JOURNAL OF INFORMATION SCIENCE AND ENGINEERING XX, XXXX-XXXX (2024)
DOI: XX.XXX.XXXX.XX

HRFPRE: Fast Proxy Re-encryption for Multi-RSU Outsourcing and Hardware-assisted Revocation in the IoV

TINGTING LI[1,3], LEYOU ZHANG[1,3,+], QING WU[2]
Fei Zhou[3] and Yuxing Wei[3]
[1]*School of Mathematics and Statistics*
Xidian University
Xi'an 710121, China
[2]*School of Automation*
Xi'an University of Posts and Telecommunications
Xi'an 710121, China
[3]*Guangxi Key Laboratory of Digital Infrastructure*
Guangxi Zhuang Autonomous Region Information Center
Nanning 530000, China
E-mail: 18661501165@163.com; lyzhang@mail.xidian.edu.cn; xiyouwuq@126.com
syszhoufei@163.com; weiyx@gxi.gov.cn

In the Internet of Vehicles (IoV), content-centric data sharing is essential for driving safety and user experience. However, the highly dynamic and distributed IoV network raises challenges such as unauthorized data access and inefficient information dissemination. Although existing proxy re-encryption (PRE) schemes with revocation partially mitigate these concerns, they still have key shortcomings: (i) computational costs that grow linearly with the number of attributes; (ii) heavy cloud-side overhead from re-encryption and outsourced decryption, causing delays or decryption failures; and (iii) revocation mechanisms that are inefficient or insufficiently responsive in handling malicious users. Recent studies have addressed these issues, but many schemes still struggle to ensure reliable message recovery in dynamic IoV scenarios.

To overcome these limitations, we propose HRFPRE, an efficient proxy re-encryption mechanism based on multi-RSU outsourcing and hardware-assisted revocation. Our scheme provides constant-size public parameters and lightweight user-side decryption over asymmetric pairing-friendly groups, while supporting an unbounded attribute space. Re-encryption requires only four pairing operations and supports a novel "encrypt-then-offline hosting" model for vehicles. Simultaneously, Roadside Units (RSUs) provide outsourced re-encryption, key generation assistance, and decryption services to resource-constrained onboard units, effectively shifting computational load away from the cloud. By integrating a key-decoupled Trusted Execution Environment (TEE), HRFPRE enables immediate revocation and keeps plaintext recovery dependent on the user-held key even under TEE-side side-channel leakage. Under the Decisional Linear (DLIN) assumption, HRFPRE achieves adaptive security while resisting replay and collusion attacks. Theoretical analysis and experiments show that HRFPRE reduces computational and communication overhead, making it suitable for secure data exchange in dynamic IoV environments.

Keywords: Internet of Vehicles (IoV), Attribute-based Proxy Re-encryption, Revocation, Security, Data Sharing

Received TODO; revised TODO; accepted TODO.
[+]Corresponding author

1. INTRODUCTION

With the deepening integration between automobiles and the digital ecosystem, the Internet of Vehicles (IoV) and cloud-based solutions have made significant progress, while connected-vehicle technologies and autonomous driving are undergoing rapid innovation. According to industry forecasts, the IoV market is expected to grow from USD 274.7 billion in 2025 to USD 395.4 billion in 2030, with a CAGR of 7.56% [1]. IoV technologies enable vehicles to collect and transmit data on traffic conditions, route planning, charging stations, vehicle failures, and accident locations, achieving real-time vehicle-to-vehicle data sharing that enhances driving safety and user experience [2].

Driven by advances in 6G communications, data transmission, and electronic sensing, vehicle safety functions and communication technologies continue to evolve. Nevertheless, in highly dynamic IoV environments, efficient processing of massive data, prevention of data leakage and cyber attacks, and flexible secure inter-vehicle data sharing still face pressing challenges [3, 4]. These challenges are mainly reflected in three aspects: (1) how to ensure confidentiality during data sharing, since shared vehicle data such as road conditions, speed, and destination contain sensitive private information and must be accessible only to authorized vehicles [5]; (2) how to achieve efficient cross-vehicle ciphertext sharing, since high mobility and limited onboard resources require low-overhead and time-efficient sharing schemes in dynamic environments; and (3) how to promptly revoke malicious vehicles once they attempt to tamper with or forge information [6].

Identity-based encryption (IBE) [7] and attribute-based encryption (ABE) [8] provide potential solutions to data confidentiality. IBE enables efficient identity-based decryption but only supports one-to-one sharing, whereas ABE enables one-to-many fine-grained access control through policies. However, ABE computation usually grows linearly with the attribute-set size, imposing a heavy burden on resource-constrained IoV devices [9, 10]. Although traditional outsourced decryption schemes shift complex computation and storage to cloud servers [11], centralized clouds still struggle with the growing computational and storage pressure caused by massive IoV users.

Proxy re-encryption (PRE) [12, 13] can transform IBE ciphertexts into ABE ciphertexts, combining IBE efficiency with ABE-based flexible access control. Existing IBE-to-ABE PRE schemes improve security, efficiency, and functionality [14], and some incorporate outsourced decryption and revocation mechanisms to keep user-side computation constant [15].

However, existing schemes [16, 17] still have several limitations. First, revocation mechanisms typically rely on periodically updating re-encryption keys or on server-side ciphertext delegation to block revoked users. Second, re-encryption and outsourced decryption performed by cloud service providers still incur considerable computational pressure. Third, most schemes only satisfy selective-security models, making it difficult to meet the rigorous security requirements of IoV.

To overcome these obstacles, we first propose FPRE-OD, a novel unbounded outsourced PRE scheme that combines the benefits of IBE and ABE. This scheme achieves adaptive security in the random oracle model under the DLIN assumption and addresses the needs for computational efficiency, access flexibility, and communication security. Inspired by hardware-based ABE revocation schemes [18], we further propose HRFPRE, a proxy re-encryption mechanism based on multi-RSU outsourcing and hardware-assisted

revocation. In HRFPRE, an authority maintains vehicle status lists and time-indexed credentials, while a data owner encrypts messages under a trusted delegator's identity so that the sender can go offline after encryption. The delegator supports key escrow and generates re-encryption keys for ciphertext transformation, while RSUs assist in partial re-encryption-key generation, re-encryption, status-list verification, and outsourced decryption, reducing the computational burden on both clouds and vehicles. Since RSUs are semi-trusted, a trusted execution environment (TEE) establishes an authenticated channel with the authority for secondary status verification and generates helper ciphertexts for non-revoked users. Thus, the final recipient can decrypt with minimal computation. The proposed scheme is proven adaptively secure under the DLIN assumption. Our work highlights the following main contributions:

(1) We introduce an efficient IoV data-sharing model with fine-grained access control, restricting data access to authorized vehicles only. Through delegation, it realizes key escrow and allows the data sender to go offline after encryption. The model also provides unbounded extensibility, with constant-size public parameters regardless of the number of attributes.

(2) We design a hardware-based fast revocation mechanism for highly dynamic multi-RSU environments. By converting ciphertexts from IBE to ABE through proxy re-encryption, HRFPRE offloads complex computations to RSUs and uses a TEE to verify user status, achieving immediate revocation, efficient outsourced decryption, and reduced ciphertext delegation and key updates. Furthermore, HRFPRE introduces a key-decoupling mechanism with strict time synchronization: TEE-held helper keys alone cannot recover plaintext without the user-held UK, and any revocation bypass caused by TEE leakage is bounded within the current short time epoch. During user authentication, Verkle-tree-based vector commitments further limit authentication overhead to $O(\log n)$. Extensive evaluations show that HRFPRE outperforms existing schemes in computation and communication efficiency.

(3) We achieve adaptive security. The original FPRE-OD scheme satisfies adaptive IND-CPA security in the random oracle model, and we formally prove through hybrid games that HRFPRE achieves adaptive security under the DLIN assumption while resisting replay and collusion attacks.

2. RELATED WORK

Secure Data Sharing Schemes: To achieve secure information sharing over insecure channels, public key encryption has been widely applied. Boneh et al. [19] introduced the first pairing-based identity-based encryption (IBE) scheme, enabling confidential transmission based on the receiver's identity. However, IBE only supports one-to-one sharing, limiting its flexibility in IoV scenarios. To support fine-grained access control, Goyal et al. [8] proposed Attribute-Based Encryption (ABE), and its ciphertext-policy variant (CP-ABE) [20] allows data owners to embed access policies into ciphertexts so that only users with matching attributes can decrypt. CP-ABE has therefore been widely used in IoV data sharing and storage. Later works improved CP-ABE in security, efficiency, and functionality [10, 21], while Green et al. [11] introduced outsourced decryption. Agrawal et al. [22] further proposed an unbounded ABE scheme with adaptive security

under standard assumptions. Nevertheless, these schemes often suffer from large key or ciphertext sizes caused by complex access structures, creating computational and communication bottlenecks in resource-constrained IoV environments.

Regarding ciphertext transformation, Blaze et al. [12] first introduced proxy re-encryption (PRE). Inspired by IBE, identity-based PRE schemes [13] have been used in IoV for anonymous reception and privacy protection, but they generally lack fine-grained access control. To improve the adaptability of ABE frameworks, Liang et al. [23] proposed attribute-based PRE, enabling ciphertext transformation under different access policies. Subsequent studies extended PRE toward verifiability [24], multi-authority architectures [25], and collaborative edge computing [26]. However, most existing schemes rely on homogeneous cryptosystems and are difficult to apply to cross-domain sharing among heterogeneous cryptographic entities in IoV. Wang et al. [14] proposed an unbounded cross-domain PRE model combining IBE and ABE with standard-model security, and Zhang et al. [27] improved efficiency through outsourced re-encryption and decryption. Even so, IoV still lacks a systematic solution for secure and efficient cross-domain data sharing under high network dynamics and potential attacks.

Efficient User Revocation Mechanisms: In ABE systems, user revocation mechanisms are mainly divided into direct revocation [28] and indirect revocation schemes [29, 30]. The direct approach embeds a revocation list directly within the ciphertext, whereas the indirect method relies on updating keys at specific time intervals. Recently, some works have introduced server assistance [31] or trusted execution environments (TEE) [18] to optimize revocation efficiency and security. However, research on revocation for ABE re-encryption schemes remains scarce. Ge et al. [17] proposed direct revocation by issuing new re-encryption keys, ensuring that revoked users cannot decrypt, but this scheme incurs high computational overhead. Li et al. [16] proposed an efficient PRE scheme supporting dynamic permission management and utilized puncturable re-encryption key updates to achieve flexible revocation, yet it relies on specific policy structures ("AND" gates). Li et al. [15] proposed EPREAR, a cross-domain PRE scheme supporting attribute revocation and outsourced decryption, and utilizing blockchain to ensure verifiability, but its applicability in environments with extremely high real-time requirements like IoV remains to be examined.

3. PRELIMINARIES

3.1 Access Structures and LSSS

Let U be the universe of attributes. A monotone access structure $\mathbb{A}$ over U is a collection of authorized attribute sets such that, if $B \in \mathbb{A}$ and $B \subseteq C$, then $C \in \mathbb{A}$ [21]. We represent an LSSS access policy by $(\mathbb{M}, \rho)$ [20], where $\mathbb{M} \in \mathbb{Z}_p^{l \times n}$ is a share-generating matrix and ρ maps each row to an attribute. For a secret s, choose $\vec{v} = (s, v_2, \ldots, v_n)^T$ and define $\lambda_i = \mathbb{M}_i \vec{v}$. If an attribute set S satisfies the policy, then for $I = \{i : \rho(i) \in S\}$ there exist constants $\{\omega_i\}_{i \in I}$ such that

$$\sum_{i \in I} \omega_i \mathbb{M}_i = (1, 0, \ldots, 0), \quad \sum_{i \in I} \omega_i \lambda_i = s. \tag{1}$$

Table 1. Notations

Notation	Definition	Notation	Definition
$\mathbb{G};\mathbb{G}_1,\mathbb{G}_2,\mathbb{G}_T$	Symmetric/asymmetric pairing groups	par	Public parameters
MSK	System master key	$\mathscr{H},\mathscr{F}$	Hash functions
ID	User identity	S	Attribute set
ISK	IBE user secret key	UK	ABE user secret key
CK	Transformation key	TK	Helper key
RK_{out}	Re-encryption outsourcing key	msg	Plaintext data
CT	Original ciphertext	REC	Re-encrypted ciphertext
PTC	Transformed ciphertext	TC	Helper ciphertext
$List$	Vehicle user list	UL	Delegator user list
σ	Vehicle user list signature	$(\mathbb{M},\rho)$	LSSS access policy

3.2 Asymmetric Pairing Groups

Let $\mathsf{GroupGen}(1^{\mathscr{K}})$ output $par=(p,\mathbb{G}_1,\mathbb{G}_2,\mathbb{G}_T,e,g_1,g_2)$ [22], where $\mathbb{G}_1,\mathbb{G}_2,\mathbb{G}_T$ are cyclic groups of prime order p, g_1,g_2 are generators of $\mathbb{G}_1,\mathbb{G}_2$, and $e:\mathbb{G}_1\times\mathbb{G}_2\to\mathbb{G}_T$ is a bilinear map. It satisfies $e(g_1^a,g_2^b)=e(g_1,g_2)^{ab}$ for all $a,b\in\mathbb{Z}_p$, and is non-degenerate, i.e., $e(g_1,g_2)\neq 1$.

