RESEARCH ARTICLE

A Blockchain-Based Revocable Identity-Based Proxy Re-Encryption Scheme with Cryptographic Reverse Firewalls for Secure Data Sharing

Negalign Wake Hundera[1,2], **Rashad Elhabob**[1,2], **Deepak Adhikari**[1,2] and **Hu Xiong**[1,2,*]

[1] School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China
[2] Network and Data Security Key Laboratory of Sichuan Province, Chengdu, China

Abstract

With the rapid growth of Internet of Things (IoT) applications and decentralized data-driven systems, secure and flexible data sharing remains a critical challenge. Identity-Based Proxy Re-Encryption (IB-PRE) is an effective cryptographic primitive for enabling fine-grained access delegation without exposing plaintext data. However, existing IB-PRE schemes remain vulnerable to algorithm substitution attacks (ASA), malicious key generation, inefficient ciphertext management, and the lack of practical revocation mechanisms. To address these limitations, we propose a blockchain-based revocable identity-based proxy re-encryption scheme with cryptographic reverse firewalls (BRIBPR-CRF) for secure IoT data sharing. The proposed scheme integrates CRFs into the key generation, encryption, and proxy re-encryption key generation processes to mitigate ASA and limit the impact of potentially compromised cryptographic components. An efficient identity revocation mechanism enables the removal of compromised or expired users without requiring global key updates. In addition, a consortium blockchain is employed to record initial ciphertexts and execute proxy re-encryption via smart contracts, thereby eliminating single points of failure and reducing trust assumptions compared with traditional proxy-based systems. We formally prove the security of BRIBPR-CRF in the random-oracle model. Finally, extensive performance evaluations demonstrate that the proposed scheme achieves lower communication, computation, and energy overhead compared with existing schemes, making it suitable for secure and scalable data sharing in decentralized and semi-trusted IoT environments.

Keywords: secure data sharing, proxy re-encryption, blockchain, revocation, cryptographic reverse firewalls.

1 Introduction

The rapid growth of IoT applications and decentralized data-driven systems has made secure and efficient data sharing a fundamental requirement in modern information systems [1, 2]. Applications such as cloud computing, IoT, the Internet of Vehicles (IoV), and

Submitted: 22 January 2026
Accepted: 23 March 2026
Published: 27 March 2026

Vol. 2, **No.** 1, 2026.
10.62762/JRSC.2026.796877

***Corresponding author:**
✉ Hu Xiong
xionghu.uestc@gmail.com

Citation
Hundera, N. W., Elhabob, R., Adhikari, D., & Xiong, H. (2026). A Blockchain-Based Revocable Identity-Based Proxy Re-Encryption Scheme with Cryptographic Reverse Firewalls for Secure Data Sharing. *Journal of Reliable and Secure Computing*, 2(1), 50–65.

edge–cloud architectures require continuous sharing of sensitive data among multiple parties over open networks [3]. These environments typically involve dynamic user participation, resource-constrained devices, and untrusted third-party entities, which introduce significant security and privacy challenges [4]. In typical IoT scenarios, data generated by resource-constrained devices are outsourced to edge or cloud servers, where flexible, fine-grained, and privacy-preserving access control is required for multiple data consumers. Ensuring the confidentiality, access control, and accountability of shared data under such conditions remains a critical problem, particularly for encrypted outsourced data where ciphertext sharing is restricted [4, 26, 30].

To address these challenges, proxy re-encryption (PRE) has been proposed to allow a semi-trusted proxy to transform ciphertexts from one user to another without disclosing the underlying plaintext, thereby enabling flexible and fine-grained access control [5]. In this setting, a data owner can delegate decryption rights to selected users without exposing secret keys while reducing the risk of data leakage. However, PRE systems typically rely on semi-trusted proxy servers, which introduce inherent security and efficiency concerns. Such proxy servers may be compromised due to financial incentives, political pressure, or insider attacks, resulting in incomplete security guarantees and limited scalability in decentralized environments.

To mitigate these limitations, blockchain technology has been introduced as a decentralized alternative to traditional proxy servers [6]. In blockchain-based PRE schemes, smart contracts can replace third-party proxies for re-encryption operations while providing transparency, tamper resistance, and resilience against single points of failure. Moreover, data owners can verify the integrity and authenticity of their data on the blockchain without sharing their decryption keys with external entities. Building on PRE, IB-PRE further simplifies key management by eliminating certificates and enabling access delegation based on user identities [10]. As a result, IB-PRE has been widely applied in healthcare data sharing, cloud storage, IoT systems, and vehicular networks [11, 12, 28, 29]. Nevertheless, existing IB-PRE schemes still suffer from inefficient ciphertext management, limited scalability, and a lack of effective and practical revocation mechanisms, which significantly restrict their applicability in dynamic IoT environments.

Although cryptographic algorithms are designed to operate securely, they may be maliciously modified or compromised in practice. For instance, an insider adversary can embed hidden backdoors into cryptographic implementations to exfiltrate sensitive information, as highlighted by real-world disclosures on compromised cryptographic systems [7]. Bellare et al. [8] formalized this threat as ASA, in which an adversary subtly modifies cryptographic algorithms to leak secret information. CRFs, introduced by Mironov et al. [9], act as a defensive layer that sanitizes cryptographic inputs and outputs, reducing information leakage and strengthening resilience against ASA under partial compromise of cryptographic implementations. Motivated by this approach, we design BRIBPR-CRF, which incorporates CRFs into the key generation, encryption, and proxy re-encryption generation processes to enhance resistance against ASA while supporting an efficient identity revocation mechanism in decentralized and semi-trusted IoT environments.

1.1 Related Work

PRE is a cryptographic technique that can transform ciphertexts without revealing the plaintexts. This enables secure data sharing among different parties in decentralized environments. Blaze et al. [5] introduced the first PRE scheme in which the data owner encrypts the plaintext and sends the ciphertext to a proxy. The proxy then re-encrypts the ciphertext using a re-encryption key so that both the data owner and an authorized receiver can decrypt it. However, this scheme assumes that the data owner knows the receiver's public key beforehand which may not be realistic in many practical scenarios. To address this, an IB-PRE scheme was proposed [13] where the data owner only needs the receiver's identity and the proxy re-encrypts ciphertexts without learning the plaintext. However, their scheme is only secure against CPA [14] and an adversary may still learn partial information by submitting chosen ciphertexts to the proxy. To achieve stronger security, Susilo et al. [15] proposed the honest re-encryption attack (HRA) notion under which a proxy learns no information about the plaintext even when it follows the re-encryption protocol honestly.

Inspired by their work, Zhao et al. [16] proposed HRA-secure PRE and further extended it to attribute-based PRE, which enables fine-grained access control based on receiver attributes. Some PRE schemes were later constructed using bilinear pairings [17] which achieve adaptive security and revocability. These schemes allow a data owner

to revoke re-encryption rights at any time without affecting the access rights of legitimate users and provide resistance to proxy collusion attacks, in which a proxy cannot cooperate with unauthorized users to recover plaintexts. Some PRE schemes have been designed to support complex data processing using homomorphic encryption [18], enabling a proxy to execute function transformations on ciphertexts without learning plaintexts or private keys. However, many existing PRE schemes still suffer from ciphertext expansion, large re-encryption key sizes, and high computational overhead which limit their practicality in dynamic and resource-constrained environments.

Considering that a proxy server may behave maliciously, it can tamper with ciphertexts or leak sensitive information to unauthorized parties. To mitigate this risk, blockchain has been integrated into PRE schemes to decentralize trust. Using this approach, Manzoor et al. [19] designed a PRE scheme that enables data owners and data users to dynamically create smart contracts at runtime without relying on a trusted third party. Fan et al. [39] further proposed a more practical model in which encrypted data are stored on cloud servers, while access control and re-encryption operations are maintained on the blockchain. Subsequently, Singh et al. [21] developed a blockchain-based PRE scheme for distributed systems. However, these schemes typically incur substantial computational and communication overhead and remain vulnerable to backdoor attacks, making them unsuitable for practical large-scale data-sharing scenarios.

