext CT'_{DO}, it computes the re-encrypted ciphertext CT_{DR}.
9. **Decryption**$(\boldsymbol{PP}, \boldsymbol{sk^{'}_{u,n'}}, \boldsymbol{CT^{'}_{DO}}/\boldsymbol{CT_{DR}}) \rightarrow \boldsymbol{M}$: It can be divided into two cases, i.e., one for the data owner and the other for the data user. In general, given PP, a private key $sk'_{u,n'}$ for the current time period n', and the evolved ciphertext CT'_{DO} or the re-encrypted ciphertext CT_{DR}, the algorithm derives the symmetric key Y to recover the message M.

Correctness: The correctness property guarantees that (1) a legitimate user holding the valid private key for the current time period n' can successfully decrypt the ciphertext, and (2) a revoked user whose identity has been invalidated is unable to correctly decrypt the ciphertext. Concretely speaking, given the public parameters (PP), a symmetric key (Y), current time period (n'), a user identity (ID_u) with his/her initial private key (sk_u) and the one ($sk'_{u,n'}$) for the time period n', an access condition (c), a re-encryption key (AK), evolution parameters (TK, $TK'_{O,n'}$), a message (M), and $R_{n'}$, the set of revoked identities at time period n', the proposed system should satisfy the following conditions:

- **Correctness of decryption:**

  For all users whose identities have not been revoked at time period n', i.e., $\forall\ ID_u \notin\ R_{n'}$, the system ensures the correctness of decryption as follows:

  - *Decryption by the Data Owner*

$$(CT_{DO}, TK) \leftarrow Encryption\,(PP, Y, ID_O, sk_O, n, c, M)$$
$$CT'_{DO} \leftarrow Evolve\left(PP, CT_{DO}, TK, TK'_{O,n'}\right)$$
$$\Pr\left[Decryption_1\left(CT'_{DO}, sk'_{O,n'}\right) = M\right] = 1$$

  - *Decryption by the Data User*

$$CT_{DR} \leftarrow Re - Encryption\,(PP, ID_R, AK, CT'_{DO})$$
$$\Pr\left[Decryption_2\left(CT_{DR}, sk'_{R,n'}\right) = M\right] = 1$$

- **Correctness of revocation:**

  For all users whose identities have been revoked at time period n', i.e., $\forall\ ID_u \in R_{n'}$, the system ensures the correctness of revocation as follows:

$$\Pr\left[Decryption_2\left(CT_{DR}, sk_{R,n}\right) = M\right] \approx 0$$

3.3 Security Model

This section defines the security game of indistinguishability against adaptively chosen identity and chosen ciphertext attacks (IND-PrID-CCA), where PrID stands for Proxy Identity, between a probabilistic polynomial-time (PPT) adversary $\mathcal{A}$ and a challenge $\mathcal{B}$. We will show that the proposed system is formally proven secure in the random oracle model under the DBDH assumption.

Definition 1: *(IND-PrID-CCA) If no PPT adversary $\mathcal{A}$ can win the following game against a challenger $\mathcal{B}$ with non-negligible advantage, the proposed scheme achieves IND-PrID-CCA security*

$\boldsymbol{Game_{IND-PrID-CCA}}$:

Setup. The challenger $\mathcal{B}$ takes a security parameter l as input, runs Setup(1^l) to generate public parameters PP and Msk. Then he sends PP to $\mathcal{A}$.

Phase 1. The adversary $\mathcal{A}$ adaptively makes polynomially many queries as follows:

- *Initial Private Key Query:* $\mathcal{A}$ selects an identity ID_u and sends it to $\mathcal{B}$ who runs InitialKeyGen(PP, Msk, ID_u) to generate the initial private key sk_u which is then returned to $\mathcal{A}$.
- *Time-Update Key Query:* $\mathcal{A}$ sends an identity ID_u and a time period n to the challenger $\mathcal{B}$ who runs TimeUpdateKeyGen(PP, Msk, ID_u, n) to generate the time-update key $tk_{u,n}$ which is then returned to $\mathcal{A}$.
- *Re-Encryption Key Query:* $\mathcal{A}$ selects two valid user identities (ID_O, ID_R), a time period n', an access condition c, and an index of requested file M_{name} to $\mathcal{B}$. The challenger $\mathcal{B}$ first runs InitialKeyGen(PP, Msk, ID_O) and TimeUpdateKeyGen(PP, Msk, ID_O, n') to obtain the private key $sk'_{O,n'}$ for the time period n'. Then $\mathcal{B}$ runs Query(PP, ID_R, M_{name}) to obtain a query token Θ. Finally, the re-encryption key AK could be returned to $\mathcal{A}$ by performing ReKeyGen$(PP, ID_R, n', sk_O, sk'_{O,n'}, c, \Theta)$.
- *Decryption Query:* $\mathcal{A}$ provides a valid user identity ID_u and a ciphertext CT_{DR} associated with a time period n' for $\mathcal{B}$. The challenger $\mathcal{B}$ first runs InitialKeyGen(PP, Msk, ID_u) and TimeUpdateKeyGen(PP, Msk, ID_u, n') to obtain the private key $sk'_{u,n'}$ for the time period n'. Then $\mathcal{B}$ runs Decryption$\left(PP, sk'_{u,n'}, CT_{DR}\right)$ to recover M and returns it to $\mathcal{A}$.

Challenge. $\mathcal{A}$ defines two symmetric keys (Y_0, Y_1) of equal length, a target user identity ID^*, a time period n^*, an access condition c^*, and the message M^*. Then $\mathcal{B}$ utilizes $(PP, Y_\lambda, ID^*, n^*, c^*, M^*)$ where $\lambda \in_R \{0, 1\}$ to generate a ciphertext $CT^* = ({C_1}^*, {C_2}^*, {C_3}^*, {C_4}^*, {CT_1}^*)$ and return it to $\mathcal{A}$.

Phase 2. Upon getting CT^*, $\mathcal{A}$ executes the queries of Phase 1, but the restrictions are as follows:

- Initial private key queries for the target identity ID^* are disallowed.
- Whenever $\mathcal{A}$ acts as a revoked user (holds its initial private key), it cannot query a time-update key of the target period n^*.
- Re-encryption key queries involving ID^* as an input are disallowed.
- Decryption queries involving (ID^*, CT^*) as an input are disallowed.
- The number of queries is bounded by q_{pk} for initial private key queries, q_{tk} for time-update key queries, q_{de} for decryption queries, and q_{pr} for re-encryption key queries.

Guess. When Phase 2 is finished, the adversary $\mathcal{A}$ outputs a bit λ'. If $\lambda' = \lambda$, $\mathcal{A}$ wins the game. The advantage of $\mathcal{A}$ is defined as:

$$Adv\,(\mathcal{A}) = |\Pr[\lambda' = \lambda] - \frac{1}{2}|$$

In the security model, we defined several query restrictions for the adversary capabilities to correctly reflect a realistic attack scenario while maintaining tractability of the security proof. For instance, in phase 2, an initial private key query for the target identity is disallowed. This is a standard assumption in IND-PrID-CCA security games, since the adversary holding the target private key can easily decrypt the ciphertext and trivially break the security. This restriction ensures that the adversary's success advantage does not rely on possessing the target secret key. Furthermore, a decryption query involving the challenge ciphertext is disallowed. Such a restriction is crucial for modeling chosen-ciphertext attacks (CCA) where an adversary cannot directly submit the challenge ciphertext to decryption oracles. It simulates a circumstance where attackers cannot access the victim's decryption device, but must rely on other queries to gain information.