3.3 Decision Linear (DLIN) Assumption

We use the asymmetric DLIN assumption [21]. Let $par=(p,\mathbb{G}_1,\mathbb{G}_2,\mathbb{G}_T,e,g_1,g_2)\leftarrow\mathsf{GroupGen}(1^{\mathscr{K}})$, and sample $x_1,x_2,y_1,y_2,R\leftarrow\mathbb{Z}_p$. Define

$$D=\left(g_1^{x_1},g_1^{x_2},g_2^{x_1},g_2^{x_2},g_1^{x_1y_1},g_1^{x_2y_2},g_2^{x_1y_1},g_2^{x_2y_2}\right),T_0=\left(g_1^{y_1+y_2},g_2^{y_1+y_2}\right),T_1=\left(g_1^{R},g_2^{R}\right). \tag{2}$$

The advantage of a PPT algorithm $\mathscr{A}$ is

$$\mathrm{Adv}_{\mathscr{A}}^{\mathrm{DLIN}}(1^{\mathscr{K}})=\left|\Pr[\mathscr{A}(par,D,T_0)=1]-\Pr[\mathscr{A}(par,D,T_1)=1]\right|. \tag{3}$$

The DLIN assumption holds if this advantage is negligible for PPT algorithm $\mathscr{A}$.

4. SYSTEM ARCHITECTURE

4.1 System Model

The proposed scheme involves seven entities. Table 1 lists the main notations, and Fig. 1 shows the system architecture.

1. Key Generation Center (KGC): A fully trusted authority for system setup, vehicle registration, key generation, and revocation management. It generates PK and MSK, issues IBC and ABC keys, maintains and signs the time-based legitimate vehicle user list, uploads credentials to the CSP, and synchronizes the authenticated Delegator list.

2. Data Owner (DO): Collects traffic data, encrypts it before sharing, uploads ciphertext indices to the CSP, and designates a trusted Delegator (DL) for key escrow and authorized forwarding.

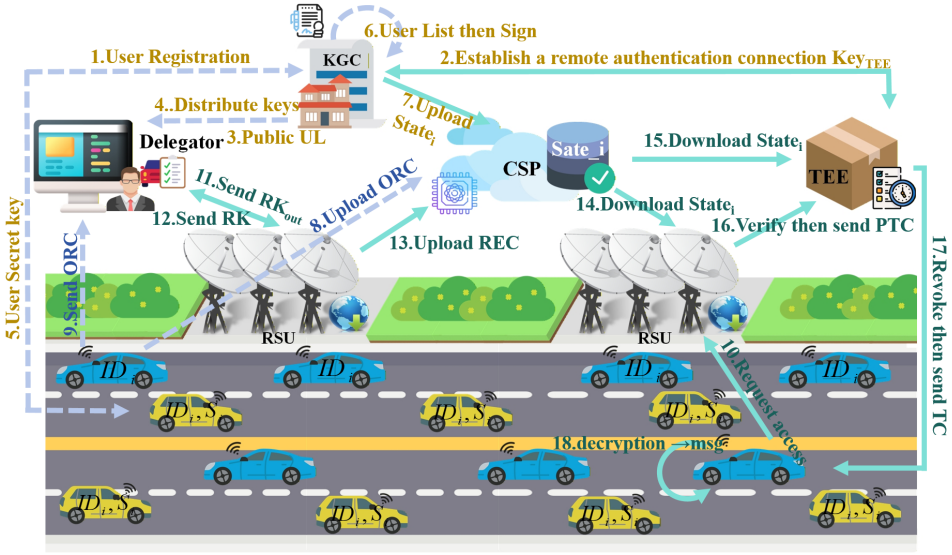


Fig. 1. System Architecture of HRFPRE.

3. Delegator (DL): A KGC-authenticated entity designated by the DO. It holds the IBE secret key, supports original-ciphertext decryption when needed, and generates re-encryption keys for ciphertext transformation.

4. Cloud Service Provider (CSP): Provides persistent ciphertext storage and returns the requested ciphertext and signed user-list state for the current time index to RSUs.

5. Roadside Unit (RSU): Provides storage and computation for outsourced key generation, re-encryption, permission verification, and partial decryption. It collaborates with the DL, obtains ciphertexts from the CSP, verifies requestors against the signed state, and outputs transformed ciphertexts.

6. Trusted Execution Environment (TEE): An independent secure enclave deployed on the CSP. It is logically separated from the semi-trusted CSP/RSU service and exposes only authenticated interfaces to RSUs. It establishes a remote-attestation channel with the KGC, verifies transformed ciphertexts and the latest user status, and generates helper ciphertexts for non-revoked authorized users.

7. Data User (DU): Sends access requests through the RSU and decrypts the content with its attribute private key after receiving the helper ciphertext from the TEE.

4.2 Algorithms

The FPRE-OD scheme mainly consists of the following algorithms.

- $\mathsf{Setup}(1^{\mathscr{K}})$: The KGC generates the public parameters PK and master secret key MSK.
- $\mathsf{Keygen}_{\mathsf{IBE}}(PK, MSK, ID)$: The KGC generates the identity secret key ISK for a registered delegator.
- $\mathsf{Keygen}_{\mathsf{ABE}}(PK, MSK, S)$: The KGC generates $ABK = (CK, TK, UK)$ for an attribute set S, where CK, TK, and UK are the transformation, helper, and user keys.

- $\mathsf{ibEncryption}(PK, ID, msg)$: The DO encrypts msg under the DL identity ID and outputs the original ciphertext CT.
- $\mathsf{ibDecryption}(PK, ISK, CT)$: The DL decrypts the original ciphertext and recovers msg.
- $\mathsf{Rekeygen}_{\mathsf{out}}(PK, (\mathbb{M}, \rho))$: The RSU generates the re-encryption outsourcing key RK_{out} for access policy $(\mathbb{M}, \rho)$.
- $\mathsf{Rekeygen}(PK, RK_{out}, ISK, (\mathbb{M}, \rho))$: The DL generates the re-encryption key RK.
- $\mathsf{abDecryption}_{\mathsf{trans}}(PK, CK, REC)$: The RSU partially decrypts REC and outputs the transformed ciphertext PTC.
- $\mathsf{abDecryption}_{\mathsf{hel}}(PTC, TK)$: The TEE uses TK to generate the helper ciphertext TC.
- $\mathsf{abDecryption}(PK, TC, UK)$: The DU uses UK and TC to recover msg.

4.3 Security Goals

In our scheme, the CSP and RSU are semi-trusted: they follow the protocol but may infer information from ciphertexts. The TEE is trusted during normal execution, while RSUs may replay outdated states or modify time intervals. Revoked DUs and unauthorized users may also collude with RSUs to recover ciphertexts. We classify these attackers into four types:

1. **Type I Adversary:** This attacker lacks the private key associated with the target ciphertext and tries to distinguish ciphertexts to obtain plaintext information.

2. **Type II Adversary:** This attacker represents collusion between a malicious RSU and unauthorized vehicles or users whose attributes do not satisfy the target access policy, where the RSU attempts to generate or transform ciphertexts and assist the vehicle in decrypting helper ciphertexts.

3. **Type III Adversary:** This attacker is a revoked user attempting to replay stale user state lists and signatures to trick the TEE into decrypting the transformed ciphertext, and then decrypt the helper ciphertext obtained from the TEE.

4. **Type IV Adversary:** This attacker compromises the TEE through hardware side-channel leakage, such as Spectre- or Meltdown-style attacks, and obtains TEE-side states including K_{TEE} and TK, then attempts to recover msg or bypass revocation without a valid user key UK.

4.4 Security Model

To evaluate the security levels of both original and re-encrypted ciphertexts within our framework, we define the security model.

In the context of this model, the system is deemed to satisfy Adaptive IND-CPA security if there exists no adversary $\mathscr{A}$ capable of winning the security game against a challenger $\mathscr{C}$ with a non-negligible advantage. The process proceeds as follows:
Initialization: The challenger $\mathscr{C}$ selects a security parameter $\mathscr{K}$ and runs the $\mathsf{Setup}(1^{\mathscr{K}})$ algorithm to generate the public parameters PK and the master secret key MSK. The adversary $\mathscr{A}$ obtains PK, whereas MSK is kept private by $\mathscr{C}$.
Phase 1: During this phase, $\mathscr{A}$ is allowed to submit the queries listed below within polynomial time constraints:

- $\mathsf{Extract}_{\mathsf{ISK}}(ID)$: $\mathscr{A}$ queries the private key corresponding to an identity ID. $\mathscr{C}$ generates $ISK_{ID} \leftarrow \mathsf{Keygen}_{\mathsf{IBE}}(PK, MSK, ID)$ and sends it to $\mathscr{A}$.

- $\mathsf{Extract}_{\mathsf{ASK}}(S)$: $\mathscr{A}$ queries the private key with an attribute set S. $\mathscr{C}$ generates $ASK_S \leftarrow \mathsf{Keygen}_{\mathsf{ABE}}(PK, MSK, S)$ and sends it to $\mathscr{A}$.

- $\mathsf{Extract}_{\mathsf{RK}_{\mathsf{out}}}(\mathbb{A})$: $\mathscr{A}$ queries the re-encryption transformation key for an access policy $\mathbb{A}$. $\mathscr{C}$ generates $RK_{out} \leftarrow \mathsf{Rekeygen}_{\mathsf{out}}(PK, \mathbb{A})$ and sends it to $\mathscr{A}$.

- $\mathsf{Extract}_{\mathsf{RK}}(ID, \mathbb{A})$: $\mathscr{A}$ queries the re-encryption key for access policy $\mathbb{A}$. $\mathscr{C}$ generates $ISK_{ID} \leftarrow \mathsf{Keygen}_{\mathsf{IBE}}(PK, MSK, ID)$ and $RK_{out} \leftarrow \mathsf{Rekeygen}_{\mathsf{out}}(PK, \mathbb{A})$, then generates $RK \leftarrow \mathsf{Rekeygen}(PK, RK_{out}, \mathsf{ISK}_{ID}, \mathbb{A})$ and sends it to $\mathscr{A}$.

Challenge: $\mathscr{A}$ chooses a user identity ID^* and an access policy $(\mathbb{M}^*, \rho^*)$ of dimensions $l^* \times n^*$, with the following constraints:

Case 1: This case aims to prove the confidentiality of the initial ciphertext before re-encryption. ID^* was not queried to $\mathsf{Extract}_{\mathsf{ISK}}(ID^*)$ during Phase 1. For any re-encryption key query regarding $\mathsf{Extract}_{\mathsf{RK}}(ID^*, (\mathbb{M}, \rho))$ made in Phase 1, the adversary must not have queried $\mathsf{Extract}_{\mathsf{ASK}}(S)$ for any S satisfying the policy $(\mathbb{M}, \rho)$. In this case, adversary $\mathscr{A}$ chooses two plaintext M_0, M_1 of equal length and sends them together with ID^* to $\mathscr{C}$. Subsequently, $\mathscr{C}$ selects a random bit $\kappa \xleftarrow{R} \{0,1\}$ and runs $\mathsf{ibEncryption}(PK, ID^*, M_\kappa)$ to obtain CT^*.

Case 2: This case captures the confidentiality of the re-encrypted ciphertext under the target policy $(\mathbb{M}^*, \rho^*)$. The adversary must not have queried ISK_{ID^*} in Phase 1, nor any attribute key for a set S satisfying $(\mathbb{M}^*, \rho^*)$, but it may query the re-encryption key for ID^* and $(\mathbb{M}^*, \rho^*)$. If these restrictions are violated, $\mathscr{C}$ outputs $\perp$; otherwise, $\mathscr{C}$ samples $\kappa \xleftarrow{R} \{0,1\}$, internally computes ISK^*, CT^*, $RK_{out}{}^*$, and RK^* by running the corresponding algorithms, and returns $REC^* \leftarrow \mathsf{Rencryption}(RK^*, CT^*)$ to $\mathscr{A}$.

Phase 2: As in Phase 1, subject to the challenge constraints.

Guess: $\mathscr{A}$ outputs a guess bit κ^*.

Where the advantage of adversary $\mathscr{A}$ winning this security game is defined as:

$$\mathrm{Adv}_{\mathscr{A}}^{\text{Adaptive-IND-CPA}}(1^{\mathscr{K}}) = \left| \Pr[\kappa^* = \kappa] - \frac{1}{2} \right|, \tag{4}$$

5. SYSTEM CONSTRUCTION

5.1 Construction of the FPRE-OD Scheme

This paper first constructs FPRE-OD, an outsourcing-based fast proxy re-encryption scheme that serves as the cryptographic core of HRFPRE. It provides efficient transmission and a helper-key-based decryption interface later executed inside the TEE. The following algorithms define FPRE-OD and are illustrated in Fig. 2.

1. System Initialization: The KGC runs $\mathsf{Setup}(1^{\mathscr{K}})$ to generate the public parameters PK and master key MSK. The details are shown in Algorithm 1.