To address this issue, Mironov et al. [9] introduced the concept of CRFs in 2015. CRFs ensure security even when a device is compromised and protect confidential information from attackers. Subsequently, Zhou et al. [22] proposed a certificateless scheme with CRFs and proved its indistinguishability against CPA. Later, in 2022, they further introduced an IB-PRE scheme for WBANs [27], which avoids certificate management issues in PKI and resists ASA attacks. Ma et al. [23] introduced a CP-ABE-CRF scheme to reduce computational costs while protecting against data exfiltration, whereas Jin et al. [20] proposed a blockchain-based PRE-CRF scheme. However, their scheme suffers from mathematical inconsistencies and remains vulnerable to backdoor and CPA attacks. Despite these advances, the integration of CRFs with revocable IB-PRE and blockchain-based re-encryption remains unexplored. This gap motivates the design of BRIBPR-CRF, which combines revocable IB-PRE, blockchain-based re-encryption, and CRFs to achieve secure, efficient, and leakage-resilient data sharing.

1.2 Contributions

To resist backdoor attacks and support efficient revocation in secure data-sharing systems while ensuring decentralization, data tamper resistance, and the elimination of single points of failure introduced by semi-trusted proxies, this paper proposes BRIBPR-CRF scheme. The main contributions are as follows:

1. First, we propose a secure BRIBPR-CRF scheme that enables efficient data sharing in decentralized environments by integrating revocable IB-PRE, blockchain-based re-encryption, and CRFs.

2. BRIBPR-CRF integrates two CRFs to prevent ASA attacks: one at the PKG and the other at the data owner side. These CRFs randomize keys, ciphertexts, and proxy keys such that adversaries cannot distinguish between the original and re-randomized values. The proposed scheme ensures strong resistance to backdoor and data exfiltration attacks even if secret keys are exposed.

3. Smart contracts deployed on the blockchain are used to perform proxy re-encryption operations thereby reducing the reliance on the trustworthiness of any single participant. This approach improves efficiency and eliminates single points of failure.

4. We prove that BRIBPR-CRF is secure under standard cryptographic assumptions against IND-sID-CPA and ASA attacks. Furthermore, performance evaluations indicate that the proposed scheme achieves lower computational and communication overhead and higher energy efficiency than existing schemes.

1.3 Organization

The remainder of this paper is organized as follows. Section 2 introduces the preliminaries. Section 3 presents the formal model of BRIBPR-CRF and Section 4 describes the construction. Sections 5 and 6 provide the security analysis and performance evaluation respectively. Section 7 concludes the paper.

2 Preliminaries

This section briefly introduces the notations, bilinear pairing, complexity assumptions, CRFs, and blockchain and smart contracts.

2.1 Notations

The notations used throughout this paper are summarized in Table 1.

Table 1. Notation and descriptions.

Notation	Descriptions
λ	Security parameter
$\mathbb{G}_1$	Additive cyclic group
$\mathbb{G}_2$	Multiplicative cyclic group
$\hat{e}$	A bilinear pairing
s	Master secret key of the PKG
P_{pub}	Master public key of the PKG
H_1, H_2	Cryptographic hash functions
ID_U	Identity of a user U
ID_S	Identity of the data sender
ID_R	Identity of the data receiver
Q_U	Public key of user U
SK_U	Private key of user U
Q_S	Public key of the data sender
SK_S	Data sender private key
Q_R	Receiver's public key
SK_R	Private key of receiver
RL_t	Revocation list at epoch t
CRF	Cryptographic reverse firewall
ROM	Random oracle model
CS	Cloud server
par	Public system parameters
$rk_{S\to R}$	Proxy key
C	Original ciphertext
C_R	Re-encrypted ciphertext

2.2 Bilinear Pairing

Let $\mathbb{G}_1$ be an additive cyclic group and $\mathbb{G}_2$ be a multiplicative cyclic group, both of prime order q. Let P be a generator of $\mathbb{G}_1$. A bilinear pairing is a map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ that satisfies the following properties.

1. *Bilinearity:* For all $P, Q \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_q$, $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$.
2. *Non-degeneracy:* There exist $P, Q \in \mathbb{G}_1$ such that $\hat{e}(P, Q) \neq 1$, where 1 denotes the identity element in $\mathbb{G}_2$.
3. *Computability:* $\hat{e}(P, Q)$ is efficiently computable for all $P, Q \in \mathbb{G}_1$.

2.3 Complexity Assumptions

Given $\mathbb{G}_1$, $\mathbb{G}_2$, a prime order q, a generator $P \in \mathbb{G}_1$, and a bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ as defined above, the security of BRIBPR-CRF relies on the following hardness assumption.

Definition 1: [Decisional Bilinear Diffie–Hellman (DBDH)] Let $a, b, c \in \mathbb{Z}_q$ and let $\theta \in \mathbb{G}_2$. Given a tuple (P, aP, bP, cP, θ), the DBDH problem is to decide whether $\theta = \hat{e}(P, P)^{abc}$. The advantage of adversary $\mathcal{A}$ in solving DBDH is defined as

$$\mathrm{Adv}_{\mathcal{DBDH}} = \left| Pr\left[\mathcal{A}(P, aP, bP, cP, \hat{e}(P, P)^{abc}) = 1\right]\right| - \left|\Pr[\mathcal{A}(P, aP, bP, cP, \theta) = 1]\right|.$$

2.4 Cryptographic Reverse Firewall (CRF)

A cryptographic reverse firewall (CRF) [9] is a protective algorithmic wrapper that filters the inputs and outputs of a potentially compromised cryptographic party in order to preserve correctness and security, even in the presence of ASA. Formally, let P denote a cryptographic party implementing a set of algorithms, and let $\mathcal{W}$ denote a CRF. The composition $\mathcal{W} \circ P$ represents the execution of P with all inputs to and outputs from P filtered by $\mathcal{W}$. In this work, we restrict attention to stateless CRFs for simplicity. A valid CRF must satisfy the following properties.

1. *Functionality Preservation:* The CRF preserves the correctness of the underlying cryptographic algorithms, even if the implementation of the party is maliciously modified.
2. *Weak Security Preservation:* If a cryptographic scheme satisfies a security notion when executed by an honest party P, then the same security notion remains satisfied when P is replaced by any functionality-preserving subverted implementation P', provided that all inputs and outputs are filtered by the CRF $\mathcal{W}$.
3. *Weak Exfiltration Resistance:* The CRF prevents the exfiltration of sensitive information through algorithm-substitution attacks, except with negligible probability.

Functionality-Maintaining CRFs: A CRF $\mathcal{W}$ is said to be functionality maintaining for a party P if, for all $k \geq 1$, the composed execution $\mathcal{W}^k \circ P$ is functionally equivalent to P. We write $\mathcal{W} \circ P \equiv P$.

Weak Security-Preserving CRFs: Let $\mathcal{F}$ be a functionality criterion and $\mathcal{S}$ a security criterion. A CRF $\mathcal{W}$ is weakly security preserving if,for any functionality-preserving subverted party P', whenever P satisfies $\mathcal{S}$, the composed execution $\mathcal{W} \circ P'$ also satisfies $\mathcal{S}$.

Weak Exfiltration Resistance: A CRF $\mathcal{W}$ is weakly exfiltration resistant if no adversary can distinguish, with non-negligible advantage, between the real execution of $\mathcal{W} \circ P'$ and the ideal execution of $\mathcal{W} \circ P$ in the leakage experiment $\mathsf{G}_{\mathsf{LEAK}}$ defined in Section X.