Although these restrictions are standard in formal security models, it is important to consider their alignment with real world attack scenarios, particularly in a fog computing environment with multiple semi-trusted proxies. In practice, an adversary might compromise one or more fog nodes and attempt to use them as re-encryption or decryption oracles. Our security model implicitly addresses this by providing the adversary with more query capabilities. For instance, the adversary can request re-encryption keys for any non-challenge identity and decrypt any ciphertext except for the specific challenge ciphertext itself. This simulates a powerful attacker who has significant control over the network but has not yet compromised the specific target's final decryption device or process. The core security of the challenge ciphertext relies on the cryptographic hardness of the underlying problem, even if the adversary can observe and manipulate other related ciphertexts through compromised proxies. To the best of our knowledge, no model can perfectly capture all real-world nuances. The IND-PrID-CCA model provides a well-accepted abstraction for demonstrating security against adaptive attacks in such distributed environments.

4 Construction of Proposed Scheme

Building upon the previously defined algorithms, we now present a concrete construction.

- **Setup**$(\mathbf{1}^{l}) \to (\boldsymbol{PP}, \boldsymbol{Msk})$

  Given a security parameter l, the Private Key Generator (PKG) outputs a master secret key Msk and the public parameters $PP = \{G_1, G_2, g, p, e, Mpk, H_1, H_2, H_3, H_4, E(\cdot), D(\cdot)\}$ defined as follows:

  (1) G_1 and G_2 are two prime-order multiplicative groups of the order p, g is a generator of G_1, and e is a bilinear mapping function, $e\colon G_1 \times G_1 \to G_2$.

  (2) A random value $\alpha \in Z_p$ is selected and set as the master secret key, i.e., $Msk = \alpha$. The master public key Mpk is then computed as $Mpk = g_1 = g^{\alpha}$.

  (3) H_1, H_2, H_3, and H_4 are four one-way hash functions satisfying that $H_1\colon \{0,1\}^* \to G_1$, $H_2\colon \{0,1\}^* \to G_1$, $H_3\colon \{0,1\}^* \to G_1$, and $H_4\colon G_2 \to G_1$.

  (4) $E(\cdot)$ and $D(\cdot)$ denote symmetric encryption and decryption functions, respectively.

- **InitialKeyGen**$(\boldsymbol{PP}, \boldsymbol{Msk}, \boldsymbol{ID_u}) \to \boldsymbol{sk_u}$

  The initial private key is generated through the interactive steps:

  (1) The registering user ID_u selects random values $r_u, Zu \in Z_p$ and computes:

$$Zu = g^{Zu} \tag{1}$$

$$R_u = Zu \cdot H_2(ID_u \parallel r_u) \tag{2}$$

  (2) ID_u sends (ID_u, R_u) to the PKG which then computes:

$$sk_u' = {R_u}^{\alpha} = (Zu \cdot H_2(ID_u \parallel r_u))^{\alpha} \tag{3}$$

  (3) PKG returns sk_u' to the user ID_u who will compute:

$$sk_u^1 = sk_u'/{g_1}^{zu} = H_2(ID_u \parallel r_u)^{\alpha} \tag{4}$$

$$sk_u^2 = r_u \tag{5}$$

  (4) The user ID_u sets his/her initial private key $sk_u = (sk_u^1, sk_u^2)$.

  (5) The sk_u could be further verified for correctness using the following equation:

$$e\left(sk_u^1, g\right) = e(H_2(ID_u \parallel r_u), g_1) \tag{6}$$

- **TimeUpdateKeyGen**$(\boldsymbol{PP}, \boldsymbol{Msk}, \boldsymbol{ID_u}, \boldsymbol{n}) \to \boldsymbol{tk_{u,n}}$

  The time-update key is generated through the interactive steps:

  (1) ID_u sends (ID_u, n) to the PKG which then computes:

$$tk_{u,n} = H_1(ID_u \parallel n)^{\alpha} \tag{7}$$

  (2) PKG returns the time-update key $tk_{u,n}$ to ID_u.

(3) The $tk_{u,n}$ could be further verified for correctness using the following equation:

$$e\left(tk_{u,n}, g\right) = e\left(H_1\left(ID_u \parallel n\right), g_1\right) \tag{8}$$

(4) ID_u finally derives the private key $sk_{u,n}$ for the time period n as

$$sk_{u,n} = sk_u^1 \cdot tk_{u,n} = \left(H_1\left(ID_u \parallel n\right) \cdot H_2\left(ID_u \parallel r_u\right)\right)^{\alpha} \tag{9}$$

- **Encryption$(\boldsymbol{PP}, \boldsymbol{Y}, \boldsymbol{ID_O}, \boldsymbol{sk_O}, \boldsymbol{n}, \boldsymbol{c}, \boldsymbol{M}) \rightarrow (\boldsymbol{CT_{DO}}, \boldsymbol{TK})$**

  Let $M = (m_1, m_2, \ldots, m_s)$ be a message associated with the file index M_{name}, Y a chosen symmetric key, and $c \in \{0,1\}^*$ a predefined access condition. The Data Owner ID_O selects a random value $k \in Z_p$ to compute the original ciphertext CT_{DO} for M as follows:

  (1) Compute

$$C_1 = Y \cdot e\Big(\left(H_1\left(ID_O \parallel n\right) \cdot H_2\left(ID_O \parallel r_O\right)\right)^{k \cdot sk_O^2}, g_1\Big) \tag{10}$$

$$C_2 = g^{k \cdot sk_O^2} \tag{11}$$

$$C_{3,n} = \left(H_1\left(ID_O \parallel n\right)\right)^{sk_O^2} \tag{12}$$

$$C_4 = H_3\left(ID_O \parallel c\right)^{k \cdot sk_O^2} \tag{13}$$

  (2) Compute an evolution parameter as

$$TK = {g_1}^{k} \tag{14}$$

  (3) Set $CT_0 = (C_1, C_2, C_3, C_4)$ and compute

$$CT_1 = \left(E\left(Y, m_1\right), E\left(Y, m_2\right), \ldots, E\left(Y, m_s\right)\right) \tag{15}$$

  (4) Output $CT_{DO} = (CT_0, CT_1)$

  The data owner ID_O will store M_{name} with the corresponding access condition c, and sends $(ID_O, n, CT_{DO}, TK, M_{name})$ to the proxy. Upon receiving it, the proxy keeps (ID_O, M_{name}) and forwards $(ID_O, n, CT_{DO}, TK, M_{name})$ to the cloud for storage.

- **Query $(\boldsymbol{PP}, \boldsymbol{ID_R}, \boldsymbol{M_{name}}) \rightarrow \boldsymbol{\Theta}$**

  When a data user ID_R requests the ciphertext stored in the cloud with respect to the file index M_{name}, he selects a random value $\rho \in Z_p$ to compute

$$Q = g^{\rho} \tag{16}$$

  Then (ID_R, Q, M_{name}) is sent to the nearby proxy which generates a query token $\Theta = (ID_R, Q)$ delivered to the data owner ID_O.