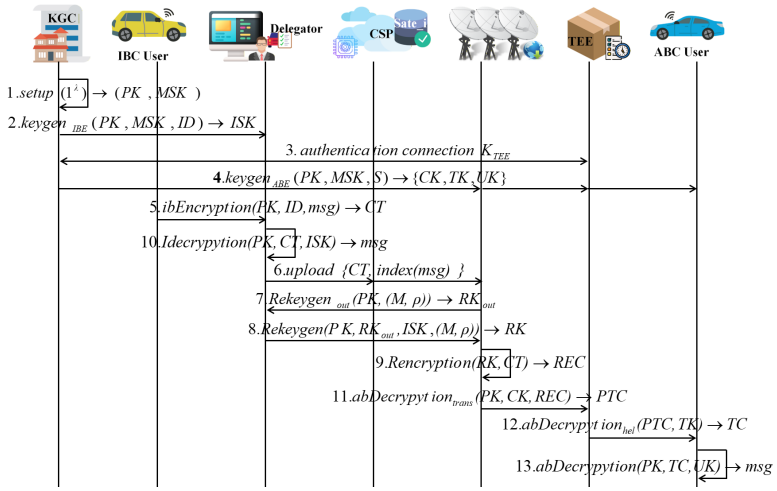


Fig. 2. Detailed workflow of the FPRE-OD scheme.

Algorithm 1 System Initialization: Setup

Input security parameter $\mathscr{K}$
Generate a bilinear pairing $par = (e, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p)$
Choose hash functions $\mathscr{H} : \{0,1\}^* \to \mathbb{G}_1$, $\mathscr{F} : \mathbb{G}_T \to \mathbb{G}_2$
Choose random numbers $b_1, b_2, \alpha \in \mathbb{Z}_p$, $g_1, g_3, g_4 \in \mathbb{G}_1$, $g_2 \in \mathbb{G}_2$
return $PK = (par, \mathscr{H}, \mathscr{F}, g_1, g_2, g_3, g_4, g_2^{b_1}, g_2^{b_2}, e(g_1, g_2)^{\alpha})$, $MSK = (\alpha, b_1, b_2)$

2. Key Generation: IBC users obtain ISK from $\mathsf{Keygen}_{\mathsf{IBE}}(PK, MSK, ID)$, while ABC users obtain $ABK = (CK, TK, UK)$ from $\mathsf{Keygen}_{\mathsf{ABE}}(PK, MSK, S)$, where $CK = (ck_1, ck_2)$, $TK = (t_1, t_2)$, and $UK = \beta$. The details are shown in Algorithms 2 and 3.

Algorithm 2 IBC secret key generation: $\mathsf{Keygen}_{\mathsf{IBE}}$

Input: PK, MSK, ID
Choose random numbers $r \in \mathbb{Z}_p$
Compute $isk_1 = g_2^r, isk_2 = g_1^{\alpha} \cdot g_3^{-r}, isk_3 = \mathscr{H}(0 \parallel ID)^{\frac{r}{b_1}}, isk_4 = \mathscr{H}(1 \parallel ID)^{\frac{r}{b_2}}$
return $ISK = (isk_1, isk_2, isk_3, isk_4)$

Algorithm 3 ABC secret key generation: $\mathsf{Keygen}_{\mathsf{ABE}}$

Input: PK, MSK, S
Choose random numbers $\beta, \eta, t_1, t_2 \in \mathbb{Z}_p$
For $j \in \{1,2\}$ do: Compute $ck_{j,1} = g_2^{\frac{\eta}{\beta t_j}}$, $ck_{j,2} = g_1^{\frac{\alpha}{\beta t_j}} \cdot g_3^{\frac{-\eta}{\beta t_j}}$
For $i \in S$ do: Compute $ck_{j,3,i} = \mathscr{H}(0 \parallel u(i))^{\frac{\eta}{\beta b_1 t_j}}$, $ck_{j,4,i} = \mathscr{H}(1 \parallel u(i))^{\frac{\eta}{\beta b_2 t_j}}$
Set $ck_1 = (ck_{1,1}, ck_{1,2}, ck_{1,3,i}, ck_{1,4,i})$
Set $ck_2 = (ck_{2,1}, ck_{2,2}, ck_{2,3,i}, ck_{2,4,i})$
Set $CK = (ck_1, ck_2)$, $TK = (t_1, t_2)$, $UK = \beta$
return $ABK = (CK, TK, UK)$

3. Re-encryption Outsourcing Key Generation: The DL delegates part of re-

encryption-key generation to the RSU, which runs $\mathsf{Rekeygen}_{\mathsf{out}}(PK,(\mathbb{M},\rho))$ to generate RK_{out}, where $\mathbb{M}$ is an $l \times n$ access matrix and ρ maps each row to an attribute. The details are shown in Algorithm 4.

Algorithm 4 Re-encryption outsourcing key generation: $\mathsf{Rekeygen}_{\mathsf{out}}$

Input: $PK,(\mathbb{M},\rho)$
Choose random numbers $z'_1, z'_2 \in \mathbb{Z}_p$
Set $z' = z'_1 + z'_2$
For $i \in \{1,2,\ldots,l\}$
    Choose random numbers $\lambda'_i \in \mathbb{Z}_p$, Compute $rk'_{1,i} = g_3^{\lambda'_i} \cdot \mathscr{H}(0 \parallel \rho(i))^{z'_1} \cdot \mathscr{H}(1 \parallel \rho(i))^{z'_2}$
Compute $rk'_2 = g_2^{z'}, rk'_3 = g_2^{b_1 z'_1}, rk'_4 = g_2^{b_2 z'_2}$
return $RK_{out} = (rk'_{1,i}, rk'_2, rk'_3, rk'_4, \lambda'_i, z'_1, z'_2, z')$

4. Encryption: The DO runs $\mathsf{ibEncryption}(PK, ID, msg)$ under the DL's identity to generate CT, ensuring that only the trusted DL can decrypt the original ciphertext. The details are shown in Algorithm 5.

Algorithm 5 Original ciphertext encryption: ibEncryption

Input: PK, ID, msg
Choose random numbers $s_1, s_2 \in \mathbb{Z}_p$
set $s = s_1 + s_2$
Compute $ct_1 = \mathscr{H}(0 \parallel ID)^{s_1} \cdot \mathscr{H}(1 \parallel ID)^{s_2} \cdot g_3^s, ct_2 = g_2^s,$
    $ct_3 = g_2^{b_1 s_1}, ct_4 = g_2^{b_2 s_2}, ct_5 = msg \cdot e(g_1, g_2)^{\alpha s}, ct_6 = g_4^s$
return $CT = (ct_1, ct_2, ct_3, ct_4, ct_5, ct_6)$

5. Re-encryption Key Generation: Given ISK, RK_{out}, PK, and $(\mathbb{M},\rho)$, the DL runs Rekeygen to derive the full re-encryption key RK with low local overhead. The details are shown in Algorithm 6.

Algorithm 6 Re-encryption key generation: Rekeygen

Input: $PK,(\mathbb{M},\rho), RK_{out}, ISK$
Choose random numbers $z_1, z_2, \gamma \in \mathbb{Z}_p$, $v' \in \mathbb{Z}_p^{n-1}$
Set $z = z_1 + z_2$, $v = (z \parallel v')$
Compute $(\lambda_1, \lambda_2, \ldots, \lambda_l)^T = \mathbb{M} \cdot v$
Set $rk_1 = isk_1$, $rk_2 = isk_2 \cdot g_4^{\gamma}$, $rk_3 = isk_3$, $rk_4 = isk_4$
For each $i \in \{1,2,\ldots,l\}$ do: set $rk_{5,i} = \lambda_i - \lambda'_i$
Set $rk_6 = z_1 - z'_1$, $rk_7 = z_2 - z'_2$, $rk_8 = z - z'$
Compute $rk_9 = \mathscr{F}(e(g_1,g_2)^{\alpha \cdot z}) \cdot g_2^{\gamma}$
return $RK = (rk_1, rk_2, rk_3, rk_4, rk_{5,i}, rk_6, rk_7, rk_8, rk_9, rk'_{1,i}, rk'_2, rk'_3, rk'_4)_{i \in [1,l]}$

6. Re-encrypted Ciphertext Generation: The RSU runs $\mathsf{Rencryption}(RK, CT)$ to convert the original ciphertext into an access-policy-bound re-encrypted ciphertext REC. The details are shown in Algorithm 7.

7. Original Ciphertext Decryption: In the IBC setting, the trusted DL runs $\mathsf{ibDecryption}(PK, ISK, CT)$ by computing $D_1 = e(isk_3, ct_3) \cdot e(isk_4, ct_4)$ and $D_2 = e(ct_1, isk_1) \cdot e(isk_2, ct_2)$, then recovers $msg = ct_5 \cdot D_1 / D_2$.

8. Re-encrypted Ciphertext Decryption: In the ABC setting, for an authorized attribute set satisfying $(\mathbb{M},\rho)$, constants $\delta_i (i \in I)$ satisfy $\sum_{i \in I} \delta_i \mathbb{M}_i = (1,0,\ldots,0)$. The

Algorithm 7 Re-encrypted ciphertext generation: Rencryption

Input: RK, CT
Compute $rec_1 = ct_5 \cdot e(rk_3, ct_3) \cdot e(rk_4, ct_4) / (e(ct_1, rk_1) \cdot e(rk_2, ct_2))$
Set $rec_{2,i} = rk'_{1,i}$, $rec_3 = rk'_2$, $rec_4 = rk'_3$, $rec_5 = rk'_4$, $rec_{6,i} = rk_{5,i}$, $rec_7 = rk_6$, $rec_8 = rk_7$, $rec_9 = rk_8$, $rec_{10} = rk_9$, $rec_{11} = ct_6$
return $REC = (rec_1, \{rec_{2,i}\}_{i\in[1,l]}, rec_3, rec_4, rec_5, \{rec_{6,i}\}_{i\in[1,l]}, rec_7, rec_8, rec_9, rec_{10}, rec_{11})$

RSU first runs $\mathsf{abDecryption}_{\mathsf{trans}}$ with CK to obtain PTC. The details are shown in Algorithm 8.

Algorithm 8 Transformed ciphertext generation: $\mathsf{abDecryption}_{\mathsf{trans}}$

Input: $PK, CK, REC, I, \{\delta_i\}_{i\in I}$
for $j \in \{1,2\}$ **do**
Compute $P_{j,1} = e\left(\prod_{i\in I}(ck_{j,3,i})^{\delta_i}, rec_4 \cdot g_2^{b_1 rec_7}\right)$, $P_{j,2} = e\left(\prod_{i\in I}(ck_{j,4,i})^{\delta_i}, rec_5 \cdot g_2^{b_2 rec_8}\right)$,
$P_{j,3} = e\left(\prod_{i\in I}\left(rec_{2,i} \cdot g_3^{rec_{6,i}} \cdot \mathscr{H}(0 \parallel \rho(i))^{rec_7} \cdot \mathscr{H}(1 \parallel \rho(i))^{rec_8}\right)^{\delta_i}, ck_{j,1}\right)$,
$P_{j,4} = e(ck_{j,2}, rec_3 \cdot g_2^{rec_9})$
Compute $ptc_1 = (P_{1,1} \cdot P_{1,2}) / (P_{1,3} \cdot P_{1,4})$, $ptc_2 = (P_{2,1} \cdot P_{2,2}) / (P_{2,3} \cdot P_{2,4})$,
return $PTC = (ptc_1, ptc_2)$

The remaining decryption is deliberately split into two independent steps. First, the TEE runs $\mathsf{abDecryption}_{\mathsf{hel}}(PTC, TK)$: it checks $ptc_1^{t_1} \stackrel{?}{=} ptc_2^{t_2}$ and outputs $TC \leftarrow ptc_1^{t_1}$ if valid, otherwise $\perp$. Second, the user runs $\mathsf{abDecryption}(PK, TC, UK)$ by computing $\Delta = TC^{-UK}$ and recovering $msg = rec_1 \cdot e(rec_{11}, rec_{10}/\mathscr{F}(\Delta))$. Therefore, the TEE-held TK alone only produces a helper ciphertext, while plaintext recovery still requires the user-held UK.

5.2 Construction of HRFPRE Scheme

The proposed HRFPRE system is built upon the FPRE-OD model to enable efficient and revocable data sharing. FPRE-OD provides the core encryption, re-encryption, and decryption operations, while a Verkle tree-based signature scheme Π_{sig} [32] and a TEE remote attestation service Π_{TEE} with replay protection together support secure user revocation. The workflow consists of six phases, as illustrated in Table 3.

1. System Initialization: The KGC initiates the system by running the Setup algorithm of $\Pi_{\text{FPRE-OD}}$ to generate global public parameters PK and the master secret key MSK. Simultaneously, the KGC generates signature keys for Π_{sig} and negotiates an authenticated encryption key K_{TEE} via remote attestation. Two core TEE modules are activated (see Table 2):

- *Time Management Module:* Synchronizes clocks with the KGC and maintains the current time interval T_i.
- *Ciphertext Transformation Module:* Verifies transformed ciphertexts PTC and user status, then generates helper ciphertexts TC for authorized users.

Additionally, a Verkle tree is constructed to manage user membership (see Fig. 3). For a user list with 9 vehicles, three vector commitments (V_1, V_2, V_3) and a root commitment V_{root} are generated. The KGC signs V_{root} and publishes the signature σ_i as part of $State_i$.