2.5 Blockchain and Smart Contracts

Blockchain technology provides a decentralized and tamper-resistant ledger that enables publicly verifiable data storage and the execution of predefined logic without relying on a trusted third party [40]. In BRIBPR-CRF, the blockchain functions as an immutable storage and coordination layer rather than a trusted key management entity. It stores encrypted and re-encrypted ciphertexts in a publicly auditable manner to ensure data integrity and availability. Smart contracts deterministically execute the proxy re-encryption process based on publicly verifiable inputs such as randomized ciphertexts and re-encryption keys [31]. This design prevents unauthorized ciphertext modification and ensures transparent and correct re-encryption. The consortium blockchain smart contract (CBSC) executes the ReEnc algorithm and stores the resulting ciphertexts on-chain. Since all sensitive cryptographic material is protected by CRFs, the blockchain and smart contracts need not be trusted for confidentiality; instead, they provide auditability, non-repudiation, and verifiable correctness of re-encryption.

3 Formal Model of BRIBPR-CRF

This section presents the system model, the BRIBPR-CRF framework, and the associated security notions.

3.1 System Model

The proposed BRIBPR-CRF scheme consists of six entities: a Private Key Generator (PKG), a PKG cryptographic reverse firewall (PKG-CRF), a data sender (DS), a data sender cryptographic reverse firewall (DS-CRF), a consortium blockchain server center (CBSC), and a data receiver (DR), as shown in Figure 1.

The PKG initializes the system and generates private keys for users, while the PKG-CRF randomizes public parameters and user secret keys in an epoch-dependent manner, and enforces user revocation by issuing valid keys only to non-revoked users. The data sender encrypts messages under its identity and interacts with the DS-CRF, which randomizes ciphertexts and proxy re-encryption keys to mitigate key exfiltration and ASA. The CBSC stores ciphertexts and performs re-encryption operations using proxy keys without learning any secret information. Finally, the data receiver retrieves re-encrypted ciphertexts from the CBSC and decrypts them using its epoch-dependent private key, provided that the receiver has not been revoked.

3.2 BRIBPR-CRF Framework

The BRIBPR-CRF framework consists of the following algorithms:

1. *Setup:* Takes λ as input and outputs par and s.
2. *PKG-CRF$_{Setup}$:* At the beginning of each epoch t, the PKG-CRF selects $\alpha_t \in \mathbb{Z}_q^*$, randomizes $P'_{pub_t} = \alpha_t P_{pub}$ and broadcasts par' to all users.
3. *KeyGen:* Takes s and $ID_U \in \{0,1\}^*$ as input and outputs SK_U which is forwarded to the PKG-CRF.
4. *PKG-CRF$_{KeyGen}$:* Takes SK_U and the current epoch t as input and outputs $SK'_{U_t} = \alpha_t s Q_U$ if $ID_U \notin RL_t$; otherwise, it outputs $\perp$.
5. *Enc:* Takes M, ID_S, and the current epoch t as input and outputs C_{S_t}.
6. *DS-CRF$_{Enc}$:* Takes C_{S_t} as input, selects $\beta \in \mathbb{Z}_q^*$, and outputs C'_{S_t}.
7. *ReKeyGen:* Takes the sender secret key SK'_{S_t} and the receiver public key Q_R as input and outputs $rk_{S\to R}$, which is forwarded to the DS-CRF.
8. *DS-CRF$_{ReKeyGen}$:* Takes $rk_{S\to R}$ as input, randomizes it using β, and outputs the randomized proxy key $rk'_{S\to R}$.
9. *ReEnc:* Takes $rk'_{S\to R}$ and C'_{S_t} as input and outputs the re-encrypted ciphertext C_R.
10. *Dec$_1$:* Takes C'_{S_t} and SK'_{S_t} as input and outputs the plaintext message M.
11. *Dec$_2$:* Takes C_R, SK'_{R_t}, and the identities (ID_S, ID_R) as input and returns M if valid; otherwise, it outputs $\perp$.

3.3 Security Definition

The BRIBPR-CRF scheme provides confidentiality (IND-sID-CPA) and resistance to algorithm-substitution attacks (ASAs). These security properties are defined below.

Confidentiality: The IND-sID-CPA security of BRIBPR-CRF is defined via a game played between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ as follows.

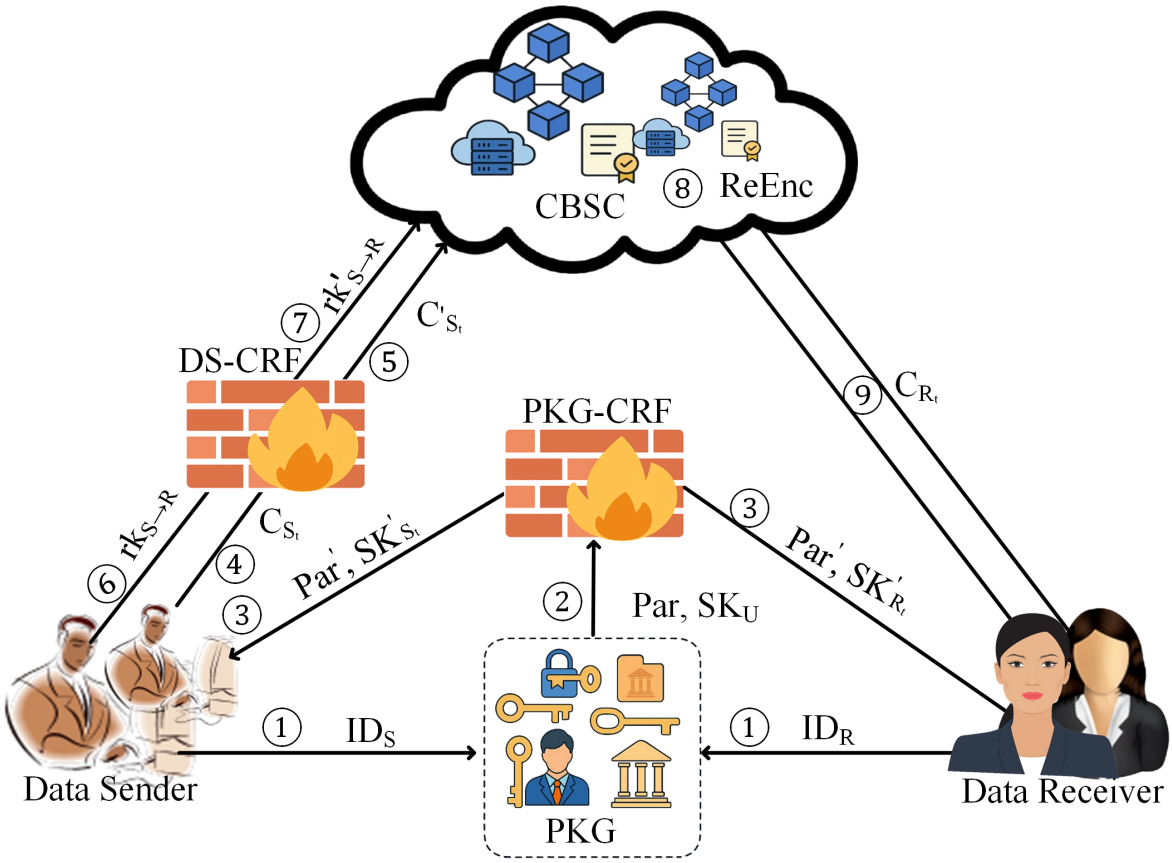

Figure 1. The System model for BRIBPR-CRF.

Initialization: $\mathcal{C}$ runs *Setup* and provides par to $\mathcal{A}$.

Phase 1: $\mathcal{A}$ adaptively issues polynomially many queries.