- **ReKeyGen $\left(\boldsymbol{PP}, \boldsymbol{ID_R}, \boldsymbol{sk_O}, \boldsymbol{sk'_{O,n'}}, \boldsymbol{n'}, \boldsymbol{c}, \boldsymbol{\Theta}\right) \rightarrow (\boldsymbol{AK}, \boldsymbol{TK'_{O,n'}})$**

  After receiving the token Θ, ID_O derives the re-encryption key AK by the following steps:

  (1) Select random values $\omega, x, \pi \in Z_p$.

  (2) Compute the hash value of the access condition c as

$$W = H_3\left(ID_O \parallel c\right) \tag{17}$$

(3) Compute the re-encryption key $AK = (A_1, A_2, A_3, A_4)$ as

$$A_1 = W^x \cdot {g_1}^{\omega} \tag{18}$$

$$A_2 = \frac{\left(sk'_{O,n'}\right) \cdot {g_1}^{\omega}}{H_4\left(e\left(H_1\left(ID_R \parallel n'\right) \cdot Q^{\pi}, g_1\right)\right)} \tag{19}$$

$$A_3 = e(g^{\pi}, g_1) \tag{20}$$

$$A_4 = g^x \tag{21}$$

(4) Compute the evolution parameter $TK'_{O,n'}$ for the current time period n' as

$$TK'_{O,n'} = (H_1(ID_O \parallel n'))^{sk_O^2} \tag{22}$$

Subsequently, the data owner sends $\left(AK, TK'_{O,n'}\right)$ to the proxy which also transmits $\left(ID_O, TK'_{O,n'}, M_{name}\right)$ to the cloud for ciphertext evolution.

- **Evolve** $\left(\boldsymbol{PP}, \boldsymbol{CT_{DO}}, \boldsymbol{TK}, \boldsymbol{TK'_{O,n'}}\right) \rightarrow \boldsymbol{CT^{'}_{DO}}$

  Upon receiving $\left(ID_O, TK'_{O,n'}, M_{name}\right)$, the cloud server (CS) searches for stored $\{CT_{DO} = (CT_0, CT_1), TK_{O,n}\}$ using M_{name} and then

  (1) Compute

  $$C'_1 = C_1 \cdot e\left(TK'_{O,n'}/C_{3,n}, TK\right) \tag{23}$$

  where $CT_0 = (C_1, C_2, C_3, C_4)$.

  (2) Set the evolved ciphertext as $CT'_{DO} = \{CT'_0 = (C'_1, C_2, C_3, C_4), CT_1\}$ and deliver it to the proxy.

- **Re-Encryption** $\left(\boldsymbol{PP}, \boldsymbol{ID_R}, \boldsymbol{AK}, \boldsymbol{CT^{'}_{DO}}\right) \rightarrow \boldsymbol{CT_{DR}}$

  When the proxy gains the re-encryption key $AK = (A_1, A_2, A_3, A_4)$ and $TK'_{O,n'}$ from ID_O and the evolved ciphertext CT'_{DO} from the CS, it performs the re-encryption process below:

  (1) Compute C'_1 as

  $$C''_1 = C'_1 \cdot e(A_1, C_2)/e(C_4, A_4) \tag{24}$$

  At this stage, the ciphertext and re-encryption key must associate with the same access condition c for the proxy to generate a valid re-encrypted ciphertext CT_{DR}. In other words, this mechanism restricts the re-encryption privilege of the semi-trusted proxy, preventing it from illegally re-encrypting other unauthorized ciphertexts stored by the data owner in the CS.

  (2) Compute C''_2, C''_3, and C''_4 as

  $$C''_2 = C_2 = g^{k \cdot sk_O^2} \tag{25}$$

  $$C''_3 = A_2 = \left(sk'_{O,n'}\right) \cdot {g_1}^{\omega}/H_4(e(H_1\left(ID_R \parallel n'\right) \cdot Q^{\pi}, g_1)) \tag{26}$$

  $$C''_4 = A_3 = e(g^{\pi}, g_1) \tag{27}$$

  (3) Output the ciphertext $CT_{DR} = \{CT''_0 = (C''_1, C''_2, C''_3, C''_4), CT_1\}$ and send it to ID_R. Fig. 2 illustrates the ciphertext evolution and re-encryption process across time periods.

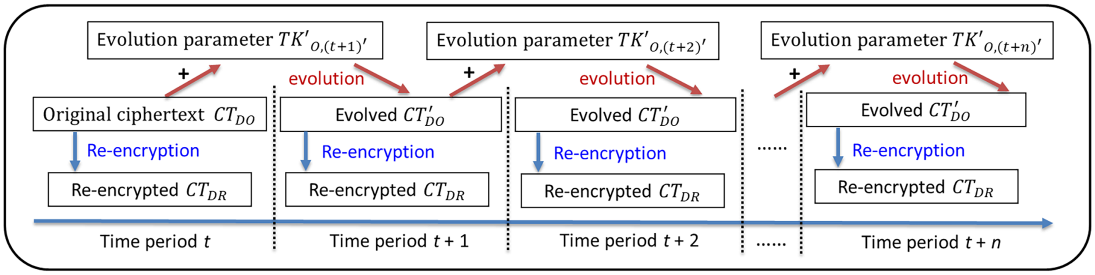

Figure 2: Illustration of ciphertext evolution and re-encryption process across time periods

- **Decryption** $\boldsymbol{(PP, sk'_{u,n'}, CT'_{DO}/CT_{DR}) \to M}$

  The decryption process is classified into two cases. The first is performed by a data owner ID_O with the following steps:

  (1) Computes the decryption key

$$Y = \frac{C_1''}{e\left(C_2, sk'_{O,n'}\right)} \tag{28}$$

The correctness of Eq. (28) could be easily confirmed, since

$$\frac{C_1''}{e(C_2, sk_{O,n''})} = \frac{Y \cdot e\left((H_1(ID_O \parallel n') \cdot H_2(ID_O \parallel r_O))^{k\cdot sk_O^2}, g_1\right)}{e\left(g^{k\cdot sk_O^2}, (H_1(ID_O \parallel n') \cdot H_2(ID_O \parallel r_O))^{\alpha}\right)} = Y.$$

  (2) Recover the message M by computing

$$\{M_i = D(Y, E(Y, m_i))\}_{i=1,\ldots,s} \tag{29}$$

  The second is performed by the data user ID_R with the following steps:

  (1) Compute the parameter E as

$$E = C_3'' \cdot H_4\left(\frac{C_4''^{\rho} \cdot e\left(sk'_{R,n'}, g\right)}{e(H_2(ID_R \parallel r_R), g_1)}\right) = \left(sk'_{O,n'}\right) \cdot g_1^{\omega} \tag{30}$$

  (2) Compute the decryption key

$$Y = \frac{C_1''}{e(E, C_2'')} \tag{31}$$

The correctness of Eq. (31) could be easily confirmed, since

$$\begin{aligned}\frac{C_1''}{e(E, C_2'')} &= \frac{Y \cdot e\left((H_1(ID_O \parallel n') \cdot H_2(ID_O \parallel r_O))^{k\cdot sk_O^2}, g_1\right) \cdot e\left(g_1^{\omega}, g^{k\cdot sk_O^2}\right)}{e\left((H_1(ID_O \parallel n') \cdot H_2(ID_O \parallel r_O))^{\alpha} \cdot g_1^{\omega}, g^{k\cdot sk_O^2}\right)} \\ &= \frac{Y \cdot e\left((H_1(ID_O \parallel n') \cdot H_2(ID_O \parallel r_O))^{k\cdot sk_O^2}, g_1\right) \cdot e\left(g_1^{\omega}, g^{k\cdot sk_O^2}\right)}{e\left((H_1(ID_O \parallel n') \cdot H_2(ID_O \parallel r_O))^{\alpha}, g^{k\cdot sk_O^2}\right) \cdot e\left(g_1^{\omega}, g^{k\cdot sk_O^2}\right)} = Y\end{aligned}$$

  (3) Recover the message M with Eq. (29).