Table 2. TEE Operational Code Implementation

Pseudocode for TEE Operational Algorithms

Time Management Module $prog_t$

- Execute $sgx_get_time() \to t_0$ (Timestamp t_0), $sgx_set_period \to T$ (Time update period T)
- Store (t_0, T) to RPS
- Output (t_0, T) to KGC

Ciphertext Transformation Module $prog_v$

(i) **Time Verification**

- Download $State_i = (\sigma_i, V_{root_i}, i)$
- $t_{\text{now}} \leftarrow sgx_get_time()$, $i' \leftarrow \lfloor (t_{\text{now}} - t_0)/T \rfloor$ (Time state index i')
- If $i' \neq i$, output $\perp$ and terminate

(ii) **State Verification**

- $v_1 \leftarrow \Pi_{\text{sig}}.\mathsf{verify}(vk, \sigma_i, V_{root_i}, i)$, $v_2 \leftarrow \Pi_{\text{sig}}.\mathsf{Verkle.verify}(V_{root_i}, \pi_{ID}, ID)$
- If $v_1 \neq 1$ or $v_2 \neq 1$, output $\perp$ and terminate

(iii) **Ciphertext Transformation**

- Get $Enc(K_{\text{TEE}}, TK_{ID}) \leftarrow List_i[ID]$
- $TK_{ID} \leftarrow Dec(K_{\text{TEE}}, Enc(K_{\text{TEE}}, TK_{ID}))$, $TC \leftarrow \Pi_{\text{FPRE-OD}}.\mathsf{abDecryption}_{\mathsf{hel}}(PTC, TK_{ID})$
- Output TC

To prove the existence of a user, e.g., U_1, the proof π_{ID_1} consists of the membership proof S_1 linking U_1 to V_1 and the path proof S_{10} linking V_1 to V_{root}. During decryption, the RSU retrieves the proof path to verify v_2 without storing the full user list.

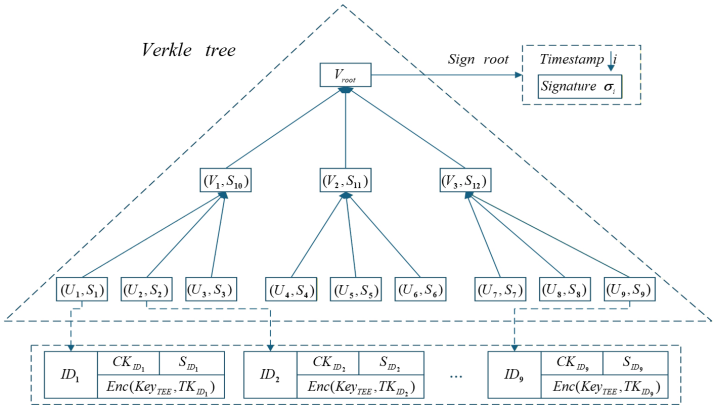


Fig. 3. The construction of the Verkle tree [32].

2. Key Generation and User Registration: In the IBC system, a delegator registers identity ID to obtain ISK, and the KGC broadcasts the trusted delegator list $UL = \{ID_1, \ldots, ID_l\}$. In the ABC system, each user obtains an attribute key UK bound to its attributes.

3. Dynamic State Update: To enable fine-grained revocation, the KGC maintains an authoritative user list $List_i$ synchronized with $prog_t$, containing user identities, attributes, and encrypted transformation keys. It builds a Verkle tree over the updated list, computes V_{root}, and signs it to generate $State_i = (\sigma_i, V_{root_i}, i)$ for CSP-side integrity and freshness

Table 3. HRFPRE Protocol

1. System Initialization.
The KGC establishes a remote-attestation channel with the TEE and negotiates K_{TEE}, starts $prog_t$ and $prog_v$, and synchronizes time via $prog_t$. It runs $\Pi_{\text{FPRE-OD}}.\mathsf{Setup}(1^{\mathscr{K}}) \to (PK_F, MSK_F)$ and $\Pi_{\text{sig}}.\mathsf{keygen}(1^{\mathscr{K}}) \to (vk, sk)$, setting $PK = (PK_F, vk)$ and $MSK = (MSK_F, sk, K_{\text{TEE}})$. It initializes $i = 0$, $List_0 = \varnothing$, $V_{root_0} = 0$, signs V_{root_0} to obtain σ_0, sets $State_0 = (\sigma_0, V_{root_0}, 0)$, outputs $(PK, MSK, State_0, List_0)$.

2. Key Generation.
The KGC runs $prog_t$ to synchronize the time i at both ends.
- **IBC User Registration**: The KGC runs $\Pi_{\text{FPRE-OD}}.\mathsf{Keygen}_{\mathsf{IBE}}(PK_F, MSK_F, ID) \to ISK$ and returns it to the registered user.
- **ABC User Registration**: The KGC runs $\mathsf{Keygen}_{\mathsf{ABE}}(PK, MSK, S) \to \{CK, TK, UK\}$ and returns UK to the user.

3. State Update.
- **User Registration**: When a new user registers, the KGC adds the tuple $\{ID, S_{ID}, CK_{ID}, Enc(K_{\text{TEE}}, TK_{ID})\}_{ID \in U}$ to $List_{i-1}$.
- **User Revocation**: For a malicious user ID_p in the ABC system, the KGC deletes their identity information tuple from $List_{i-1}$.
- **List Update**: The KGC maintains $UL = \{ID_1, ID_2, \ldots, ID_l\}$ as the trusted delegator list. After registration or revocation, it updates the authorized ABC-user list $List_i$ and constructs a Verkle tree to derive the root commitment V_{root_i} via intermediate vector commitments $(V_1, V_2, \ldots)$. It generates a signature $\Pi_{\text{sig}}.\mathsf{sig}(sk, V_{root_i}, i) \to \sigma_i$. The updated user State $State_i = (\sigma_i, V_{root_i}, i)$ is published. $List_i$ and the corresponding proofs $\pi_{ID_j} = (S_j, S_{group})$ for users in $List_i$ are uploaded to the CSP, where S_j is the leaf proof and S_{path_j} corresponds to S_{10}, S_{11}, or S_{12} in Fig. 3.

4. Encryption.
The vehicle selects a trusted delegator from UL and runs $\Pi_{\text{FPRE-OD}}.\mathsf{ibEncryption}(PK, ID, msg) \to CT$. It also specifies the access policy $(\mathbb{M}, \rho)$ and index $index(msg)$, sending them along with the ciphertext to the DL.

5. Transmission.
The RSU runs $\Pi_{\text{FPRE-OD}}.\mathsf{Rekeygen}_{\mathsf{out}}(PK, (\mathbb{M}, \rho)) \to RK_{\text{out}}$ to execute the re-encryption key outsourcing work. The DL runs $\Pi_{\text{FPRE-OD}}.\mathsf{Rekeygen}(PK, RK_{\text{out}}, ISK, (\mathbb{M}, \rho)) \to RK$ and sends it to the RSU. The RSU runs $\Pi_{\text{FPRE-OD}}.\mathsf{Rencryption}(RK, CT) \to REC$ to generate the ciphertext and store it for a short term.

6. Decryption & Revocation.
- **IBC User Decryption**: The DL runs $\Pi_{\text{FPRE-OD}}.\mathsf{ibDecryption}(PK, CT, ISK)$ to obtain the msg.
- **ABC User Decryption and Revocation**: The ABC user applies to access data of interest based on $index(msg)$.
  i. **Transformed Ciphertext Generation**: The RSU downloads $State_i$ from the state module and obtains the target user's proof path $\pi_{ID} = (S_{leaf}, S_{path})$ from the CSP (e.g., S_1, S_{10} for user U_1). It verifies the state signature $\Pi_{\text{sig}}.\mathsf{verify}(vk, \sigma_i, V_{root_i}, i) \to v_1$ and the Verkle membership proof $\Pi_{\text{sig}}.\mathsf{Verkle.verify}(V_{root_i}, \pi_{ID}, ID) \to v_2$. If $v_1 = v_2 = 1$, the RSU uses CK to run $\Pi_{\text{FPRE-OD}}.\mathsf{abDecryption}_{\mathsf{trans}}(PK, CK, REC) \to PTC$.
  ii. **Helper Ciphertext Generation**: After verifying V_{root_i} and the target ID commitment, the TEE runs $prog_v(PTC, State_i, \pi_{ID}, ID) \to TC$ for the authorized vehicle.
  iii. **User Decryption**: The vehicle user runs $\Pi_{\text{FPRE-OD}}.\mathsf{abDecryption}(PK, TC, UK) \to msg$ to obtain the original information.

verification. Since $State_i$ comes from the latest list, only users included in the latest signed $List_i$ can provide membership proofs accepted for the current epoch, and RSUs only need the compact state.

4. Secure Information Transmission: A sender selects a target delegator from UL and uses $\Pi_{\text{FPRE-OD}}$ to encrypt data under access policy $(\mathbb{M},\rho)$, generating CT. The delegator and RSU then generate RK_{out} and re-encrypt CT into a shareable ciphertext.

5. Decryption and Real-time Revocation: When a vehicle requests data access, the RSU verifies the latest $State_i$ and the user's Verkle proof without storing the full list. If the requester remains in $List_i$, it runs $\Pi_{\text{FPRE-OD}}$ to generate PTC. The TEE module $prog_v$ then verifies PTC against the internal clock and user status, decrypts the helper key with K_{TEE}, and generates TC, preventing compromised RSUs from serving revoked users. Finally, the authorized vehicle uses UK to decrypt TC and recover the message.

6. SECURITY ANALYSIS

The security of the HRFPRE scheme presented in this study is validated through a sequence of interactive games modeled as a game between a challenger and an adversary. Given that this proposal functions as an IBC-ABC proxy re-encryption mechanism, it is essential to independently demonstrate indistinguishability regarding both the original ciphertext and the re-encrypted version. The formal security proof for our scheme is detailed below.

6.1 Security Proof

Theorem 1: In the random oracle model, if the DLIN assumption holds, then the basic scheme $\Pi_{\text{FPRE-OD}}$ proposed in this paper satisfies Adaptive Indistinguishability under Chosen Plaintext Attack (Adaptive IND-CPA). That is, for any polynomial-time adversary $\mathscr{A}$ that breaks the Adaptive IND-CPA security of $\Pi_{\text{FPRE-OD}}$ with non-negligible advantage $\text{Adv}^{\text{Adaptive-IND-CPA}}_{\Pi_{\text{FPRE-OD}},\mathscr{A}}$, then there exists an algorithm $\mathscr{B}$ that solves the DLIN problem with non-negligible advantage.

Proof: Assume there exists a polynomial-time adversary $\mathscr{A}$ capable of breaking the adaptive IND-CPA security with a non-negligible advantage ε. We can then derive an algorithm $\mathscr{B}$ designed to solve the hard problem underlying the DLIN assumption with an advantage of $\varepsilon/2$. First, algorithm $\mathscr{B}$ randomly chooses $b_1,b_2,s_1,s_2 \in \mathbb{Z}_p$, $g_1 \in \mathbb{G}_1$, $g_2 \in \mathbb{G}_2$, and sets $s = s_1 + s_2$, $R \xleftarrow{R} \mathbb{Z}_p$. At this point, define a DLIN tuple:

$$\begin{aligned} &par = (e,\mathbb{G}_1,\mathbb{G}_2,\mathbb{G}_T,p) \leftarrow \mathsf{GroupGen}(1^{\mathscr{K}}),\\ &(C,D,E,F,G,H) = (g_1,g_2,g_1^{b_1},g_1^{b_2},g_2^{b_1},g_2^{b_2}), \quad (I,J,K,L) = (g_1^{b_1 s_1},g_1^{b_2 s_2},g_2^{b_1 s_1},g_2^{b_2 s_2}). \end{aligned} \tag{5}$$

The challenger $\mathscr{C}$ randomly flips a binary coin κ. When $\kappa = 0$, challenger $\mathscr{C}$ sets $A = g_1^s, B = g_2^s$; conversely, when $\kappa = 1$, challenger $\mathscr{C}$ sets $A = g_1^R, B = g_2^R$. The interactive game between $\mathscr{A}$ and $\mathscr{B}$ proceeds as follows:

Initialization: $\mathscr{B}$ runs the $\mathsf{Setup}(\cdot)$ algorithm, randomly chooses $\alpha,c,d \in \mathbb{Z}_p$, and generates $MSK = \alpha$. $\mathscr{B}$ sends the following PK to $\mathscr{A}$ while keeping MSK private:

$$PK = (par,\mathscr{H},\mathscr{F},C,D,C^c,C^d,G,H,e(C,D)^{\alpha}). \tag{6}$$

Phase 1: The adversary $\mathscr{A}$ initiates queries to the random oracles defined below, with $\mathscr{B}$ providing the corresponding simulations:

- $\mathscr{H}(\cdot)$: $\mathscr{B}$ maintains two $\mathscr{H}(\cdot)$ query lists $\mathscr{L}_1, \mathscr{L}_2$, with entries $\{0 \parallel \tau_i, \mu_i, h_{(0\parallel\tau_i)}\}$ and $\{1 \parallel \tau_i, \nu_i, h_{(1\parallel\tau_i)}\}$ respectively. $\mathscr{B}$ simulates the following two types of random oracle queries:

  Query $0 \parallel \tau_i$: $\mathscr{B}$ first checks $\mathscr{L}_1$ for an entry containing $0 \parallel \tau_i$. If it exists, it selects $\mathscr{H}(0 \parallel \tau_i) = h_{(0\parallel\tau_i)}$ from $\{0 \parallel \tau_i, \mu_i, h_{(0\parallel\tau_i)}\}$ as the response. Otherwise, $\mathscr{B}$ randomly chooses $\mu_i \in \mathbb{Z}_p$, computes $h_{(0\parallel\tau_i)} = E^{\mu_i}$, stores the new tuple $\{0 \parallel \tau_i, \mu_i, h_{(0\parallel\tau_i)}\}$ in $\mathscr{L}_1$, and returns $\mathscr{H}(0 \parallel \tau_i) = E^{\mu_i}$ as the response.