1. *QKeyGen:* $\mathcal{A}$ selects $ID_U \in \{0,1\}^*$ and epoch $t \in \mathbb{N}$. $\mathcal{C}$ runs *KeyGen* and *PKG-CRF*$_{KeyGen}$ to return the epoch secret key SK'_{U_t} if $ID_U \notin RL_t$. Otherwise it returns $\perp$.
2. *QReKeyGen:* $\mathcal{A}$ selects identities (ID_S, ID_R) and epoch t. $\mathcal{C}$ obtains SK'_{S_t} via *QKeyGen*, runs *ReKeyGen* and *DS-CRF*$_{ReKeyGen}$, and returns $rk'_{S\to R}$.
3. *QEnc:* $\mathcal{A}$ selects a message M, an identity ID_S, and epoch t. $\mathcal{C}$ runs *Enc* and *DS-CRF*$_{Enc}$ to return C'_{S_t}.
4. *QReEnc:* $\mathcal{A}$ submits C'_{S_t} and $rk'_{S\to R}$. $\mathcal{C}$ runs *ReEnc* and returns C_R.

Challenge: At the end of Phase 1, $\mathcal{A}$ selects m_0, m_1 of equal length, identities (ID^*_S, ID^*_R), and an epoch t^* such that $ID^*_R \notin RL_{t^*}$. $\mathcal{C}$ picks $b \xleftarrow{\$} \{0,1\}$ and computes the challenge ciphertext as follows: it runs $Enc(m_b, ID^*_S, t^*)$ and *DS-CRF*$_{Enc}$ to obtain $C'^*_{S_{t^*}}$, then runs *ReKeyGen* and *DS-CRF*$_{ReKeyGen}$ to obtain $rk'^*_{S^*\to R^*}$, and finally runs *ReEnc* to obtain $C^*_{R_{t^*}}$. $\mathcal{C}$ returns $C^*_{R_{t^*}}$ to $\mathcal{A}$.

Phase 2: $\mathcal{A}$ continues to issue queries as in Phase 1 subject to the following restrictions: $\mathcal{A}$ is not allowed to query *QKeyGen* on (ID^*_R, t^*) obtain $SK'_{R_{t^*}}$ and is not allowed to query *QReKeyGen* on (ID^*_S, ID^*_R, t^*).

Guess: $\mathcal{A}$ outputs $b' \in \{0,1\}$. The advantage of $\mathcal{A}$ is

$$\mathrm{Adv}^{\text{sIND-CPA}}_{\mathcal{A}} = \left|\Pr[b' = b] - \frac{1}{2}\right|.$$

Definition 2 (IND-sID-CPA Security): The BRIBPR-CRF scheme achieves IND-sID-CPA security if no adversary $\mathcal{A}$ running in time t can achieve advantage ε after making at most q_{kg} *QKeyGen* queries, q_{rk} *QReKeyGen* queries, q_{enc} *QEnc* queries,and q_{rec} *QReEnc* queries in the above security game. (The proof is provided in Section 5.)

ASA Resistance via CRF: We define the resistance of the BRIBPR-CRF scheme against ASAs using a weak exfiltration resistance game, denoted by $\mathsf{G}_{\mathsf{LEAK}}$. Under this model, the proposed CRF-based defense is intended to satisfy three essential properties against ASA, namely: (i) functionality preservation, (ii)

weak security preservation, and (iii) weak exfiltration resistance.

Game $\mathsf{G}_{\mathsf{LEAK}}$**.** The game is played between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ as follows.

Tampering: $\mathcal{A}$ specifies a set of potentially malicious but functionality-preserving algorithms *Setup**, *KeyGen**, *ReKeyGen**, and *Enc**, and submits them to $\mathcal{C}$. These algorithms are required to preserve correctness of the scheme on all valid inputs.

Initialization: $\mathcal{C}$ runs *Setup** to obtain par and initializes the PKG-CRF. For any identity query ID, $\mathcal{C}$ runs *KeyGen**, and filters its output through the PKG-CRF, and returns the corresponding epoch-dependent secret key SK'_{ID_t} to $\mathcal{A}$.

Phase 1: $\mathcal{A}$ adaptively issues polynomially many queries as in the IND-sID-CPA game. All outputs of modified algorithms are filtered by the corresponding CRFs before being returned to $\mathcal{A}$.

Challenge: At the end of Phase 1, $\mathcal{A}$ selects two equal-length messages m_0 and m_1 and a challenge identity ID^*_S. $\mathcal{C}$ picks a random bit $b \in \{0,1\}$ and encrypts m_b using *Enc** under ID^*_S in the challenge epoch t^* to obtain $C_{S^*_t}$. The ciphertext is then filtered through the DS-CRF to obtain $C'_{S_{t^*}}$, which is returned to $\mathcal{A}$.

Phase 2: $\mathcal{A}$ continues to issue polynomially many queries as in Phase 1 under the same restrictions.

Guess: Finally, $\mathcal{A}$ outputs a bit $b' \in \{0,1\}$. The advantage of $\mathcal{A}$ is defined as

$$\mathrm{Adv}^{\mathsf{ASA}}_{\mathcal{A}} = \left| \Pr[b' = b] - \frac{1}{2} \right|.$$

Definition 3 (*Weak ASA Resistance*)*:* The BRIBPR-CRF scheme is said to achieve weak exfiltration resistance against ASAs if, for any $\mathcal{A}$, the advantage $\mathrm{Adv}^{\mathsf{ASA}}_{\mathcal{A}}$ in the game $\mathsf{G}_{\mathsf{LEAK}}$ is negligible, even when $\mathcal{A}$ is allowed to arbitrarily modify all non-CRF algorithms while preserving their functionality.

4 BRIBPR-CRF Construction

The scheme involves a data sender with identity ID_S and a data receiver with identity ID_R. It employs two CRFs, one associated with the PKG and the other with the data sender. The system operates over discrete time periods indexed by epochs $t \in \mathbb{N}$. At each epoch, a revocation list RL_t records the set of revoked identities. The scheme consists of the following eleven algorithms.

1. Setup: Given λ, the PKG selects an additive group $\mathbb{G}_1$ and a multiplicative group $\mathbb{G}_2$, both of prime order q. Let P be a generator of $\mathbb{G}_1$. A bilinear pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$. The PKG defines two cryptographic hash functions $H_1 : \{0,1\}^* \to \mathbb{G}_1$ and $H_2 : \mathbb{G}_2 \times \{0,1\}^* \times \{0,1\}^* \times \mathbb{N} \to \mathbb{G}_1$. It chooses a master secret $s \in \mathbb{Z}^*_q$, computes $P_{pub} = sP$, and sends $par = \{\mathbb{G}_1, \mathbb{G}_2, q, P, \hat{e}, P_{pub}, H_1, H_2\}$ to the PKG-CRF and keeps s secret.

2. PKG-CRF$_{Setup}$**:** At the beginning of each epoch t, the PKG-CRF selects a random value $\alpha_t \in \mathbb{Z}^*_q$, computes $P'_{pub_t} = \alpha_t P_{pub}$, and broadcasts $par' = \{\mathbb{G}_1, \mathbb{G}_2, q, \hat{e}, P, P'_{pub_t}, H_1, H_2\}$ to all users.

3. KeyGen: Given an identity $ID_U \in \{0,1\}^*$, the PKG computes $Q_U = H_1(ID_U)$ and generates the user secret key $SK_U = sQ_U$. It then sends SK_U to the PKG-CRF.

4. PKG-CRF$_{KeyGen}$**:** For each epoch t, the PKG-CRF randomizes SK_U using α_t to obtain $SK'_{U_t} = \alpha_t s Q_U$ and sends SK'_{U_t} to the user if and only if $ID_U \notin RL_t$; otherwise, it outputs $\perp$.

5. Enc: Given a message M, the identity ID_S of the data sender, and the current epoch t, the data sender computes $Q_S = H_1(ID_S)$ and selects $r \in \mathbb{Z}^*_q$. It then executes:

1. Compute $C_1 = rP$.
2. Compute $C_2 = M \cdot \hat{e}(P'_{pub_t}, Q_S)^r$.
3. Output $C_{S_t} = (t, C_1, C_2)$.

The data sender sends C_{S_t} to the DS-CRF for further randomization.