5 Security Proof and Comparison

Based on previously defined security model, this section provides the formal security proof for the proposed R-CPRE-CE scheme. Meanwhile, some comparisons with similar systems are also made.

5.1 Security Proof

We show that the proposed R-CPRE-CE system is provably secure in the security notion of IND-PrID-CCA as Theorem 1. We prove the security of our scheme using a reductionist argument in the random oracle model (ROM). The core idea is to demonstrate that if a PPT adversary can break the IND-PrID-CCA security of our scheme, then we can construct an algorithm to solve the DBDH problem, which is assumed to be computationally intractable. In the ROM, an adversary must query a random oracle for a result rather than computing it independently. This model is a widely used idealization in cryptography and can simplify the proof of complex cryptographic protocols. When instantiating the random oracles with real world hash functions like Secure Hash Algorithm 3 (SHA-3), we follow common cryptographic practices to mitigate potential attacks. Specifically, the inputs to the hash functions are carefully structured, e.g., concatenating identities with nonces or other distinct values, to prevent collision attacks and other structural vulnerabilities that could arise from a naive implementation. Thus, we believe that such an instantiation will not introduce new weaknesses into our scheme. However, a true random oracle does not exist in the real world. A ROM-proven-secure scheme could also be insecure in the standard model. Extending our current mechanism to fulfill standard model security requires significant modifications and inevitably increases the computation complexity. Consequently, we leave the adaptation to achieve standard model security as our future work.

Theorem 1: *(Proof of IND-PrID-CCA)*

Let H_1, H_2, and H_3 be random oracles. Under the assumption of DBDH, the constructed R-CPRE-CE system fulfills the IND-PrID-CCA security. Concretely speaking, if there exists any PPT adversary $\mathcal{A}$ breaking the IND-PrID-CCA security of our proposed scheme with a non-negligible advantage ε, under the query limits q_{tk}, q_{de}, and q_{pr}, an algorithm $\mathcal{B}$ that is able to solve the DBDH problem with the non-negligible advantage ε' where

$$\varepsilon' \geq \frac{\varepsilon}{e\left(q_{tk} + q_{pr} + q_{de} + 1\right)}$$

and e denotes the base of the natural logarithm.

Proof: Let $\left(g, g^f, g^s, g^k, \gamma\right)$ be a DBDH instance provided to $\mathcal{B}$, in which $f, s, k \in Z_p$ and $\gamma \in G_2$. The proof shows how a challenger $\mathcal{B}$ takes the advantage of adversary $\mathcal{A}$ to decide whether γ equals to $e(g, g)^{fsk}$ or not.

Setup. The challenger $\mathcal{B}$ first initializes Setup$\left(1^l\right)$ to generate $PP = \{e, G_1, G_2, g, p, Mpk, H_4, E(\cdot), D(\cdot)\}$ and sends PP to the adversary $\mathcal{A}$. The master public key Mpk is defined as $P = g^s$, and the challenger $\mathcal{B}$ does not know $Msk = s$.

Phase 1. The adversary $\mathcal{A}$ adaptively makes polynomially many queries as follows:

- $H_1(ID_u \parallel n)$ Oracle: For any query to $H_1(ID_u \parallel n)$, the challenger $\mathcal{B}$ checks the maintained H_1-table, i.e., $HT1$ using the index (ID_u, n). If not found, $\mathcal{B}$ randomly selects a bit $bt \in \{0, 1\}$ such that $\Pr[bt = 1] = \psi$, where ψ will be decided later. If $bt = 0$, $\mathcal{B}$ computes $HO_1 = \left(g^f\right)^{s_1}$, in which $s_1 \in Z_p$. If $bt = 1$, $\mathcal{B}$ calculates $HO_1 = g^{s_1}$. Finally, $HT1$ is updated as $HT1 \cup \{(ID_u, n, bt, s_1, HO_1)\}$, and HO_1 is returned to $\mathcal{A}$.
- $H_2(ID_u \parallel r_u)$ Oracle: For any query to $H_2(ID_u \parallel r_u)$, $\mathcal{B}$ checks the maintained H_2-table, i.e., $HT2$ using the index (ID_u, r_u). If not found, $\mathcal{B}$ calculates $HO_2 = g^{s_2}$ in which $s_2 \in Z_p$. Finally, $HT2$ is updated as $HT2 \cup \{(ID_u, r_u, s_2, HO_2)\}$, and HO_2 is returned to $\mathcal{A}$.

- $H_3(ID_u \parallel c)$ Oracle: For any query to $H_3(ID_u \parallel c)$, $\mathcal{B}$ checks the maintained H_3-table, i.e., $HT3$ using the index (ID_u, c). If not found, $\mathcal{B}$ calculates $HO_3 = g^{s_3}$ in which $s_3 \in Z_p$. Finally, $HT3$ is updated as $HT3 \cup \{(ID_u, c, s_3, HO_3)\}$, and HO_3 is returned to $\mathcal{A}$.
- Initial Private Key Query: For any initial private key queries of an identity ID_u, the challenger $\mathcal{B}$ uses ID_u as an index to search for an entry (ID_u, r_u, s_2, HO_2) in $HT2$. If not found, $\mathcal{B}$ simulates a query to the random oracle H_2 on behalf of $\mathcal{A}$ and then computes $sk_u^1 = P^{s_2}$, $sk_u^2 = r_u$, and $sk_u = (sk_u^1, sk_u^2)$. Finally, $\mathcal{B}$ returns sk_u to $\mathcal{A}$.
- Time-Update Key Query: For any time-update key queries of (ID_u, n), the challenger $\mathcal{B}$ searches the $HT1$ for an entry (ID_u, n, bt, s_1, HO_1). If not found, $\mathcal{B}$ simulates a query to the random oracle H_1 on behalf of $\mathcal{A}$. If $bt = 0$, $\mathcal{B}$ aborts. Otherwise, it computes $tk_{u,n} = P^{s_1}$ and returns it to the adversary $\mathcal{A}$.
- Re-Encryption Key Query: For any re-encryption key queries of $(ID_O, ID_R, n, c, M_{name})$ where $ID_O \neq ID_R$, and ID_R is a non-revoked user, $\mathcal{B}$ simulates the $H_3(ID_O \parallel c)$ oracle, the initial private key, and the time-update key queries to obtain the user private key $sk_{O,n}$ for the time period n, and then retrieves the corresponding entries from $HT1$, $HT2$, and $HT3$. If $bt = 0$, the query is aborted. If not, $\mathcal{B}$ chooses random values $\rho, \omega, x, \pi \in Z_p$ to compute $Q = g^{\rho}$, $W = HO_3$, $A_1 = W^x \cdot P^{\omega}$, $A_2 = \frac{(sk_{O,n}) \cdot P^{\omega}}{H_4(e(H_1(ID_R \| n) \cdot Q^{\pi}, P))}$, $A_3 = e(g^{\pi}, P)$, and $A_4 = g^x$. Finally, $\mathcal{B}$ gives the re-encryption key $AK = (A_1, A_2, A_3, A_4)$ to $\mathcal{A}$.
- Decryption Query: For any decryption queries of (ID_u, CT_{DR}, n') where ID_u is a valid user, $\mathcal{B}$ simulates the initial private key and the time-update key queries to obtain the private key $sk'_{u,n'}$ for the time period n'. However, if the corresponding entry $(ID_u, n', bt, s_1, HO_1)$ in $HT1$ satisfies that $bt = 0$, it will be aborted. Otherwise, $\mathcal{B}$ is capable of retrieving another entry (ID_u, r_u, s_2, HO_2) from $HT2$ and then calculating the private key $sk'_{u,n'} = P^{s_1+s_2}$ to run the decryption algorithm and return its result to $\mathcal{A}$.