  Query $1 \parallel \tau_i$: $\mathscr{B}$ first checks $\mathscr{L}_2$ for an entry containing $1 \parallel \tau_i$. If it exists, it selects $\mathscr{H}(1 \parallel \tau_i) = h_{(1\parallel\tau_i)}$ from $\{1 \parallel \tau_i, \nu_i, h_{(1\parallel\tau_i)}\}$ as the response. Otherwise, $\mathscr{B}$ randomly chooses $\nu_i \in \mathbb{Z}_p$, computes $h_{(1\parallel\tau_i)} = F^{\nu_i}$, stores the new tuple $\{1 \parallel \tau_i, \nu_i, h_{(1\parallel\tau_i)}\}$ in $\mathscr{L}_2$, and returns $\mathscr{H}(1 \parallel \tau_i) = F^{\nu_i}$ as the response.

- $\mathsf{Extract}_{\mathsf{ISK}}(ID)$: $\mathscr{A}$ queries the private key for identity ID_i. $\mathscr{B}$ checks $\mathscr{L}_{ISK}$ for an entry containing (ISK, ID_i). If it exists, it returns ISK. Otherwise, $\mathscr{B}$ runs $\mathscr{H}(\cdot)$ queries to obtain the corresponding tuples for ID_i in lists $\mathscr{L}_1, \mathscr{L}_2$. Let $\mathscr{H}(0 \parallel ID_i) = h_{(0\parallel ID_i)}, \mathscr{H}(1 \parallel ID_i) = h_{(1\parallel ID_i)}$. Subsequently, $\mathscr{B}$ randomly chooses $r \in \mathbb{Z}_p$ and simulates the identity key ISK as follows:

$$isk_1 = D^r, \quad isk_2 = C^{\alpha - cr}, \quad isk_3 = C^{\mu_i r}, \quad isk_4 = C^{\nu_i r}. \tag{7}$$

  where $isk_3 = \mathscr{H}(0 \parallel ID_i)^{r/b_1} = E^{\mu_i r/b_1} = C^{\mu_i r}$ and $isk_4 = \mathscr{H}(1 \parallel ID_i)^{r/b_2} = F^{\nu_i r/b_2} = C^{\nu_i r}$, such that b_1, b_2 are cancelled out. $\mathscr{B}$ adds (ISK, ID_i) to $\mathscr{L}_{ISK}$ and returns identity key ISK to $\mathscr{A}$.

- $\mathsf{Extract}_{\mathsf{ASK}}(S)$: $\mathscr{A}$ queries for each attribute u_i in the private key for attribute set S. $\mathscr{B}$ checks $\mathscr{L}_{ASK}$ for an entry containing $(ASK = (CK, TK, UK), S)$. If it exists, it returns ASK. Otherwise, $\mathscr{B}$ runs $\mathscr{H}(\cdot)$ queries to obtain the corresponding tuples for u_i in lists $\mathscr{L}_1, \mathscr{L}_2$. Let $\mathscr{H}(0 \parallel u_i) = h_{(0\parallel u_i)}, \mathscr{H}(1 \parallel u_i) = h_{(1\parallel u_i)}$. Subsequently, $\mathscr{B}$ randomly chooses $\beta, t_1, t_2, \eta \in \mathbb{Z}_p$ and simulates the attribute key $\mathrm{ASK} = (CK, TK, UK)$. For $j \in \{1, 2\}$:

$$ck_{j,1} = D^{\eta/(\beta t_j)}, ck_{j,2} = C^{(\alpha - c\eta)/(\beta t_j)}, ck_{j,3,i} = C^{\mu_i \eta/(\beta t_j)}, \quad ck_{j,4,i} = C^{\nu_i \eta/(\beta t_j)}. \tag{8}$$

  where

$$\begin{aligned} ck_{j,3,i} &= \mathscr{H}(0 \parallel u_i)^{\eta/(b_1 t_j \beta)} = E^{\mu_i \eta/(\beta b_1 t_j)} = C^{\mu_i \eta/(\beta t_j)}, \\ ck_{j,4,i} &= \mathscr{H}(1 \parallel u_i)^{\eta/(b_2 t_j \beta)} = F^{\nu_i \eta/(\beta b_2 t_j)} = C^{\nu_i \eta/(\beta t_j)}, \end{aligned} \tag{9}$$

  At this point, let $ck_1 = (ck_{1,1}, ck_{1,2}, ck_{1,3,i}, ck_{1,4,i})$, $ck_2 = (ck_{2,1}, ck_{2,2}, ck_{2,3,i}, ck_{2,4,i})$, simulating a properly distributed attribute key ASK. $\mathscr{B}$ adds $(ASK = (CK, TK, UK), S)$ to $\mathscr{L}_{ASK}$ and returns ASK to $\mathscr{A}$.

- $\mathsf{Extract}_{\mathsf{RK_{out}}}(\mathbb{M}, \rho)$: $\mathscr{A}$ queries the re-encryption transformation key associated with the access structure $(\mathbb{M}, \rho)$. $\mathscr{B}$ checks $\mathscr{L}_{RK_{out}}$ for an entry containing $(RK_{out}, (\mathbb{M}, \rho))$. If it exists, it returns RK_{out}. Otherwise, for $\rho(i)_{i \in \{1,2,\ldots,l\}}$, $\mathscr{B}$ randomly chooses $\lambda_i', z_1', z_2' \in \mathbb{Z}_p$, sets $z' = z_1' + z_2'$, runs $\mathscr{H}(\cdot)$ queries, and retrieves

the corresponding tuples in lists $\mathscr{L}_1, \mathscr{L}_2$. It simulates the re-encryption outsourcing key RK_{out} as:

$$rk'_{1,i} = C^{c\lambda'_i} E^{\mu_i z'_1} F^{v_i z'_2}, \quad rk'_2 = D^{z'}, \quad rk'_3 = G^{z'_1}, \quad rk'_4 = H^{z'_2}. \tag{10}$$

where $rk'_{1,i} = g_3^{\lambda'_i} \mathscr{H}(0 \parallel \rho_i)^{z'_1} \mathscr{H}(1 \parallel \rho_i)^{z'_2} = C^{c\lambda'_i} E^{\mu_i z'_1} F^{v_i z'_2}$. $\mathscr{B}$ returns RK_{out} to $\mathscr{A}$ and adds it to $\mathscr{L}_{RK_{out}}$.

- $\mathsf{Extract}_{\mathsf{RK}}(ID, (\mathbb{M}, \rho), RK_{out})$: $\mathscr{A}$ queries the re-encryption key for access policy $\mathbb{A}$. $\mathscr{B}$ checks $\mathscr{L}_{RK}$ for an entry containing $(ID, (\mathbb{M}, \rho))$. If it exists, it returns RK. Otherwise, $ISK_{ID} \leftarrow \mathsf{Extract}_{\mathsf{ISK}}(ID)$ and $RK_{out} \leftarrow \mathsf{Extract}_{\mathsf{RK_{out}}}(\mathbb{M}, \rho)$. Finally, it generates $RK \leftarrow \mathsf{Rekeygen}(PK, RK_{out}, ISK_{ID}, (\mathbb{M}, \rho))$, adds it to $\mathscr{L}_{RK}$, and returns RK to $\mathscr{A}$.

Challenge: The adversary $\mathscr{A}$ first selects a target identity ID^* and an access policy $(\mathbb{M}^*, \rho^*)$, and sends them to $\mathscr{B}$. The constraints are as follows:

- **Case 1:** ID^* was not queried to $\mathsf{Extract}_{\mathsf{ISK}}(ID^*)$ during Phase 1. In addition, for any re-encryption-key query involving ID^* and an access policy $(\mathbb{M}, \rho)$, the adversary must not have queried $\mathsf{Extract}_{\mathsf{ASK}}(S)$ for any S satisfying $(\mathbb{M}, \rho)$. At this point, adversary $\mathscr{A}$ chooses two plaintext messages M_0, M_1 of equal length and sends them together with ID^* to $\mathscr{B}$. $\mathscr{B}$ randomly selects $\sigma \in \{0,1\}$, runs $\mathscr{H}(\cdot)$ queries to obtain $h_{(0\|ID^*)}, h_{(1\|ID^*)}$. $\mathscr{B}$ simulates the original ciphertext CT' as follows:

$$ct_1 = I^{\mu'} J^{v'} A^c, ct_2 = B, ct_3 = K, ct_4 = L, ct_5 = M_\sigma e(A, D)^\alpha, ct_6 = A^d. \tag{11}$$

where $ct_1 = \mathscr{H}(0 \parallel ID^*)^{s_1} \cdot \mathscr{H}(1 \parallel ID^*)^{s_2} \cdot g_3^s = E^{\mu' s_1} F^{v' s_2} C^{cs} = I^{\mu'} J^{v'} A^c$.

when $\kappa = 0$, $A = {g_1}^s, B = {g_2}^s$: $ct_1 = I^{\mu'} J^{v'} {g_1}^{cs}, ct_5 = M_\sigma e(g_1, g_2)^{\alpha s}, ct_6 = {g_1}^{ds}$.
when $\kappa = 1$, $A = {g_1}^R, B = {g_2}^R$: $ct_1 = I^{\mu'} J^{v'} {g_1}^{cR}, ct_5 = M_\sigma e(g_1, g_2)^{\alpha R}, ct_6 = {g_1}^{dR}$.

- **Case 2:** The private key of identity ID^* (ISK_{ID^*}) was not queried in Phase 1; no attribute set S that satisfies $(\mathbb{M}^*, \rho^*)$ was queried to $\mathsf{Extract}_{\mathsf{ASK}}(S)$ in Phase 1; $\mathscr{A}$ is allowed to query the re-encryption key corresponding to ID^* and $(\mathbb{M}^*, \rho^*)$.

  Adversary $\mathscr{A}$ sends (M_0, M_1) and ID^* to $\mathscr{B}$. $\mathscr{B}$ randomly selects $\sigma \in \{0,1\}$ and outputs CT^* for ID^* as in Case 1. Then $\mathscr{B}$ directly constructs ${ISK^*}, {RK_{out}}^*, {RK^*}$. At this point, $\mathscr{B}$ runs $\mathsf{Rencryption}$ (RK^*, CT^*) to obtain the re-encrypted ciphertext as follows (for $i \in \{1, 2, \ldots, l^*\}$):

$$\begin{aligned} &rec_1 = M_\sigma / e(C, B)^{d\gamma}, \quad rec_{2,i} = C^{c\lambda_i} E^{\mu^*_i z'_1} F^{v^*_i z'_2}, \quad rec_3 = D^{z'}, \\ &rec_4 = G^{{z_1}'}, \quad rec_5 = H^{{z_2}'}, \quad rec_{6,i} = rk^*_{5,i}, \quad rec_7 = rk^*_6, \\ &rec_8 = rk^*_7, \quad rec_9 = rk^*_8, \quad rec_{10} = rk^*_9, \quad rec_{11} = A^d. \end{aligned} \tag{12}$$

When $\kappa = 0, A = {g_1}^s, B = {g_2}^s$: $rec_1 = M_\kappa / e(C, g_2^s)^{d\gamma}$, $rec_{11} = {g_1}^{sd}$.
When $\kappa = 1, A = {g_1}^R, B = {g_2}^R$: $rec_1 = M_\kappa / e(C, g_2^R)^{d\gamma}$, $rec_{11} = {g_1}^{Rd}$.

Phase 2: Consistent with Phase 1, while satisfying the constraints of both cases in the Challenge.

Guess: $\mathscr{A}$ outputs a bit $\sigma' \in \{0,1\}$. When $\sigma = \sigma'$, $\mathscr{B}$ outputs $\kappa' = 0$, indicating (A,B) is a DLIN tuple. Otherwise, $\mathscr{B}$ outputs $\kappa' = 1$, indicating (A,B) are random tuples.

When $\kappa = 1$, A,B are uniformly distributed, satisfying $\Pr[\sigma = \sigma' \mid \kappa = 1] = \frac{1}{2}$. Since $\mathscr{B}$ guesses $\kappa = 1$ when $\sigma \neq \sigma'$, we have $\Pr[\kappa' = \kappa \mid \kappa = 1] = \frac{1}{2}$.

When $\kappa = 0$, defining the adversary's advantage as ε, we satisfy $\Pr[\sigma = \sigma' \mid \kappa = 0] = \frac{1}{2} + \varepsilon$. Since $\mathscr{B}$ guesses $\kappa = 0$ when $\sigma = \sigma'$, we have $\Pr[\kappa' = \kappa \mid \kappa = 0] = \frac{1}{2} + \varepsilon$.