6. DS-CRF$_{Enc}$**:** Upon receiving $C_{S_t} = (t, C_1, C_2)$, the DS-CRF performs the following steps:

1. Select a random value $\beta \in \mathbb{Z}^*_q$.
2. Compute $C'_1 = C_1 + \beta P = (r + \beta)P$.
3. Compute

$$C'_2 = C_2 \cdot \hat{e}(P'_{pub_t}, \beta Q_S) = M \cdot \hat{e}(P'_{pub_t}, Q_S)^{r+\beta}.$$

4. Output $C'_{S_t} = (t, C'_1, C'_2)$.

The DS-CRF sends C'_{S_t} to the CBSC and stores it in the CS.

7. ReKeyGen: Given the sender's epoch-dependent secret key SK'_{S_t} and the identities ID_S and ID_R, the data sender generates a proxy key as follows:

1. Compute $Q_R = H_1(ID_R)$.
2. Compute $\tau = H_2\big(\hat{e}(SK'_{S_t}, Q_R), ID_S, ID_R, t\big)$.
3. Output $rk_{S\to R} = \tau - SK'_{S_t}$, and send $rk_{S\to R}$ to the DS-CRF.

8. DS-CRF$_{ReKeyGen}$: Upon receiving $rk_{S\to R}$, the DS-CRF further randomizes it using β to obtain $rk'_{S\to R} = rk_{S\to R} - \beta Q_S$, sets $U = \beta Q_S$ as auxiliary re-encryption information and sends $(rk'_{S\to R}, U)$ to the CBSC.

9. ReEnc: Upon receiving $C'_{S_t} = (t, C'_1, C'_2)$, $rk'_{S\to R}$, and U, the CBSC smart contract re-encrypts the ciphertext as follows:

$$\begin{aligned}
C_R &= (t, C''_1, C''_2)\\
&= \big(t, C'_1,\ C'_2 \cdot \hat{e}(C'_1, rk'_{S\to R}) \cdot \hat{e}(C'_1, U)\big),\\
C''_2 &= C'_2 \cdot \hat{e}(C'_1, rk'_{S\to R}) \cdot \hat{e}(C'_1, U)\\
&= M \cdot \hat{e}(P'_{pub_t}, Q_S)^{r+\beta}\\
&\quad \cdot \hat{e}((r+\beta)P,\ rk_{S\to R} - \beta Q_S) \cdot \hat{e}((r+\beta)P,\ \beta Q_S)\\
&= M \cdot \hat{e}\big((r+\beta)P,\ SK'_{S_t}\big) \cdot \hat{e}((r+\beta)P,\ rk_{S\to R})\\
&\quad \cdot \hat{e}((r+\beta)P,\ -\beta Q_S) \cdot \hat{e}((r+\beta)P,\ \beta Q_S)\\
&= M \cdot \hat{e}\big((r+\beta)P,\ SK'_{S_t}\big) \cdot \hat{e}((r+\beta)P,\ rk_{S\to R})\\
&= M \cdot \hat{e}\big((r+\beta)P,\ SK'_{S_t}\big) \cdot \hat{e}\big((r+\beta)P,\ \tau - SK'_{S_t}\big)\\
&= M \cdot \hat{e}((r+\beta)P,\ \tau)\,.
\end{aligned}$$

The re-encrypted ciphertext $C_R = (t, C''_1, C''_2)$ is then sent to the data receiver.

10. Dec$_1$: If the data sender wishes to recover the message, the randomized ciphertext $C'_{S_t} = (t, C'_1, C'_2)$ is sent to the DS-CRF. The DS-CRF de-randomize C'_{S_t} using β as follows:

$$\begin{aligned}
C_1 &= C'_1 - \beta P,\\
C_2 &= C'_2 \cdot \hat{e}(P'_{pub_t}, -\beta Q_S),
\end{aligned}$$

and returns $C_{S_t} = (t, C_1, C_2)$ to the data sender. The sender then computes

$$M = \frac{C_2}{\hat{e}(C_1, SK'_{S_t})}.$$

11. Dec$_2$: Given $C_R = (t, C''_1, C''_2)$, the receiver secret key SK'_{R_t}, and the identities ID_S and ID_R, the decryption algorithm operates as follows:

1. Compute $Q_S = H_1(ID_S)$ and
$$\tau' = H_2\big(\hat{e}(SK'_{R_t}, Q_S), ID_S, ID_R, t\big).$$
2. Compute
$$M' = \frac{C''_2}{\hat{e}(C''_1, \tau')}.$$

If the ciphertext is valid, $M' = M$. Otherwise, the algorithm outputs $\perp$.

4.1 Correctness

Correctness of Dec$_1$:

$$\begin{aligned}
M &= \frac{C_2}{\hat{e}(C_1, SK'_{S_t})}\\
&= \frac{M \cdot \hat{e}(P'_{pub_t}, Q_S)^r}{\hat{e}(rP,\ \alpha_t s Q_S)}\\
&= \frac{M \cdot \hat{e}(\alpha_t s P, Q_S)^r}{\hat{e}(rP,\ \alpha_t s Q_S)}\\
&= \frac{M \cdot \hat{e}(rP,\ \alpha_t s Q_S)}{\hat{e}(rP,\ \alpha_t s Q_S)}\\
&= M.
\end{aligned}$$

Correctness of Dec$_2$:

$$\begin{aligned}
M' &= \frac{C''_2}{\hat{e}(C''_1, \tau')}\\
&= \frac{M \cdot \hat{e}((r+\beta)P,\ \tau)}{\hat{e}(C''_1, \tau')}\\
&= \frac{M \cdot \hat{e}((r+\beta)P,\ \tau)}{\hat{e}((r+\beta)P,\ \tau')}\\
&= \frac{M \cdot \hat{e}((r+\beta)P,\ \tau')}{\hat{e}((r+\beta)P,\ \tau')}\\
&= M.
\end{aligned}$$

Correctness under user revocation: For any epoch t, a user U with identity ID_U can successfully decrypt ciphertexts generated in epoch t if and only if $ID_U \notin RL_t$. If $ID_U \notin RL_t$, the PKG-CRF issues the epoch-dependent secret key SK'_{U_t}, which enables the user to correctly execute the decryption algorithms Dec$_1$ and Dec$_2$. If $ID_U \in RL_t$, the PKG-CRF does not issue SK'_{U_t}. As a result, the revoked user lacks the necessary secret key material to compute the pairing terms $\hat{e}(C_1, SK'_{U_t})$ or $\hat{e}(C''_1, \tau')$ and therefore cannot successfully decrypt the ciphertext. Therefore, the proposed scheme enforces correct access control by preventing revoked users from decrypting ciphertexts generated after their revocation.

5 Security Proof

In this section, we show that BRIBPR-CRF achieves confidentiality and resistance to

algorithm-substitution attacks, as shown in Theorems 1 and 2.

Theorem 1 (IND-sID-CPA Confidentiality): In the ROM, if there exists an adversary $\mathcal{A}$ that breaks the IND-sID-CPA confidentiality of the BRIBPR-CRF scheme with advantage ε in time t, after making at most q_{H_1} queries to H_1, q_{H_2} queries to H_2, q_{kg} *QKeyGen* queries, q_{rk} *QReKeyGen* queries, q_{enc} *QEnc* queries, and q_{re} *QReEnc* queries, then there exists an algorithm $\mathcal{C}$ that solves the DBDH problem with advantage

$$Adv_{\mathcal{C}}^{\mathrm{DBDH}} \geq \frac{\varepsilon}{2q_{H_1}},$$

within time

$$t' \leq t + O(q_{H_1} + q_{H_2} + q_{kg} + q_{rk} + q_{enc} + q_{re}) \cdot t_p,$$

where t_p is the time cost of one pairing. Figure 2 illustrates the IND-sID-CPA security proof.