Challenge. $\mathcal{A}$ chooses two symmetric keys (Y_0, Y_1) of equal length, the target user identity ${ID_u}^*$, a time period n^*, an access condition c^*, and the message $M^* = ({m_1}^*, {m_2}^*, \ldots, {m_s}^*)$. The challenger $\mathcal{B}$ takes the input of $(PP, Y_\lambda, {ID_u}^*, n^*, c^*, M^*)$ where $\lambda \in_R \{0,1\}$ to generate a challenge ciphertext $CT^* = ({C_1}^*, {C_2}^*, {C_3}^*, {C_4}^*, {CT_1}^*)$ for the adversary $\mathcal{A}$ as follows:

1. It can be assumed, without loss of generality, that $\mathcal{A}$ has already chosen a random r_u and queried the corresponding random oracles of ${ID_u}^*$. If $bt^* = 1$, the challenger $\mathcal{B}$ aborts.
2. Otherwise, $\mathcal{B}$ can retrieve the associated (s_1, s_2, s_3) from maintained tables and compute

$$
\begin{aligned}
&{C_1}^* = Y_\lambda \cdot \gamma^{s_1 \cdot r_u} \cdot e\left(\left(g^k\right)^{s_2 \cdot r_u}, P\right)\\
&{C_2}^* = \left(g^k\right)^{r_u}\\
&{C_3}^* = \left(g^f\right)^{s_1 \cdot r_u}\\
&{C_4}^* = \left(g^k\right)^{s_3 \cdot r_u}\\
&{CT_1}^* = \left(E(Y_\lambda, m_1^*), E(Y_\lambda, m_2^*), \ldots, E(Y_\lambda, m_s^*)\right)
\end{aligned}
$$

Phase 2. After getting the challenge ciphertext CT^*, the adversary $\mathcal{A}$ will go on Phase 1 queries under the constraints defined in Definition 1.

Guess. At the end of Phase 2, $\mathcal{A}$ outputs a bit λ'. If $\lambda' = \lambda$, the challenger $\mathcal{B}$ outputs 1, indicating $\gamma = e(g,g)^{fsk}$. Otherwise, it outputs 0, indicating $\gamma \neq e(g,g)^{fsk}$.

Analysis: The confidentiality of the proposed scheme is directly guaranteed by the hardness of the DBDH assumption. The core of this proof lies in how the challenger $\mathcal{B}$ uses the ability of adversary $\mathcal{A}$ to compromise confidentiality to solve the given DBDH instance. According to the challenge phase, as long as $e(g,g)^{fsk} = \gamma$, the generated ciphertext CT^* is correct, which indicates that the winning probability of $\mathcal{A}$

is non-negligible. Namely, $Adv(\mathcal{A}) = |\Pr[\lambda' = \lambda] - 1/2| \geq \varepsilon$. Conversely, whenever $e(g,g)^{fsk} \neq \gamma$, the ciphertext is invalid, and $\mathcal{A}$ gains no advantage in suspecting the right λ. Consequently, $Adv(\mathcal{A}) = |\Pr[\lambda' = \lambda]| = 1/2$. Let $\Pr[Perfect]$ be the probability that the entire simulation game does not abort. The advantage of the challenger $\mathcal{B}$ in solving the DBDH problem is thus given by the inequality below:

$$\begin{aligned}&\left|\Pr\left[\left(g, g^{f}, g^{s}, g^{k}, e(g,g)^{fsk}\right) = 1\right] - \Pr\left[\left(g, g^{f}, g^{s}, g^{k}, \gamma\right) = 1\right]\right| \\ &\quad \geq |(1/2 + \varepsilon) - 1/2| \cdot \Pr[Perfect] = \varepsilon \cdot \Pr[Perfect]\end{aligned}$$

To better estimate $\Pr[Perfect]$, we analyze the following event probabilities:

- $\Pr[\neg TkQ]$: All time-update key queries are handled without aborting.
- $\Pr[\neg PrQ]$: All re-encryption key queries are handled without aborting.
- $\Pr[\neg DeQ]$: All decryption queries are handled without aborting.
- $\Pr[\neg Ch]$: The challenge phase completes without aborting.

Since $\Pr[\neg TkQ]$, $\Pr[\neg PrQ]$, $\Pr[\neg DeQ]$, and $\Pr[\neg Ch]$ are independent events, the probability $\Pr[Perfect]$ can be expressed as

$$\Pr[\neg TkQ] \cdot \Pr[\neg PrQ] \cdot \Pr[\neg DeQ] \cdot \Pr[\neg Ch]$$

To be precise, in any time-update key query, $\mathcal{B}$ terminates when $bt = 0$, meaning $\Pr[\neg TkQ] \leq \Psi^{q_{tk}}$. In any re-encryption key query, $\mathcal{B}$ terminates when $bt = 0$, meaning $\Pr[\neg PrQ] \leq \Psi^{q_{pr}}$. In any decryption query, $\mathcal{B}$ terminates when $bt = 0$, meaning $\Pr[\neg DeQ] \leq \Psi^{q_{de}}$. In the challenge phase, $\mathcal{B}$ aborts provided that $bt^* = 1$, i.e., $\Pr[\neg Ch] \leq (1 - \Psi)$. By combining these probability events, we obtain

$$\Pr[Perfect] \leq (\Psi)^{q_{tk}} \cdot (\Psi)^{q_{pr}} \cdot (\Psi)^{q_{de}} \cdot (1 - \Psi) = (\Psi)^{q_{tk}+q_{pr}+q_{de}} \cdot (1 - \Psi)$$

By calculation, when Ψ is maximized at $1 - 1/(q_{tk} + q_{pr} + q_{de} + 1)$, the probability that the simulation process is perfect without aborting is at least $1/e(q_{tk} + q_{pr} + q_{de} + 1)$ where e is the base of natural logarithm. Therefore, the advantage ε' of the challenger $\mathcal{B}$ in breaking the problem of DBDH could be written as

$$\varepsilon' \geq \frac{\varepsilon}{e(q_{tk} + q_{pr} + q_{de} + 1)}. \square$$

5.2 Comparison

This subsection presents a functional comparison and performance analysis with similar studies, including Zhang et al.'s (PRE+ for short) [26], Yao et al.'s (R-CPRE-CE+ for short) [17], Fang et al.'s (C-PRE for short) [6], Zhou et al.'s (CL-CPRE for short) [22], and the Lin-Chen (R-PRE for short) [23]. Table 1 summarizes the comparison of functionality. It can be observed that the schemes of R-CPRE-CE+ and R-PRE do not support anonymous key generation, and neither of them, including the PRE+, can resist a malicious PKG. Among the schemes of PRE+, C-PRE, CL-CPRE, and R-PRE, only the R-PRE provides the property of time-update key, and most of them could not simultaneously fulfill conditional proxy re-encryption, time-update key, and ciphertext evolution.