$\mathscr{B}$'s advantage in solving the DLIN problem is:

$$\begin{aligned} \mathrm{Adv}^{\text{Adaptive-IND-CPA}}_{\text{HRFPRE},\mathscr{B}}(1^{\mathscr{K}}) &= \frac{1}{2}\Pr[\kappa' = \kappa \mid \kappa = 0] + \frac{1}{2}\Pr[\kappa' = \kappa \mid \kappa = 1] - \frac{1}{2} \\ &= \frac{1}{2}\left(\frac{1}{2} + \varepsilon\right) + \frac{1}{2}\cdot\frac{1}{2} - \frac{1}{2} = \frac{\varepsilon}{2}. \end{aligned} \tag{13}$$

Theorem 2: If the basic scheme $\Pi_{\text{FPRE-OD}}$ satisfies Adaptive IND-CPA security, the TEE remote attestation scheme Π_{TEE} satisfies IND-CCA security, and the user list signature scheme Π_{sig} satisfies EUF-CMA security, then the HRFPRE scheme constructed in this paper achieves adaptive security under the DLIN assumption. Specifically, for any polynomial-time adversary $\mathscr{A}$, attempting to attack the HRFPRE scheme, the following relationship holds:

$$\mathrm{Adv}^{\text{Adaptive}}_{\text{HRFPRE},\mathscr{A}} \leq \mathrm{Adv}^{\text{Adaptive-IND-CPA}}_{\Pi_{\text{FPRE-OD}},\mathscr{B}'} + \mathrm{Adv}^{\text{IND-CCA}}_{\Pi_{\text{TEE}},\mathscr{T}} + \mathrm{Adv}^{\text{EUF-CMA}}_{\Pi_{\text{sig}},\mathscr{F}} \tag{14}$$

Where $\mathscr{B}'$, $\mathscr{T}$, and $\mathscr{F}$ represent the adversaries constructed against the respective underlying schemes, with running times polynomially related to that of adversary $\mathscr{A}$. This theorem considers the normal execution setting without TEE-side key leakage; the Type IV leakage case is analyzed separately below, where confidentiality relies on the user-held UK and revocation damage is bounded by the current epoch.

Proof: The proof of this theorem is primarily based on the hybrid argument model from game theory. By constructing the indistinguishability of GAME_0–GAME_2, this implies that the advantage of adversary $\mathscr{A}$ in breaking the original HRFPRE security experiment is equivalent to the advantage in distinguishing between GAME_0–GAME_2. Let E_i denote the event that the adversary correctly guesses the challenge bit in GAME_i, and define the advantage as $\text{Adv}_i := \left|\Pr[E_i] - \frac{1}{2}\right|$. Finally, we provide the upper bound for $\mathrm{Adv}^{\text{Adaptive}}_{\text{HRFPRE},\mathscr{A}}$.

The definitions of the games are as follows:

- **GAME_0**: This game is identical to the real security experiment of the HRFPRE scheme.
- **GAME_1**: This game is identical to GAME_0, unless the adversary provides a user list signature $State_i^* = (\sigma_i^*, V_{root_i}^*, i^*)$ that was not generated by the challenger. In this case, the challenger outputs $\perp$ and terminates the game.
- **GAME_2**: Identical to GAME_1, except that in the authentication communication between the TEE and the KGC, $Enc(Key_{TEE}, TK)$ is replaced with a random ciphertext of equal length. If the adversary attempts to distinguish these ciphertexts via the

decryption/authentication interface, the IND-CCA rules are followed, specifically that the challenge ciphertext cannot be submitted to the decryption interface.

Claim 0: If Π_{sig} satisfies EUF-CMA security, then the advantage of the adversary in distinguishing between GAME_0 and GAME_1 is negligible.

Proof: From the adversary's perspective, GAME_0 and GAME_1 are indistinguishable unless the adversary can forge a correctly distributed user list signature $State_i^* = (\sigma_i^*, V_{root_i}^*, i^*)$ that passes verification. In this case, there exists a forger $\mathscr{F}$ such that the difference in winning GAME_0 and GAME_1 is:

$$|\Pr[E_0] - \Pr[E_1]| \leq \text{Adv}_{\Pi_{\text{sig}},\mathscr{F}}^{\text{EUF-CMA}}. \tag{15}$$

Claim 1: If Π_{TEE} satisfies IND-CCA security, then the advantage of the adversary in distinguishing between GAME_1 and GAME_2 is negligible.

Proof: From the adversary's perspective, GAME_1 and GAME_2 are indistinguishable unless the adversary can distinguish between the random ciphertext and the real ciphertext in the authenticated encryption scheme between the TEE and KGC. In this case, there exists an adversary $\mathscr{T}$ such that the difference in winning GAME_1 and GAME_2 is:

$$|\Pr[E_1] - \Pr[E_2]| \leq \text{Adv}_{\Pi_{\text{TEE}},\mathscr{T}}^{\text{IND-CCA}}. \tag{16}$$

Claim 2: If there exists an adversary who can obtain a non-negligible advantage in GAME_2, then it is possible to construct a simulator $\mathscr{B}'$ capable of breaking the adaptive IND-CPA security of the basic scheme $\Pi_{\text{FPRE-OD}}$ with a non-negligible advantage.

Proof: Assume there exists an adversary $\mathscr{A}$ that can win GAME_2 with a non-negligible advantage. We can construct a simulator $\mathscr{B}'$ to win the adaptively IND-CPA game for the FPRE-OD scheme.

Initialization: $\mathscr{C}'$ performs the initialization of $\Pi_{\text{FPRE-OD}}$ to generate the public parameters PK' and master secret key MSK', and sends PK' to $\mathscr{B}'$. $\mathscr{B}'$ initializes an empty user list $List$, runs a subroutine of the simulated signature scheme $\Pi_{\text{sig}}.\mathsf{keygen}(1^{\mathscr{K}})$ to generate (vk', sk'), and generates a set of random symmetric keys K_{TEE}. $\mathscr{B}'$ queries the empty $list$ to generate the signature $\sigma_0^{'}$. At this point, $\mathscr{B}'$ sends $PK', vk', \sigma_0^{'}$ to $\mathscr{A}$, while keeping MSK', sk' private.

Phase 1: $\mathscr{A}$ makes polynomial queries to the following random oracles. $\mathscr{B}'$ maintains lists $\mathscr{L}_{user}, \mathscr{L}_{sign}, \mathscr{L}_{ISK}, \mathscr{L}_{ASK}, \mathscr{L}_{RK_{out}}, \mathscr{L}_{RK}$ and responds as follows:

- $\mathsf{Query}_{\mathsf{JOIN}}(ID, S)$: $\mathscr{A}$ submits an identity ID and the corresponding attribute set S_{ID} to $\mathscr{B}'$ to apply for registration. $\mathscr{B}'$ checks if a tuple $\{ID, S_{ID}, ISK_{ID}, CK_{ID}, \mathsf{Enc}(K_{\text{TEE}}, TK_{ID})\}$ exists in $\mathscr{L}_{user}$. If it exists, $\mathscr{B}'$ returns the tuple to $\mathscr{A}$. Otherwise, $\mathscr{B}'$ interacts with the $\Pi_{\text{FPRE-OD}}$ game to obtain the corresponding CK and TK, adds the new tuple to $\mathscr{L}_{user}$, and then returns it to $\mathscr{A}$.

- $\mathsf{Query}_{\mathsf{REVOCATION}}(ID, S)$: $\mathscr{A}$ queries for the user list after revoking identity ID. $\mathscr{B}'$ removes the tuple containing identity ID (i.e., $\{ID, S_{ID}, ISK_{ID}, CK_{ID}, \mathsf{Enc}(K_{\text{TEE}}, TK_{ID})\}$) from the original identity list, generates a new user list $\mathscr{L}_{user}$, and returns it to $\mathscr{A}$.

- $\mathsf{Query}_{\mathsf{SIG}}(i)$: $\mathscr{A}$ queries for the state signature at timestamp i. $\mathscr{B}'$ checks the list $\mathscr{L}_{sign}$ for an existing tuple. If found, it returns the stored signature. Otherwise, $\mathscr{B}'$ retrieves the current user list $\mathscr{L}_{user}$ and constructs the Verkle tree to compute the root commitment V_{root_i}. Then, $\mathscr{B}'$ signs (V_{root_i}, i) using sk' to obtain σ_i. Finally, $\mathscr{B}'$ records $State_i = (\sigma_i, V_{root_i}, i)$ in $\mathscr{L}_{sign}$ and returns it to $\mathscr{A}$.

- $\mathsf{Query}_{\mathsf{ISK}}(ID)$: $\mathscr{A}$ queries for the identity secret key ISK of identity ID. $\mathscr{B}'$ searches for the $\{ID, ISK\}$ tuple in $\mathscr{L}_{ISK}$. If found, it returns the key. Otherwise, $\mathscr{B}'$ forwards the query to $\mathscr{C}'$. $\mathscr{C}'$ runs $\mathsf{Extract}_{\mathsf{ISK}}(ID)$ in the $\Pi_{\text{FPRE-OD}}$ game and returns ISK to $\mathscr{B}'$. $\mathscr{B}'$ adds $\{ID, ISK\}$ to $\mathscr{L}_{ISK}$ and returns it to $\mathscr{A}$.

- $\mathsf{Query}_{\mathsf{ASK}}(S)$: $\mathscr{A}$ queries for the attribute private key $(ASK = (CK, TK, UK), S)$ corresponding to attribute set S. $\mathscr{B}'$ searches for the $\{S, ASK\}$ tuple in $\mathscr{L}_{ASK}$. If found, it returns the key. Otherwise, $\mathscr{B}'$ forwards the query to $\mathscr{C}'$. $\mathscr{C}'$ runs $\mathsf{Extract}_{\mathsf{ASK}}(S)$ in the $\Pi_{\text{FPRE-OD}}$ game and returns $\{S, ASK\}$ to $\mathscr{B}'$. $\mathscr{B}'$ adds it to $\mathscr{L}_{ASK}$ and returns ASK to $\mathscr{A}$.

- $\mathsf{Query}_{\mathsf{RK_{out}}}(\mathbb{A})$: $\mathscr{A}$ queries for the re-encryption outsourcing key RK_{out} for access policy $\mathbb{A}$. $\mathscr{B}'$ checks $\mathscr{L}_{RK_{out}}$. If found, it returns the key. Otherwise, $\mathscr{B}'$ forwards the query to $\mathscr{C}'$. Upon receiving RK_{out} from $\mathscr{C}'$, $\mathscr{B}'$ returns it to the adversary and adds $\{\mathbb{A}, RK_{out}\}$ to $\mathscr{L}_{RK_{out}}$.

- $\mathsf{Query}_{\mathsf{RK}}(PK, ID, \mathbb{A})$: $\mathscr{A}$ queries for the re-encryption key RK corresponding to access policy $\mathbb{A}$ and the ciphertext CT encrypted under ID. $\mathscr{B}'$ searches for the key in $\mathscr{L}_{RK}$. If found, it returns the key. Otherwise, $\mathscr{B}'$ forwards the query to $\mathscr{C}'$. After obtaining RK, $\mathscr{B}'$ adds $\{ID, \mathbb{A}, RK\}$ to $\mathscr{L}_{RK}$ and returns it to $\mathscr{A}$.

Challenge: $\mathscr{A}$ first chooses an identity identifier ID' and an access policy $(\mathbb{M}', \rho')$, and sends them to $\mathscr{B}'$. Here, $\mathbb{M}'$ is an access matrix with l' rows and n' columns. The following constraints must be satisfied:

- **Case 1:** ID' was not queried in $\mathsf{Extract}_{\mathsf{ISK}}(ID')$ during Phase 1. Simultaneously, for the target identity ID', no access policy $(\mathbb{M}', \rho') = \mathbb{A}$ was queried in $\mathsf{Query}_{\mathsf{RK}}(PK, ID, (\mathbb{A}))$. At this point, adversary $\mathscr{A}$ chooses two plaintext messages M_0, M_1 of equal length and sends them together with ID' to $\mathscr{B}'$. $\mathscr{B}'$ sends them to $\mathscr{C}'$ to obtain the ciphertext CT and transmits it to $\mathscr{A}$.

- **Case 2:** ID' and $(\mathbb{M}', \rho')$ were queried in Phase 1, but no attribute set S satisfying the access policy $(\mathbb{M}', \rho')$ was queried in $\mathsf{Query}_{\mathsf{ASK}}(S)$ during Phase 1. At this point, adversary $\mathscr{A}$ chooses two plaintext messages M_0, M_1 of equal length and sends them together with ID' to $\mathscr{B}'$. $\mathscr{B}'$ sends them to $\mathscr{C}'$ to obtain the ciphertext REC and transmits it to $\mathscr{A}$.

Phase 2: Consistent with Phase 1, while satisfying the constraints of both cases in the Challenge.

Guess: Upon receiving the bit κ^* from $\mathscr{A}$, $\mathscr{B}'$ outputs the same bit κ^* as its guess.

Since in GAME_2: (i) Signature forgery bad events have been excluded (or controlled in advantage by Claim 0), (ii) TEE authentication ciphertexts are all replaced by random and indistinguishable ones (or controlled in advantage by Claim 1), all observable information

for $\mathscr{A}$ is indistinguishable from the real environment, except for the challenge ciphertext of the basic scheme. Therefore, any non-negligible distinguishing advantage of $\mathscr{A}$ is due to the indistinguishability of the challenge ciphertext of the basic scheme. Thus,

$$\left|\Pr[E_2]-\frac{1}{2}\right| \leq \mathrm{Adv}^{\text{Adaptive-IND-CPA}}_{\Pi_{\text{FPRE-OD}},\mathscr{B}'}. \tag{17}$$

Based on the above claims, we conclude:

$$\begin{aligned}\mathrm{Adv}^{\text{Adaptive}}_{\text{HRFPRE},\mathscr{A}} &= \left|\Pr[E_0]-\frac{1}{2}\right| \\ &\leq |\Pr[E_0]-\Pr[E_1]|+|\Pr[E_1]-\Pr[E_2]| \quad +\left|\Pr[E_2]-\frac{1}{2}\right| \\ &\leq \mathrm{Adv}^{\text{Adaptive-IND-CPA}}_{\Pi_{\text{FPRE-OD}},\mathscr{B}'}+\mathrm{Adv}^{\text{IND-CCA}}_{\Pi_{\text{TEE}},\mathscr{T}} \quad +\mathrm{Adv}^{\text{EUF-CMA}}_{\Pi_{\text{sig}},\mathscr{F}}.\end{aligned} \tag{18}$$

6.2 Security Strength

Confidentiality: HRFPRE resists attacks from external entities, including honest-but-curious RSUs and CSPs, as well as malicious users (i.e., Type I adversary), ensuring that only authorized users with valid private keys can recover the plaintext. The security proof simulates the adversary's oracle queries against both original and re-encrypted ciphertexts and gives a formal reduction under the computational DLIN assumption. Thus, both ciphertext forms preserve semantic security, i.e., IND-CPA security.