Proof: We construct $\mathcal{C}$ that uses $\mathcal{A}$ to solve DBDH problem on input (P, aP, bP, cP, θ).

Selective challenge: Before the setup phase, $\mathcal{A}$ commits to the challenge identities (ID_S^*, ID_R^*) and the challenge epoch t^*.

Initialization: $\mathcal{C}$ is given a DBDH instance (P, aP, bP, cP, θ). It sets $P_{pub} = bP$. For each epoch $t \neq t^*$, $\mathcal{C}$ chooses $\alpha_t \in \mathbb{Z}_q^*$ and publishes $P'_{pub,t} = \alpha_t P_{pub} = \alpha_t bP$. For the challenge epoch t^*, $\mathcal{C}$ sets $P'_{pub,t^*} = bP$. Finally, $\mathcal{C}$ sends par to $\mathcal{A}$.

Phase 1: $\mathcal{C}$ simulates all IND-sID-CPA queries issued by $\mathcal{A}$ and maintains two lists $\mathcal{L}_{H_1}$ and $\mathcal{L}_{H_2}$ to record random oracle queries. $\mathcal{C}$ selects a random index $\ell \in \{1, \ldots, q_{H_1}\}$ and responds to queries as follows:

H_1 queries: For H_1 queries, $\mathcal{C}$ selects a random index $\ell \in \{1, \ldots, q_{H_1}\}$. For the ℓ-th distinct query $H_1(ID)$, if $ID = ID_S^*$, $\mathcal{C}$ sets $H_1(ID_S^*) = aP$. For all other identities, $\mathcal{C}$ chooses $\mu \in \mathbb{Z}_q^*$ and sets $H_1(ID) = \mu P$ and records the response in $\mathcal{L}_{H_1}$.

H_2 queries: For each query $H_2(h, ID_S, ID_R, t)$, $\mathcal{C}$ returns a uniformly random group element in $\mathbb{G}_1$ and records the response in $\mathcal{L}_{H_2}$.

QKeyGen(ID_U, t)*:* If $(ID_U, t) = (ID_R^*, t^*)$, $\mathcal{C}$ aborts. Otherwise, $\mathcal{C}$ retrieves $Q_U = H_1(ID_U)$ and returns $SK'_{U_t} = \alpha_t Q_U \cdot P_{pub}$.

QReKeyGen(ID_S, ID_R, t)*:* If $(ID_S, ID_R, t) = (ID_S^*, ID_R^*, t^*)$, then $\mathcal{C}$ aborts. Otherwise, $\mathcal{C}$ obtains SK'_{S_t} via *QKeyGen*(ID_S, t), sets $Q_S = H_1(ID_S)$ and $Q_R = H_1(ID_R)$, and computes $\tau = H_2\big(\hat{e}(SK'_{S_t}, Q_R), ID_S, ID_R, t\big)$, sets $rk_{S\to R} = \tau - SK'_{S_t}$, chooses $\beta \in \mathbb{Z}_q^*$, defines $U = \beta Q_S$, and computes $rk'_{S\to R} = rk_{S\to R} - U$. Finally, $\mathcal{C}$ stores $(ID_S, ID_R, t, rk'_{S\to R}, U)$ and returns $(rk'_{S\to R}, U)$ to $\mathcal{A}$.

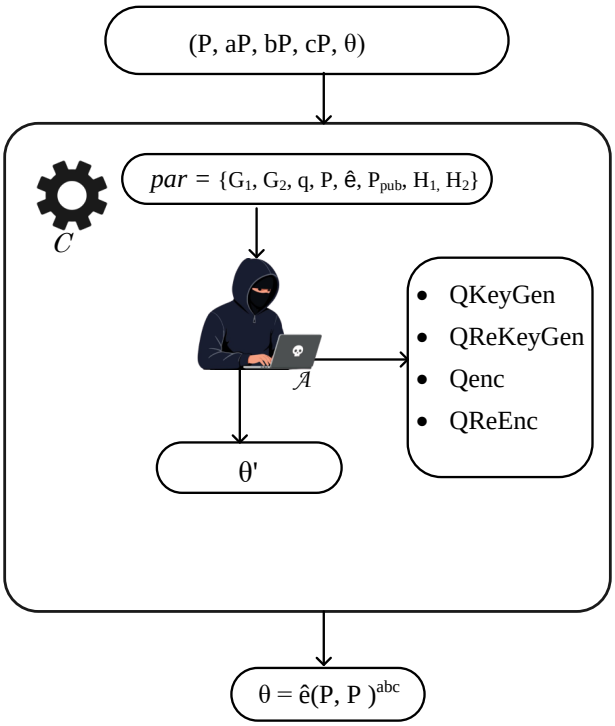

Figure 2. Security proof diagram for IND-sID-CPA.

QEnc(M, ID_S, t)*:* $\mathcal{C}$ runs *Enc*(M, ID_S, t) to obtain C_{S_t}, then runs *DS-CRF*$_{Enc}$ to obtain C'_{S_t}, and returns C'_{S_t} to $\mathcal{A}$.

QReEnc$(C'_{S_t}, rk'_{S\to R}, U)$*:* $\mathcal{C}$ runs *ReEnc*$(C'_{S_t}, rk'_{S\to R}, U)$ to obtain C_R and returns C_R to $\mathcal{A}$.

Challenge: $\mathcal{A}$ selects two equal-length messages (m_0, m_1) and identities (ID_S^*, ID_R^*) for the challenge t^*. If $ID_R^* \in RL_{t^*}$, then $\mathcal{C}$ aborts. Otherwise, $\mathcal{C}$ chooses a random bit $b \in \{0, 1\}$ and sets $C_1^* = cP$, $C_2^* = m_b \cdot \theta$, forming the ciphertext $C_{S_{t^*}}^* = (t^*, C_1^*, C_2^*)$. Next, $\mathcal{C}$ forwards $C_{S_{t^*}}^*$ to the DS-CRF, which executes *DS-CRF*$_{Enc}$. $\mathcal{C}$ then runs *ReKeyGen* and *DS-CRF*$_{ReKeyGen}$ to compute $rk'_{S\to R}$, and finally executes *ReEnc* to obtain $C_{R_{t^*}}^*$, which is returned to $\mathcal{A}$.

Phase 2: $\mathcal{A}$ continues to issue valid queries, and $\mathcal{C}$ answers them as in *Phase 1*.

Guess: Finally, $\mathcal{A}$ outputs a bit b'. If $b' = b$, $\mathcal{C}$ outputs 1; otherwise, it outputs 0. This event occurs provided that the following events hold: E_1: The challenge sender identity ID_S^*

is the ℓ-th distinct query to $H_1(\cdot)$, where $\ell \in \{1, \ldots, q_{H_1}\}$. E_2: $\mathcal{A}$ never queries $QKeyGen(ID_R^*, t^*)$ nor $QReKeyGen(ID_S^*, ID_R^*, t^*)$. Since ℓ is chosen uniformly, we have

$$\Pr[E_1] = \frac{1}{q_{H_1}}.$$

and by the IND-sID-CPA game rules, $\Pr[E_2] = 1$.

Conditioned on E_1 and E_2, when $\theta = \hat{e}(P, P)^{abc}$ the challenge ciphertext is distributed identically to a real encryption; when $\theta \xleftarrow{\$} \mathbb{G}_2$, it is independent of b. let

$$p_1 = \Pr[b' = b \mid \theta = \hat{e}(P, P)^{abc}] = \frac{1}{2} + \frac{\varepsilon}{2},$$

and

$$p_0 = \Pr[b' = b \mid \theta \xleftarrow{\$} \mathbb{G}_2] = \frac{1}{2}.$$

Therefore, the advantage of $\mathcal{C}$ in solving the DBDH problem is

$$\varepsilon_{\mathrm{DBDH}} = |p_1 - p_0| \cdot \Pr[E_1] \Pr[E_2] = \frac{\varepsilon}{2} \cdot \frac{1}{q_{H_1}} = \frac{\varepsilon}{2q_{H_1}}.$$

Theorem 2 (CRF Security and ASA Resistance): The BRIBPR-CRF scheme employs CRFs at both the PKG and the data sender to achieve (i) functionality preservation, (ii) weak security preservation, and (iii) weak exfiltration resistance against ASAs as defined in the game $\mathsf{G}_{\mathsf{LEAK}}$.