Table 1: Comparative analysis of functionality

Feature	PRE+	R-CPRE-CE+	C-PRE	CL-CPRE	R-PRE	R-CPRE-CE
Anonymous key	✓	✗	✗	✗	✗	✓
Resistance to malicious PKG	✗	✗	✓	✓	✗	✓
Conditional proxy re-encryption	✗	✓	✗	✓	✗	✓
Time-update key	✗	✓	✗	✗	✓	✓
Ciphertext evolution	✗	✓	✗	✓	✗	✓
Without revocation list	✗	✓	✓	✓	✓	✓
Resistance to revoked user	✗	✓	✗	✗	✓	✓
CCA security	✗	✓	✓	✗	✗	✓

Note: Remark: "✓" indicates that the feature is supported, and "✗" means not supported.

In Table 2, we focus on computational complexity and evaluate time-consuming operations. Among these mechanisms, ours and R-PRE have the same computational advantage in the generation of time-update key algorithm. In particular, our system is efficient in the Setup phase, requiring only one exponentiation (E). The C-PRE scheme has to additionally generate a strongly unforgeable one-time signature. Besides, our system is one of only three schemes to feature the Evolve function, which is crucial for forward-security or state updates. Its cost is a single pairing (B), which is significantly more efficient than the costs of R-CPRE-CE+ and CL-CPRE schemes. The Decryption (DO) cost is just B, on par with R-PRE, the most efficient scheme in this category. Although the computational complexity of ReKeyGen and Re-Encryption in our system is slightly higher than compared works, the proposed system offers a strong balance of performance and advanced features, making it a better choice for practical applications. Fig. 3 illustrates the approximate running time using the test environment of Intel Core i5-8250U CPU @ 1.60 GHz, 2.7 GB RAM, and Ubuntu 18.04.4 LTS with the Pairing-Based Cryptography (PBC) library (version 0.5.14).

Table 2: Comparative analysis of computational cost

Phase	PRE+	R-CPRE-CE+	C-PRE	CL-CPRE	R-PRE	R-CPRE-CE
Setup	2E	2B + E + 2F	*	B + E	E	E
InitialKeyGen	5E	5E	4E	6E + F	E	3E
TimeUpdate KeyGen	–	3E	–	–	E	E
Encryption	B + 2E + F	B + 7E + F	3B + 3F	B + 2E + F	B + 2E	B + 5E
Query	E	–	–	–	E	E
ReKeyGen	2E	6E	4E	F	2B + 3E	2B + 6E
Evolve	–	5B + 3E	–	2B + 2E + F	–	B
Re-Encryption	B	6B + 3E	B + E + F	2B	B	2B

(Continued)

Table 2 (continued)

Phase	PRE+	R-CPRE-CE+	C-PRE	CL-CPRE	R-PRE	R-CPRE-CE
Decryption (DO)	2B	5B + 3E	B + 2F	2B + E + F	B	B
Decryption (DR)	2B + E	5B + 3E	B + 3F	F	3B + F	3B + F

Note: Remark: A bilinear pairing is denoted by "B" and exponentiation in G_1 and G_2 is denoted as "E" and "F", respectively. The symbols "-" and "*" are used to denote an unsupported operation and an unforgeable one-time signature.

As shown in Fig. 3, R-PRE and ours only need 2.893 ms to run Setup. In the InitialKeyGen phase, our scheme takes 8.679 ms. In the TimeUpdateKeyGen phase, R-PRE and our schemes need a runtime of 2.893 ms which is better than that of 8.679 ms in R-CPRE-CE+ scheme. Our scheme is also efficient in Evolve (13.864 ms) and Decryption by DO (13.864 ms). The PRE+ scheme has the best runtime of 5.786 ms in ReKeyGen. Although the R-PRE scheme has a comparable runtime in TimeUpdateKeyGen, Encryption and Re-Encryption, they fail to provide the functionality of ciphertext evolution and anonymous key generation. Since the R-CPRE-CE+ scheme also supports user revocation and ciphertext evolution, we further make a detailed evaluation with our system in Fig. 4. Note that in the comparison of Fig. 4, we just consider core algorithms including Encryption, ReKeyGen, Evolve, Re-Encryption, and Decryption by DR. Here, we also added a one-time access metric to evaluate the total computation time required for a data user to complete the entire process from sending a query to the final decryption. From the comparison illustrated in Fig. 3, it is evident that the R-CPRE-CE+ scheme is more efficient in the ReKeyGen algorithm, which requires a runtime of only 17.358 ms compared to 45.086 ms for our proposed scheme. Nevertheless, our scheme still outperforms the R-CPRE-CE+ scheme in all other core algorithms. As for the one-time access metric, which simulates the total computation time from an initial query to the final decryption in practice, the result shows that our scheme reduces the runtime by approximately 47.05% when compared to 265.219 ms of the R-CPRE-CE+ scheme.