Collusion Resistance: The system mainly prevents collusion between RSUs and malicious vehicles (i.e., Type II adversary), since vehicles can only decrypt the helper ciphertext TC using their local key UK.

Against Collusion by Unauthorized Users: If an unauthorized vehicle colludes with an RSU, the RSU cannot obtain the corresponding transformation key CK from the current user list $List_i$ and thus cannot generate a valid transformed ciphertext PTC. Even if a forged $PTC=(ptc_1,ptc_2)$ is sent to the TEE, the TEE decrypts $Enc(K_{\text{TEE}},TK=(t_1,t_2))$ from $List_i$ and checks whether $ptc_1^{t_1} \stackrel{?}{=} ptc_2^{t_2}$. Invalid PTC fails the verification, so the TEE refuses to generate the helper ciphertext.

Against Collusion by Revoked Users: If a revoked user colludes with an RSU, the RSU may try to reuse the expired state $State_{i-1}$ and old list $List_{i-1}$ to generate PTC. However, after running $prog_t$, the TEE checks the current state $State_i$ and detects that the user is absent from the latest legitimate list $List_i$, thereby refusing decryption.

Replay Attack Resistance: Replay resistance is enforced by the TEE time-state module. After receiving PTC, the TEE verifies whether its internal time state i is synchronized with the timestamp in $State_i$ and rejects any outdated or invalid state, thus resisting Type III adversaries.

Resilience Against TEE Leakage: HRFPRE decouples the decryption privilege into the TEE-held helper key $TK=(t_1,t_2)$ and the user-held key $UK=\beta$. Even if a Type IV adversary obtains K_{TEE} and TK through side-channel leakage, these values cannot recover the plaintext alone, since final message recovery still requires $\Delta=TC^{-UK}$ with the secret scalar β. Thus, a compromised TEE state does not provide the pairing components needed to break the DLIN-based encryption core. In addition, the effect of TEE leakage on revocation is time-bounded: the TEE and KGC process the time-indexed

state $State_i = (\sigma_i, V_{root_i}, i)$ under synchronized clocks, so leaked states remain valid only in the current epoch. After anomaly detection or transition to the next interval, the KGC can rotate K_{TEE} and advance the state index, preventing the compromised TEE state from serving future transformed ciphertexts.

7. PERFORMANCE ANALYSIS

We compare the proposed HRFPRE scheme with existing works, demonstrating its efficiency and practicality from the perspectives of theoretical complexity and practical implementation.

7.1 Properties Comparison

Table 4 compares the functional features of HRFPRE with existing re-encryption schemes, where "×" and "✓" denote unsupported and supported features, respectively. All compared schemes are designed for data sharing. Unlike CGE [17], which follows ABE-to-ABE conversion, the remaining schemes perform IBE-to-ABE conversion to reduce standard ABE costs while preserving flexible access control. For revocation, CGE and EPREAR [15] rely on policy extension and key updates, whereas HRFPRE achieves immediate revocation with TEE assistance. Moreover, FABRIC [14] is proven secure in the standard model over asymmetric groups, while HRFPRE achieves adaptive security in the random oracle model under the DLIN assumption. FABRIC, EPREAR, and HRFPRE are unbounded and support dynamic attribute addition without system re-initialization. Similar to QYZ [27], HRFPRE also supports outsourced re-encryption and decryption, reducing user-side computation. Overall, HRFPRE provides balanced advantages in efficiency, revocation, security, and scalability for dynamic IoV environments.

Table 4. COMPARISON OF PROPERTIES

Scheme	Type	Revocation	Security	Unbounded	Asymmetric Pairing	Out-Rekey	Out-Dec
CGE [17]	ABE → ABE	✓	Selective	×	×	×	×
QYZ [27]	IBE → ABE	×	Selective	×	×	✓	✓
FABRIC [14]	IBE → ABE	×	Adaptive	✓	✓	×	×
EPREAR [15]	IBE → ABE	✓	Selective	✓	×	×	Fail to decrypt
OURS	IBE → ABE	✓	Adaptive	✓	✓	✓	✓

7.2 Computation Costs

Table 5 provides a comparative analysis regarding the computational overhead of the proposed HRFPRE scheme against existing schemes. It is important to note that FABRIC and our HRFPRE are constructed based on asymmetric pairing groups, while CGE, QYZ, and EPREAR are based on symmetric pairing groups. Although different group structures limit direct comparison, theoretical analysis can still reveal the characteristics of each scheme in terms of efficiency and overhead. It can be seen that compared to the CGE and QYZ schemes, FABRIC, EPREAR, and HRFPRE schemes have constant-size public parameters and support complex attribute structures. Meanwhile, compared to CGE, schemes using IBE for original information encryption maintain constant computational overhead during both original ciphertext generation and decryption operations, independent

Table 5. COMPARISON OF COMPUTATION COSTS

Scheme	Setup	Or-Enc.	Re-Enc.	Or-Dec.	Re-Dec.	Revocation
CGE [17]	$2T_{\mathbb{G}}+T_{\mathbb{G}_T}+T_p$	$(3a+2)T_{\mathbb{G}}+T_{\mathbb{G}_T}$	$aT_{\mathbb{G}}+(2a+2)T_p$	$aT_{\mathbb{G}}+(2a+1)T_p$	$(a+1)T_{\mathbb{G}}+(2a+1)T_p$	$(3a+2)T_{\mathbb{G}}$
QYZ [27]	$3T_{\mathbb{G}}+T_{\mathbb{G}_T}+T_p$	$4T_{\mathbb{G}}+T_{\mathbb{G}_T}+T_p$	$2T_p$	$2T_p$	$T_{\mathbb{G}_T}+T_p$	–
FABRIC [14]	$2T_{\mathbb{G}_1}+2T_{\mathbb{G}_2}+2T_{\mathbb{G}_T}+2T_p$	$9T_{\mathbb{G}_1}+3T_{\mathbb{G}_2}+2T_{\mathbb{G}_T}$	$6T_p$	$6T_p$	$9T_p$	–
EPREAR [15]	$6T_{\mathbb{G}}+3T_{\mathbb{G}_T}+3T_p$	$14T_{\mathbb{G}}$	$6T_p$	$6T_p$	$2T_p$	$(2a+1)T_{\mathbb{G}}$
OURS	$2T_{\mathbb{G}_2}+T_p$	$4T_{\mathbb{G}_1}+3T_{\mathbb{G}_2}+T_{\mathbb{G}_T}$	$4T_p$	$4T_p$	$T_{\mathbb{G}_T}+T_p$	$(\log_k n)C_{cf}$

• $T_{\mathbb{G}}, T_{\mathbb{G}_1}, T_{\mathbb{G}_2}, T_{\mathbb{G}_T}, T_p$: exponentiation in $\mathbb{G}, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ and bilinear pairing; a: number of attributes; k, n: Verkle tree branching factor and number of users; C_{cf}: cost of single commitment verification.

Table 6. COMPARISON OF COMMUNICATION COSTS

Scheme	Params	Or-CT	IBC-Key	ABC-Key	Re-Enc CT	Re-Enc Key
CGE [17]	$(a+3)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$2(a+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	–	$2(a+1)\|\mathbb{G}\|$	$(2a+1)\|\mathbb{G}\|+3\|\mathbb{G}_T\|$	$(3a+4)\|\mathbb{G}\|+\|\mathbb{G}_T\|$
QYZ [27]	$(a+7)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$3\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$2\|\mathbb{G}\|$	$(a+2)\|\mathbb{G}\|$	$(2a+3)\|\mathbb{G}\|+\|\mathbb{G}_T\|+(2a+1)\|\mathbb{Z}_p\|$	$(2a+4)\|\mathbb{G}\|+(2a+1)\|\mathbb{Z}_p\|$
FABRIC [14]	$3\|\mathbb{G}_1\|+3\|\mathbb{G}_2\|+2\|\mathbb{G}_T\|$	$6\|\mathbb{G}_1\|+3\|\mathbb{G}_2\|+\|\mathbb{G}_T\|$	$3\|\mathbb{G}_1\|+3\|\mathbb{G}_2\|$	$3(a+1)\|\mathbb{G}_1\|+3\|\mathbb{G}_2\|$	$3(a+1)\|\mathbb{G}_1\|+6\|\mathbb{G}_2\|+\|\mathbb{G}_T\|$	$3(a+1)\|\mathbb{G}_1\|+9\|\mathbb{G}_2\|$
EPREAR [15]	$6\|\mathbb{G}\|+3\|\mathbb{G}_T\|$	$9\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$5\|\mathbb{G}\|$	$\|\mathbb{G}\|$	$(2a+7)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(2a+10)\|\mathbb{G}\|$
OURS	$3\|\mathbb{G}_1\|+3\|\mathbb{G}_2\|+\|\mathbb{G}_T\|$	$2\|\mathbb{G}_1\|+3\|\mathbb{G}_2\|+\|\mathbb{G}_T\|$	$4\|\mathbb{G}_1\|$	$\|\mathbb{Z}_p\|$	$(a+1)\|\mathbb{G}_1\|+4\|\mathbb{G}_2\|+\|\mathbb{G}_T\|+(a+3)\|\mathbb{Z}_p\|$	$3\|\mathbb{G}_1\|+2\|\mathbb{G}_2\|+(a+3)\|\mathbb{Z}_p\|$

• $|\mathbb{G}|, |\mathbb{G}_1|, |\mathbb{G}_2|, |\mathbb{G}_T|, |\mathbb{Z}_p|$: the bit-length of an element in $\mathbb{G}, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathbb{Z}_p$; a: the size of attribute set and access policy.

of attribute quantity growth. Among adaptive-secure schemes, FABRIC requires 6 bilinear pairings for both re-encryption and original ciphertext decryption, whereas HRFPRE requires only 4. Finally, through the outsourced decryption mechanism, both QYZ and HRFPRE can offload most calculations to the proxy. When vehicle users decrypt re-encrypted ciphertexts, they only require one exponentiation operation in $\mathbb{G}_T$ and one bilinear pairing, resulting in lower user decryption costs. Regarding user revocation, the revocation-supporting CGE and EPREAR schemes are dependent on the number of attributes; their computational load increases as attributes increase. In contrast, the user revocation mechanism in this paper utilizes a Verkle tree verification approach, establishing a logarithmic relationship with system user population and consequently demonstrating superior scalability.

7.3 Communication Costs

By analyzing Table 6, we investigate the communication overhead of different schemes at various stages. In the initialization phase, the size of public parameters in CGE and QYZ grows linearly with the number of attributes, whereas FABRIC, EPREAR, and HRFPRE maintain constant parameter sizes. Regarding original encryption, schemes employing IBE to encrypt the original ciphertext (such as FABRIC, EPREAR, HRFPRE) have constant lengths for both the original ciphertext and the identity key, outperforming the CGE. Regarding the re-encryption phase, all evaluated schemes exhibit attribute-dependent length characteristics. However, since both QYZ and our scheme utilize re-encryption key outsourcing technology, their re-encryption keys and ciphertext lengths are associated with elements in $\mathbb{Z}_p$. Nevertheless, compared to the QYZ, ours involves only 3 elements in $\mathbb{G}_1$ and $a+3$ elements in $\mathbb{Z}_p$, resulting in lower communication overhead. Finally, for the decryption key in the ABC system, the user in the HRFPRE utilizes only one element in $\mathbb{Z}_p$ for decryption, without requiring excessive communication resources.

7.4 Implementation & Evaluation

To demonstrate practical computation and communication costs, we implemented and compared these schemes on a server. The TEE prototype was built using Intel SGX SDK (C++), with an Intel Core i5-11400 @ 2.60GHz processor and 32GB RAM. Cryptographic operations used Charm-Crypto 0.5 as the top-level interface and the PBC library [33]

for bilinear groups. To support the asymmetric structure of HRFPRE, we used BN254 curve parameters, while symmetric-pairing schemes such as CGE and QYZ used the standard Type A curve parameter "a1.param". We adopted access policies of the form "$Attr_1$ AND $Attr_2$ AND ... AND $Attr_N$", assuming all attributes are used for decryption. This represents the worst-case ABE matching scenario. Since EPREAR failed to decrypt correctly, we did not simulate its computation or communication costs. For 10, 20, 30, . . . , 100 attributes, Fig. 4 shows the time overhead under "AND" gates. In setup, EPREAR, FABRIC, and HRFPRE remain stable because of their unbounded design, and HRFPRE achieves the lowest cost with only one pairing. In key generation, competing schemes grow linearly with the number of attributes, while HRFPRE has no clear advantage because it generates two additional sets of transformation keys for ABE users. For outsourced re-encryption-key generation and delegated decryption, both QYZ and HRFPRE use cloud or edge nodes to assist users, but HRFPRE imposes significantly lower overhead. This is important in high-concurrency IoV scenarios, where large-scale outsourced computation can cause server saturation. The experiment also shows that TEE-side verification and helper ciphertext generation remain consistently low and independent of the number of attributes.