Proof: We show that the BRIBPR-CRF scheme satisfies the above three properties as follows.

(i) Functionality Preservation: *Original ciphertext.* Given an original ciphertext $C_{S_t} = (t, C_1, C_2)$ generated in epoch t, the data sender recovers the message using its epoch-specific secret key $SK'_{S_t} = \alpha_t s Q_S$ as follows:

$$\begin{aligned} M &= \frac{C_2}{\hat{e}(C_1, SK'_{S_t})} \\ &= \frac{M \cdot \hat{e}(P'_{pub,t}, Q_S)^r}{\hat{e}(rP,\ \alpha_t s Q_S)} \\ &= \frac{M \cdot \hat{e}(\alpha_t s P,\ Q_S)^r}{\hat{e}(rP,\ \alpha_t s Q_S)} \\ &= \frac{M \cdot \hat{e}(rP,\ \alpha_t s Q_S)}{\hat{e}(rP,\ \alpha_t s Q_S)} \\ &= M. \end{aligned}$$

Re-encrypted ciphertext. Given a re-encrypted ciphertext $C_R = (t, C_1'', C_2'')$, the receiver decrypts using SK'_{R_t} as follows. It first computes

$$\begin{aligned} Q_S &= H_1(ID_S), \\ \tau' &= H_2\big(\hat{e}(SK'_{R_t}, Q_S), ID_S, ID_R, t\big), \end{aligned}$$

and then outputs

$$\begin{aligned} M' &= \frac{C_2''}{\hat{e}(C_1'', \tau')} \\ &= \frac{M \cdot \hat{e}((r+\beta)P,\ \tau)}{\hat{e}((r+\beta)P,\ \tau')}, \end{aligned}$$

where

$$\tau = H_2\big(\hat{e}(SK'_{S_t}, Q_R), ID_S, ID_R, t\big).$$

Since $SK'_{S_t} = \alpha_t s Q_S$ and $SK'_{R_t} = \alpha_t s Q_R$, we have

$$\begin{aligned} \hat{e}(SK'_{S_t}, Q_R) &= \hat{e}(\alpha_t s Q_S,\ Q_R) = \hat{e}(Q_S,\ Q_R)^{\alpha_t s} \\ &= \hat{e}(Q_S,\ \alpha_t s Q_R) = \hat{e}(Q_S,\ SK'_{R_t}) \\ &= \hat{e}(SK'_{R_t},\ Q_S). \end{aligned}$$

Hence $\tau = \tau'$, and therefore

$$M' = \frac{M \cdot \hat{e}((r+\beta)P,\ \tau')}{\hat{e}((r+\beta)P,\ \tau')} = M.$$

Therefore, the outputs processed by the PKG-CRF and DS-CRF preserve the correctness of both original decryption and re-encryption decryption. Hence, functionality preservation holds.

(ii) Weak Security Preservation under ASA: To analyze the resistance of the BRIBPR-CRF scheme against ASAs under the game $\mathsf{G}_{\mathsf{LEAK}}$, we define the following hybrid games.

Game 0: The real execution of BRIBPR-CRF under ASA, where potentially modified algorithms are executed but all outputs are filtered by the PKG-CRF and DS-CRF.

Game 1: Identical to Game 0, except that SK_U and SK'_{U_t} are generated by the honest *KeyGen* and *PKG-CRF* procedures, without ASAs influence.

Game 2: Identical to Game 1, except that $rk_{S \to R}$ is generated by the honest *ReKeyGen* algorithm prior to DS-CRF filtering.

Game 3: Identical to Game 2, except that C_{S_t} is generated by the honest *Enc* algorithm prior to DS-CRF filtering.

Game 4: Identical to Game 3, except that the re-encrypted ciphertext C_R is generated by the honest *ReEnc* algorithm.

Table 2. Comparison of computational costs.

Scheme	Enc	ReKeyGen	ReEnc	Dec
IBP [27]	5M + 6E + 3P	5M + 5E + 2P	M + P	2M + E + 4P
IPV2 [32]	2M + 6E + P	4M + 7E + P	3M + E + 8P	6M + 6E + 8P
CL-PRE [33]	M + 3E + P	2M + 6E + P	3M + 6E + 2P	2M + 2P
IBPRE [34]	2M + 2E + P	2M + 3E + P	M + 4P	2M + 2P
MH-IBPRE [35]	7M + 4E + 2p	7M + 3E	M + 2E + P	3M + E + 2P
BRIBPR-CRF	3M + E + 2P	M + P	M + 2P	2P

Table 3. Comparison of communication cost and security.

Schemes	Communication cost			Security		
	RKG	C_{S_t}	C_R	IND-CPA	ASA	ReVoKe
IBP [27]	$2\|\mathbb{G}_1\| + \|\mathbb{G}_2\|$	$\|\mathbb{G}_1\| + \|\mathbb{G}_2\|$	$2\|\mathbb{G}_1\| + \|\mathbb{G}_2\|$	✓	✓	×
IPV2 [32]	$5\|\mathbb{G}_1\|+\|\mathbb{G}_2\|+\|\mathbb{Z}_q^*\|$	$4\|\mathbb{G}_1\|+\|\mathbb{G}_2\|+\|\mathbb{Z}_q^*\|$	$5\|\mathbb{G}_1\| + 2\|\mathbb{G}_2\| + 2\|\mathbb{Z}_q^*\|$	✓	×	×
CL-PRE [33]	$3\|\mathbb{G}_1\| + \|\mathbb{G}_2\|$	$2\|\mathbb{G}_1\| + \|\mathbb{G}_2\|$	$3\|\mathbb{G}_1\|+2\|\mathbb{G}_2\|$	✓	×	×
IBPRE [34]	$2n\|\mathbb{G}_1\|+\|\mathbb{G}_2\|$	$n\|\mathbb{G}_1\| + \|\mathbb{G}_2\|$	$2n\|\mathbb{G}_1\| + 2\|\mathbb{G}_2\|$	✓	×	×
MH-IBPRE [35]	$\|\mathbb{G}_1\| + \|\mathbb{Z}_q^*\|$	$2\|\mathbb{G}_1\|+2\|\mathbb{G}_2\|$	$2\|\mathbb{G}_1\|+2\|\mathbb{G}_2\|$	✓	×	×
BRIBPR-CRF	$\|\mathbb{G}_2\|$	$\|\mathbb{G}_1\| + \|\mathbb{G}_2\|$	$\|\mathbb{G}_1\| + \|\mathbb{G}_2\|$	✓	✓	✓

Next, we show that each adjacent pair of games is computationally indistinguishable.

Game 0 ≈ Game 1: These games differ only in the generation of secret keys. The PKG-CRF applies randomization to the secret keys, which preserves their correct distribution while removing any ASA-embedded leakage. Therefore, the views in the two games are computationally indistinguishable.

Game 1 ≈ Game 2: These games differ only in the generation of proxy re-encryption keys. The DS-CRF re-randomizes the proxy key, making its distribution independent of any ASAs behavior. Therefore, the views are computationally indistinguishable.

Game 2 ≈ Game 3: These games differ only in ciphertext generation. The DS-CRF applies additive re-randomization to ciphertext components while preserving correctness and their distribution. Thus the views are computationally indistinguishable.

Game 3 ≈ Game 4: These games differ only in the re-encryption step. Since the re-encrypted ciphertext distribution is preserved under CRF filtering, the resulting views are computationally indistinguishable.