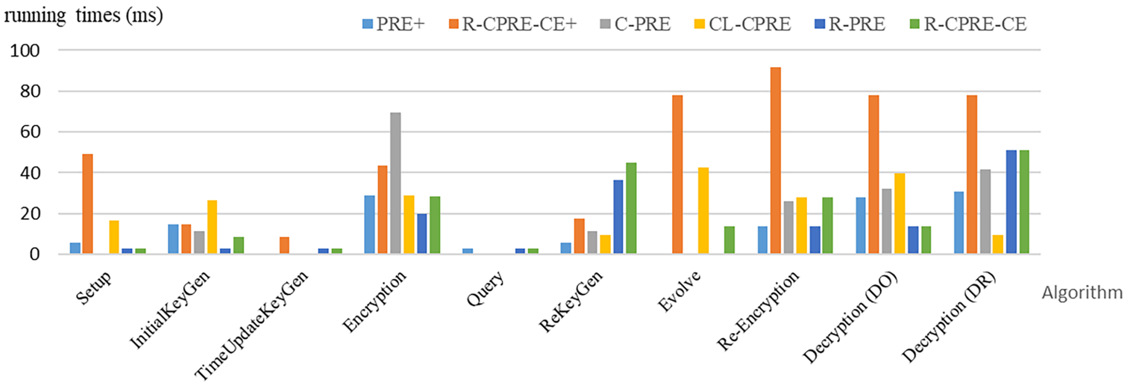

Figure 3: Comparison of approximate running times

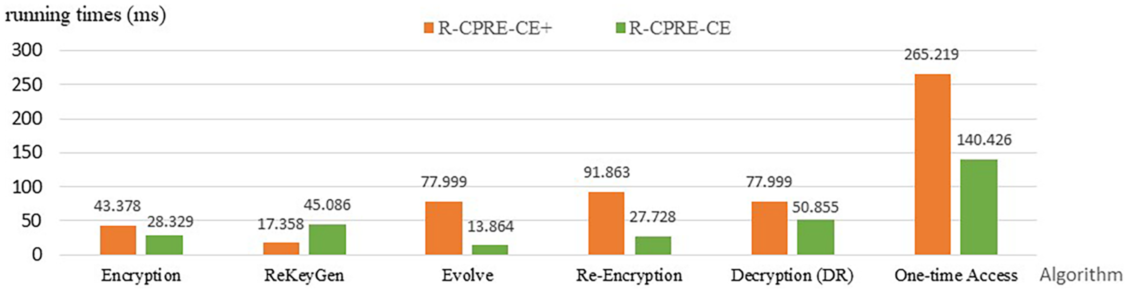

Figure 4: Comparison between the R-CPRE-CE+ and our R-CPRE-CE schemes

6 Discussion

This section discusses the practical applications and academic contributions of our proposed R-CPRE-CE scheme. We aim to highlight how our scheme addresses critical challenges in fog-assisted cloud data sharing and give a potential direction for future research.

6.1 Practical Applications

As to the assessment of our framework's effectiveness, it is essential to consider not only runtime performance, but also its accuracy. In the context of cryptographic schemes, the concept of "accuracy" is formally defined as correctness. As shown in Section 3.2, our R-CPRE-CE scheme is well-designed and correct, because any unrevoked user holding a valid private key can decrypt the corresponding ciphertext with the probability of 1. Such a deterministic guarantee is a fundamental prerequisite for any secure encryption system and ensures the reliability of our framework. Thus, the effectiveness of our scheme is demonstrated by its proven correctness and the results of computational evaluation detailed in Section 5.2.

Concretely speaking, the proposed R-CPRE-CE scheme is designed to provide a secure and integrated data sharing mechanism for distributed systems, especially in fog-assisted clouds. In the application of healthcare management, a data owner needs to revoke the access privilege of certain data users due to privacy concerns. Our system employs a periodically updated time key to achieve this without re-encrypting the original data. For the application of intelligent transportation systems, in which data gathered from vehicles and mobile sensors must remain valid for continuous time periods, the ciphertext evolution of our scheme provides crucial benefits. Additionally, the incorporation of a predefined access condition can prevent unauthorized re-encryption by a semi-trusted proxy. This property is important for assuring data confidentiality and integrity in diverse IoT environments where data delegation has to be strictly controlled.

A crucial issue of the ciphertext evolution is its security guarantees across multiple time periods, i.e., forward and backward secrecy. The proposed R-CPRE-CE scheme is designed to ensure forward secrecy, since the compromise of a previous private key does not leak the confidentiality of current and future ciphertexts. This is because each time-update key is derived from a unique hash value. Consequently, an adversary who even obtains an old time-update key cannot derive another valid one for the subsequent period. Similarly, the backward secrecy is also fulfilled in the proposed scheme, because the compromise of a current key will not result in the successful derivation of past keys. Regarding the expressiveness of the access condition, our current construction supports a single, atomic condition c for efficiency. The framework can easily support conjunctive (AND) conditions without increasing overhead. Specifically, we can express the access condition c to be a concatenation of multiple attributes (e.g., c = "role: doctor || department: cardiology") before it is hashed. However, supporting disjunctive (OR) conditions is non-trivial and would take significant changes in algorithms or increase ciphertext size.

6.2 Academic Contributions

From the perspective of academic contributions, our work addresses the inherent problem of key-escrow in traditional identity-based cryptosystems, such as those by Yao et al. [17,18], by employing an anonymous key generation technique. Although the existing literature has discussed the individual aspects such as revocability (e.g., the Lin-Chen [23]), ciphertext evolution (e.g., Yao et al. [17]), and conditional access control, an integrated mechanism with provable security is still lacking. The proposed R-CPRE-CE scheme successfully fulfills this goal by combining the above functionalities. Despite these contributions, we recognize that the main limitation of our current system is that it does not yet achieve standard model security and requires further efforts to reduce the computational complexity. The computational costs of the ReKeyGen and Re-Encryption algorithms in our scheme are higher than some counterparts, which warrants a deeper analysis of its impact on real-time performance. This higher cost is a trade-off for better functionalities such as conditional access and ciphertext evolution. Crucially, the most computationally intensive operation, ReKeyGen, is performed by a data owner, who typically has the sufficient computing power. The Re-Encryption task, performed by the fog node, has a quantifiable latency (approximately 27 ms in our tests) that is manageable for modern fog devices. Although this design mitigates the direct impact on the real-time responsiveness of fog nodes, a further optimization is still a key future research.

The credibility of the proposed framework would benefit from quantitative ablation studies and parameter sensitivity analyses. An ablation study analyzes the impact of performance by removing individual components such as the conditional access control or the ciphertext evolution. However, these components are integrated as the core security objectives of our scheme. For instance, the removal of conditional access control will bring back the over-delegation problem, while that of ciphertext evolution would break forward security. Therefore, our design represents an integration of essential functionalities rather than a collection of separable modules. Similarly, for parameter sensitivity analysis, the computational cost of the adopted cryptographic primitives (i.e., bilinear pairings and exponentiations) scales with the chosen security parameter. It would be a valuable future direction to make a more granular analysis across various aspects.

7 Conclusions

In this paper, we proposed a fog-assisted R-CPRE-CE system, i.e., revocable conditional proxy re-encryption scheme offering ciphertext evolution. Our scheme aims at dealing with two challenges of cloud data sharing: user revocation and cloud ciphertext evolution. The former is achieved by using a periodically updated time key while the latter is realized via an evolution parameter generated by the data owner. In addition, to further restrict the unauthorized behaviors of the malicious proxy, an access condition is incorporated with the proposed re-encryption mechanism. Specifically, we utilize anonymous key generation technique, so as to prevent the key-escrow problem of identity-based systems. From the perspective of security, we formally prove that the proposed system satisfies the security model of IND-PrID-CCA under the assumption of DBDH. Compared with previous similar studies, our scheme provides not only competitive performance, but also more comprehensive functionalities. However, there are still some limitations of our current system that would be important directions for future work. First, the underlying security proof is constructed in the random oracle model. It would be better to adapt the current mechanism for a more rigorous security proof in the standard model. Second, to appeal to more resource-constrained IoT mobile applications, the computational complexity of ReKeyGen and Re-Encryption algorithms requires further optimization. Third, to support more complex access policies, such as conjunctive and disjunctive conditions, would increase more practical applicability. Finally, conducting ablation studies and a comprehensive parameter sensitivity analysis would further enhance the robustness and credibility of our framework.

Acknowledgement: Not applicable.