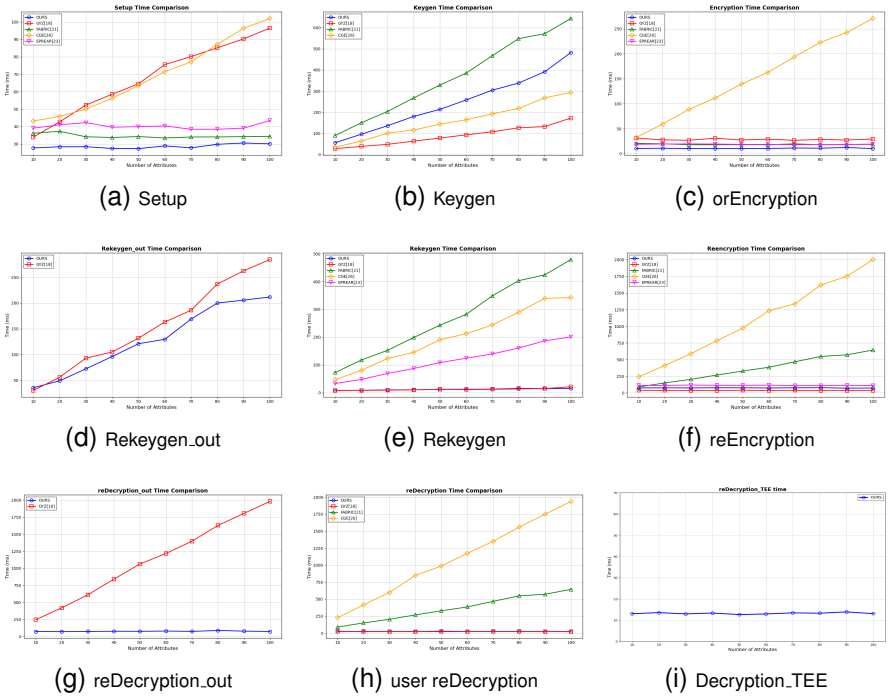
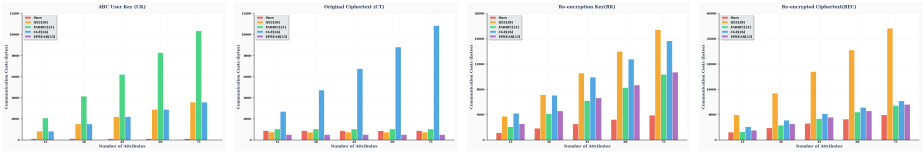


Fig. 4. Comparison of Time Overhead



(a) ABC-Key (b) orCT (c) Re-keys (d) REC

Fig. 5. Comparison of Communication Overhead

Under the same conditions, Fig. 5 compares communication overhead for UK, $orCT$, RK, and REC. For the ABC user key UK, HRFPRE uses only one random element in $|\mathbb{Z}_p|$ during registration, whereas competing schemes grow with the number of attributes. In addition, except for original ciphertext generation, HRFPRE achieves lower overhead in both re-encryption key RK and re-encrypted ciphertext REC than the other schemes.

In summary, our HRFPRE scheme excels in computational efficiency, functional completeness, and security, making it especially well-suited for highly dynamic and resource-constrained IoV deployments, offering a viable approach to enable efficient and dependable information transmission.

8. CONCLUSION

Aiming to address the efficiency and security limitations of existing schemes, we propose HRFPRE, a fast outsourced proxy re-encryption mechanism for IoV. HRFPRE employs TEE-assisted revocation, supports unbounded encryption/ decryption with dynamic attribute updates, and offloads re-encryption key and ciphertext generation to RSUs, thereby reducing the computational burden on on-board units. The scheme is proven adaptively secure under the DLIN assumption, and comparative analysis shows its advantages in functionality, computation, and communication overhead. Future work will further optimize secure IoV data sharing and explore blockchain smart contracts for cross-domain proxy re-encryption.

ACKNOWLEDGMENT

Financial support for this research was provided by the Open Project Program of Guangxi Key Laboratory of Digital Infrastructure (Grant No. GXDIOP2025007) and the Key Research and Development Program of Shaanxi (Grant Nos. 2022GY-028, 2022GY-050, and 2026CYYBXM024).

REFERENCES

1. Statista, “Automotive iot - worldwide,” Statista Market Outlook, 2025, accessed: Aug. 23, 2025. [Online]. Available: https://www.statista.com/outlook/tmo/internet-of-things/automotive-iot/worldwide

2. L. Wang, H. Zhong, J. Cui, J. Zhang, L. Wei, I. Bolodurina, and D. He, "Privacy-preserving and secure distributed data sharing scheme for vanets," *IEEE Transactions on Mobile Computing*, 2024.
3. Y. Hou, Y. Cao, H. Xiong, Y. Song, and L. Xu, "An efficient online/offline heterogeneous signcryption scheme with equality test for iovs," *IEEE Transactions on Vehicular Technology*, Vol. 72, no. 9, 2023, pp. 12 047–12 062.
4. Z. Guo, G. Wang, Y. Li, J. Ni, and G. Zhang, "Attribute-based data sharing scheme using blockchain for 6g-enabled vanets," *IEEE Transactions on Mobile Computing*, Vol. 23, no. 4, 2023, pp. 3343–3360.
5. H. Cui, R. H. Deng, and G. Wang, "An attribute-based framework for secure communications in vehicular ad hoc networks," *IEEE/ACM Transactions on Networking*, Vol. 27, no. 2, 2019, pp. 721–733.
6. X. Feng, K. Cui, L. Wang, Z. Liu, and J. Ma, "Pbag: A privacy-preserving blockchain-based authentication protocol with global-updated commitment in iovs," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 25, no. 10, 2024, pp. 13 524–13 545.
7. B. Waters, "Dual system encryption: Realizing fully secure ibe and hibe under simple assumptions," in *Annual international cryptology conference*. Springer, 2009, pp. 619–636.
8. V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM conference on Computer and communications security*, 2006, pp. 89–98.
9. Y. Zhou, L. Cao, Z. Qiao, R. Xu, Y. Han, J. Xing, B. Yang, Z. Xia, and M. Zhang, "A novel cloud-assisted authentication key agreement protocol for vanet," *IEEE Transactions on Vehicular Technology*, Vol. 73, no. 9, 2024, pp. 13 526–13 541.
10. Z. Guo, G. Wang, Y. Li, J. Ni, R. Du, and M. Wang, "Accountable attribute-based data-sharing scheme based on blockchain for vehicular ad hoc network," *IEEE Internet of Things Journal*, Vol. 10, no. 8, 2022, pp. 7011–7026.
11. M. Green, S. Hohenberger, and B. Waters, "Outsourcing the decryption of ABE ciphertexts," in *20th USENIX security symposium (USENIX Security 11)*, 2011.
12. M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," in *International conference on the theory and applications of cryptographic techniques*. Springer, 1998, pp. 127–144.
13. J. Zhang, S. Su, H. Zhong, J. Cui, and D. He, "Identity-based broadcast proxy re-encryption for flexible data sharing in vanets," *IEEE Transactions on Information Forensics and Security*, Vol. 18, 2023, pp. 4830–4842.
14. L. Wang, Y. Lin, T. Yao, H. Xiong, and K. Liang, "Fabric: Fast and secure unbounded cross-system encrypted data sharing in cloud computing," *IEEE Transactions on Dependable and Secure Computing*, Vol. 20, no. 6, 2023, pp. 5130–5142.
15. X. Li, Y. Xie, C. Peng, E. Luo, X. Li, and Z. Zhou, "Eprear: An efficient attribute-based proxy re-encryption scheme with fast revocation for data sharing in aiot," *IEEE Transactions on Mobile Computing*, 2025.
16. B. Li, L. Deng, H. Liu, Y. Mou, Y. Qu, and N. Wang, "A cca-secure pairing-free broadcast proxy re-encryption scheme for vanets," *IEEE Transactions on Vehicular Technology*, 2025.
17. C. Ge, W. Susilo, Z. Liu, J. Baek, X. Luo, and L. Fang, "Attribute-based proxy re-encryption with direct revocation mechanism for data sharing in clouds," in *Pro-*

ceedings of the ACM Turing Award Celebration Conference-China 2023, 2023, pp. 164–165.

18. X. Li, G. Yang, T. Xiang, S. Xu, B. Zhao, H. Pang, and R. H. Deng, "Make revocation cheaper: Hardware-based revocable attribute-based encryption," in *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2024, pp. 3109–3127.
19. D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Annual international cryptology conference*. Springer, 2001, pp. 213–229.
20. B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *International workshop on public key cryptography*. Springer, 2011, pp. 53–70.
21. L. Meng, L. Chen, Y. Tian, and M. Manulis, "Fabesa: Fast (and anonymous) attribute-based encryption under standard assumption," in *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, 2024, pp. 4688–4702.
22. S. Agrawal and M. Chase, "Fame: Fast attribute-based message encryption," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 665–682.
23. X. Liang, Z. Cao, H. Lin, and J. Shao, "Attribute based proxy re-encryption with delegating capabilities," in *Proceedings of the 4th international symposium on information, computer, and communications security*, 2009, pp. 276–286.
24. C. Ge, W. Susilo, J. Baek, Z. Liu, J. Xia, and L. Fang, "A verifiable and fair attribute-based proxy re-encryption scheme for data sharing in clouds," *IEEE Transactions on Dependable and Secure Computing*, Vol. 19, no. 5, 2021, pp. 2907–2919.
25. P. Duan, Z. Ma, H. Gao, T. Tian, and Y. Zhang, "Multi-authority attribute-based encryption scheme with access delegation for cross blockchain data sharing," *IEEE Transactions on Information Forensics and Security*, 2024.
26. X. Liu, W. Chen, and Y. Xia, "Security-aware information dissemination with fine-grained access control in cooperative multi-rsu of vanets," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 23, no. 3, 2020, pp. 2170–2179.
27. Q. Zhang, Y. Fu, J. Cui, D. He, and H. Zhong, "Efficient fine-grained data sharing based on proxy re-encryption in iiot," *IEEE Transactions on Dependable and Secure Computing*, Vol. 21, no. 6, 2024, pp. 5797–5809.
28. C. Ge, W. Susilo, J. Baek, Z. Liu, J. Xia, and L. Fang, "Revocable attribute-based encryption with data integrity in clouds," *IEEE Transactions on Dependable and Secure Computing*, Vol. 19, no. 5, 2021, pp. 2864–2872.
29. H. Xiong, X. Huang, M. Yang, L. Wang, and S. Yu, "Unbounded and efficient revocable attribute-based encryption with adaptive security for cloud-assisted internet of things," *IEEE Internet of Things Journal*, Vol. 9, no. 4, 2021, pp. 3097–3111.
30. J. Zhang, X. Liu, J. Cui, H. Zhong, L. Wei, I. Bolodurina, and D. He, "Cbdds: Secure and revocable cache-based distributed data sharing for vehicular networks," *IEEE Transactions on Mobile Computing*, Vol. 23, no. 6, 2023, pp. 6579–6591.
31. F. Meng and L. Cheng, "Tsr-abe: Traceable and server-aided revocable ciphertext-policy attribute-based encryption under static assumptions," *IEEE Transactions on Information Forensics and Security*, 2024.
32. J. Kuszmaul, "Verkle trees," *Verkle Trees*, Vol. 1, no. 1, 2019, pp. 1–10.

33. B. Lynn, "The pairing-based cryptography (pbc) library," 2006, [Online]. [Online]. Available: http://crypto.stanford.edu/pbc



Tingting Li received the B.S. degree from Shandong Jianzhu University, Jinan, China, in 2024. She is currently pursuing the M.S. degree in applied mathematics with Xidian University, Xi'an, China. Her current interests include cryptography, computer security, and cloud security.



Leyou Zhang received the M.S. and Ph.D. degrees from Xidian University, Xi'an, China, in 2002 and 2009, respectively. He is currently a Professor with Xidian University. His current research interests include cryptography, network security, cloud security, and computer security.



Qing Wu received the M.S. and Ph.D. degrees from Xidian University, Xi'an, China, in 2006 and 2009, respectively. She is currently a Professor with Xi'an University of Posts and Communications, Xi'an. Her current research interests include artificial intelligence security and cloud security.



Fei Zhou received the B.E. degree in geology from Guilin Institute of Technology, Guilin, China, in 1995, and on the job graduate students in geodesy and surveying engineering from Guilin Institute of Technology in 2004. He is currently the Director of Guangxi Zhuang Autonomous Region Big Data Research Institute and the Director with the Guangxi Key Laboratory of Digital Infrastructure. His research interests include big data, cloud computing, networks, computers, and network security.



Yuxing Wei received the bachelor's degree in computer science and technology from the Guangxi University for Nationalities in July 2013. He is currently affiliated with the Information Center of Guangxi Zhuang Autonomous Region. His research interests include cloud computing, government network systems, and network and data security.