Funding Statement: This work was supported in part by the National Science and Technology Council of Republic of China under the contract numbers NSTC 114-2221-E-019-055-MY2 and NSTC 114-2221-E-019-069.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Han-Yu Lin and Yi-Jia Ye; methodology, Yi-Jia Ye; validation, Han-Yu Lin, Tung-Tso Tsai and Yi-Jia Ye; formal analysis, Yi-Jia Ye; resources, Tung-Tso Tsai; data curation, Yi-Jia Ye; writing—original draft preparation, Han-Yu Lin; writing—review and editing, Tung-Tso Tsai; supervision, Han-Yu Lin; funding acquisition, Han-Yu Lin. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Not applicable.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Mell P, Grance T. The NIST definition of cloud computing. Nat Inst Stand Technol. 2011;53:1–7.
2. de Brito MS, Hoque S, Steinke R, Willner A, Magedanz T. Application of the fog computing paradigm to smart factories and cyber-physical systems. Trans Emerg Telecommun Technol. 2018;29(4):e3184. doi:10.1002/ett.3184.
3. Chen Y. The application of IoT-assisted intelligent transportation system. In: Proceedings of 2023 Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC); 2023 Apr 14–16; Dalian, China. p. 317–20. doi:10.1109/IPEC57296.2023.00062.
4. Chen Z, Li S, Huang Q, Wang Y, Zhou S. A restricted proxy re-encryption with keyword search for fine-grained data access control in cloud storage. Concurr Comput Pract Exp. 2016;28(10):2858–76. doi:10.1002/cpe.3754.
5. Deng RH, Weng J, Liu S, Chen K. Chosen-ciphertext secure proxy re-encryption without pairings. In: Cryptology and network security. Berlin/Heidelberg, Germany: Springer; 2008. p. 1–17. doi:10.1007/978-3-540-89641-8_1.
6. Fang L, Susilo W, Ge C, Wang J. Chosen-ciphertext secure anonymous conditional proxy re-encryption with keyword search. Theor Comput Sci. 2012;462(2):39–58. doi:10.1016/j.tcs.2012.08.017.
7. Liang K, Liu JK, Wong DS, Susilo W. An efficient cloud-based revocable identity-based proxy re-encryption scheme for public clouds data sharing. In: Computer security—ESORICS 2014. Cham, Switzerland: Springer International Publishing; 2014. p. 257–72. doi:10.1007/978-3-319-11203-9_15.
8. Mizuno T, Doi H. Hybrid proxy re-encryption scheme for attribute-based encryption. In: Information security and cryptology. Vol. 302. Berlin/Heidelberg, Germany: Springer; 2010. p. 288–302. doi:10.1007/978-3-642-16342-5_21.
9. Shao J, Cao Z, Liang X, Lin H. Proxy re-encryption with keyword search. Inf Sci. 2010;180(13):2576–87. doi:10.1016/j.ins.2010.03.026.
10. Son J, Kim D, Hussain R, Oh H. Conditional proxy re-encryption for secure big data group sharing in cloud environment. In: 2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS); 2014 Apr 27–May 2; Toronto, ON, Canada. p. 541–6. doi:10.1109/INFCOMW.2014.6849289.
11. Canetti R, Hohenberger S. Chosen-ciphertext secure proxy re-encryption. In: Proceedings of the 14th ACM Conference on Computer and Communications Security; 2007 Oct 29–Nov 2; Alexandria, VA, USA. p. 185–94. doi:10.1145/1315245.1315269.
12. Green M, Ateniese G. Identity-based proxy re-encryption. In: Applied cryptography and network security. Berlin/Heidelberg, Germany: Springer; 2007. p. 288–306. doi:10.1007/978-3-540-72738-5_19.
13. Weng J, Deng RH, Ding X, Chu CK, Lai J. Conditional proxy re-encryption secure against chosen-ciphertext attack. In: Proceedings of the 4th International Symposium on Information, Computer, and Communications Security; 2009 Mar 10–12; Sydney, Australia. p. 322–32. doi:10.1145/1533057.1533100.

14. Chu CK, Weng J, Chow SSM, Zhou J, Deng RH. Conditional proxy broadcast re-encryption. In: Information security and privacy. Berlin/Heidelberg, Germany: Springer; 2009. p. 327–42. doi:10.1007/978-3-642-02620-1_23.
15. Shao J, Wei G, Ling Y, Xie M. Identity-based conditional proxy re-encryption. In: 2011 IEEE International Conference on Communications (ICC); 2011 Jun 5–9; Kyoto, Japan. p. 1–5. doi:10.1109/icc.2011.5962419.
16. Zeng P, Choo KR. A new kind of conditional proxy re-encryption for secure cloud storage. IEEE Access. 2018;6:70017–24. doi:10.1109/access.2018.2879479.
17. Yao S, Dayot RVJ, Kim HJ, Ra IH. A novel revocable and identity-based conditional proxy re-encryption scheme with ciphertext evolution for secure cloud data sharing. IEEE Access. 2021;9:42801–16. doi:10.1109/access.2021.3064863.
18. Yao S, Dayot RVJ, Ra IH, Xu L, Mei Z, Shi J. An identity-based proxy re-encryption scheme with single-hop conditional delegation and multi-hop ciphertext evolution for secure cloud data sharing. IEEE Trans Inf Forensics Secur. 2023;18(2):3833–48. doi:10.1109/TIFS.2023.3282577.
19. Worapaluk K, Fugkeaw S. An efficiently revocable cloud-based access control using proxy re-encryption and blockchain. In: 2023 20th International Joint Conference on Computer Science and Software Engineering (JCSSE); 2023 Jun 28–Jul 1; Phitsanulok, Thailand. p. 178–83. doi:10.1109/JCSSE58229.2023.10202130.
20. Singh MR, Barik RK, Qurashi SN, Thokchom S, Roy DS. A novel pairing free revocable certificateless encryption with ciphertext evolution for healthcare system. IEEE Access. 2025;13:27940–51. doi:10.1109/access.2025.3533367.
21. Eltayieb N, Elhabob R, Abdelgader AMS, Liao Y, Li F, Zhou S. Certificateless proxy re-encryption with cryptographic reverse firewalls for secure cloud data sharing. Future Gener Comput Syst. 2025;162(12):107478. doi:10.1016/j.future.2024.08.002.
22. Zhou Y, Li Y, Liu Y. A certificateless and dynamic conditional proxy re-encryption-based data sharing scheme for IoT cloud. J Internet Technol. 2025;26(2):165–72. doi:10.70003/160792642025032602002.
23. Lin HY, Chen PR. Revocable and fog-enabled proxy re-encryption scheme for IoT environments. Sensors. 2024;24(19):6290. doi:10.3390/s24196290.
24. Lin HY, Tsai TT, Ting PY, Chen CC. An improved ID-based data storage scheme for fog-enabled IoT environments. Sensors. 2022;22(11):4223. doi:10.3390/s22114223.
25. Lin HY, Tsai TT, Ting PY, Fan YR. Identity-based proxy re-encryption scheme using fog computing and anonymous key generation. Sensors. 2023;23(5):2706. doi:10.3390/s23052706.
26. Zhang J, Bai W, Wang X. Identity-based data storage scheme with anonymous key generation in fog computing. Soft Comput. 2020;24(8):5561–71. doi:10.1007/s00500-018-3593-z.