By a standard hybrid argument, Game 0 and Game 4 are computationally indistinguishable. Therefore, replacing the honest algorithms with functionality-preserving modified algorithms does not give the adversary any non-negligible advantage. Hence, weak security preservation holds.

(iii) Weak Exfiltration Resistance: Since the real execution under ASAs (Game 0) is computationally indistinguishable from the ideal execution without ASA influence (Game 4), any leakage channel embedded by a modified algorithm is neutralized by the CRF filtering process. Consequently, no efficient adversary can exfiltrate information through algorithm substitution. Therefore, BRIBPR-CRF achieves weak exfiltration resistance in the sense of $\mathsf{G}_{\mathsf{LEAK}}$.

6 Performance Analysis

In this section, the proposed BRIBPR-CRF scheme is compared with IBP [27], IPV2 [32], CL-PRE [33], IBPRE [34], and MH-IBPRE [35] in terms of computational cost, communication overhead, energy consumption, end-to-end latency, and security

properties, as summarized in Tables 2 and 3. For communication overhead, RKG denotes the proxy key, C_{S_t} denotes the ciphertext generated by the data sender at epoch t, and C_R denotes the re-encrypted ciphertext that can only be decrypted by an authorized receiver. The symbols $|\mathbb{G}_1|$, $|\mathbb{G}_2|$, and $|\mathbb{Z}_q^*|$ represent the bit lengths of elements in $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{Z}_q^*$, respectively. To evaluate computational cost, we use the notation in Table 2, where M denotes scalar multiplication in $\mathbb{G}_1$, E denotes exponentiation in $\mathbb{G}_2$, and P denotes bilinear pairing operations. Since these three operations dominate the execution time, the costs of lightweight operations are omitted. As shown in Table 3, the communication cost of the IBPRE scheme depends on a parameter n. To enable a fair comparison, we set $n = 1$ in our analysis.

In Table 3, the symbol ✓ indicates that a scheme satisfies the corresponding security property, whereas $\times$ denotes its absence. For communication cost, $|x|$ denotes the bit length of an element x. From Table 3, it can be observed that only IBP [27] and the proposed scheme achieve resistance to ASAs. However, IBP [27] does not support revocation and incurs higher communication and computational overhead compared to BRIBPR-CRF. In terms of communication overhead, the proposed BRIBPR-CRF scheme produces shorter proxy keys and re-encrypted ciphertexts than the other schemes, as illustrated in Figure 4. Moreover, its original ciphertext size is comparable to those of IBP [27] and IBPRE [34], and is smaller than those of the remaining schemes. Nevertheless, IBP [27] and IBPRE [34] fail to provide resistance to ASAs and to support epoch-based revocation, and incur higher communication and computational overhead than the proposed scheme.

The experiment was conducted using Type A bilinear pairings implemented with the PBC library [36] on a desktop computer equipped with an Intel Core i5-13600KF 3.50 GHz processor, an NVIDIA GeForce RTX 3090 GPU, and 64 GB RAM. Type A pairings are constructed over the elliptic curve $y^2 = x^3 + x \bmod q$, where q is a prime satisfying $q \equiv 3 \pmod 4$ and the order of $\mathbb{G}_1$ is a prime p. According to [26], the average execution time of a scalar multiplication in $\mathbb{G}_1$ is approximately 6.38 ms, an exponentiation in $\mathbb{G}_2$ requires approximately 11.20 ms, and a bilinear pairing operation requires approximately 20.01 ms. The resulting computational cost comparison for all schemes is summarized in Table 2 and illustrated in Figure 3.

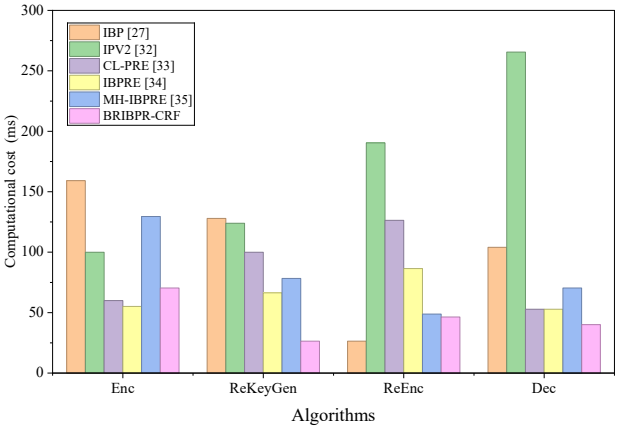

Figure 3. Comparison of computational cost.

For communication cost evaluation, the message length is set to $|m| = 160$ bits, and an 80-bit security level is adopted [37], resulting in a 512-bit prime q. Thus, an element of $\mathbb{G}_1$ is 65 bytes with point compression [25, 38, 41], an element of $\mathbb{G}_2$ is 128 bytes, and an element of $\mathbb{Z}_q^*$ is 64 bytes. Based on Table 3, the communication costs of the six schemes are compared as follows:

1. IBP [27]: the proxy key size is $2|\mathbb{G}_1| + |\mathbb{G}_2| = 2 \times 65 + 128 = 258$ bytes, the original ciphertext size is $|\mathbb{G}_1| + |\mathbb{G}_2| = 65 + 128 = 193$ bytes, and the re-encrypted ciphertext size is $2|\mathbb{G}_1| + |\mathbb{G}_2| = 258$ bytes.

2. IPV2 [32]: the proxy key size is $5|\mathbb{G}_1| + |\mathbb{G}_2| + |\mathbb{Z}_q^*| = 5 \times 65 + 128 + 64 = 517$ bytes, the original ciphertext size is $4|\mathbb{G}_1| + |\mathbb{G}_2| + |\mathbb{Z}_q^*| = 4 \times 65 + 128 + 64 = 452$ bytes, and the re-encrypted ciphertext size is $5|\mathbb{G}_1| + 2|\mathbb{G}_2| + 2|\mathbb{Z}_q^*| = 5 \times 65 + 2 \times 128 + 2 \times 64 = 709$ bytes.

3. CL-PRE [33]: the proxy key size is $3|\mathbb{G}_1| + |\mathbb{G}_2| = 3 \times 65 + 128 = 323$ bytes, the original ciphertext size is $2|\mathbb{G}_1| + |\mathbb{G}_2| = 2 \times 65 + 128 = 258$ bytes, and the re-encrypted ciphertext size is $3|\mathbb{G}_1| + 2|\mathbb{G}_2| = 3 \times 65 + 2 \times 128 = 451$ bytes.

4. IBPRE [34]: the proxy key size is $2n|\mathbb{G}_1| + |\mathbb{G}_2|$ bytes, the original ciphertext size is $n|\mathbb{G}_1| + |\mathbb{G}_2|$ bytes, and the re-encrypted ciphertext size is $2n|\mathbb{G}_1| + 2|\mathbb{G}_2|$ bytes. For $n = 1$, these sizes are 258, 193, and 386 bytes, respectively.

5. MH-IBPRE [35]: the proxy key size is $|\mathbb{G}_1| + |\mathbb{Z}_q^*| = 65 + 64 = 129$ bytes, the original ciphertext size is $2|\mathbb{G}_1| + 2|\mathbb{G}_2| = 2 \times 65 + 2 \times 128 = 386$ bytes, and the re-encrypted ciphertext size is $2|\mathbb{G}_1| + 2|\mathbb{G}_2| = 386$ bytes.

6. BRIBPR-CRF, the proxy size is $|\mathbb{G}_2| = 128$ bytes, the original ciphertext size is $|\mathbb{G}_1| + |\mathbb{G}_2| = 65 + 128 = 193$ bytes, and the re-encrypted ciphertext size is $|\mathbb{G}_1| + |\mathbb{G}_2| = 193$ bytes.

The communication costs of the six schemes are summarized in Table 3. As illustrated in Figure 4, the proposed BRIBPR-CRF scheme produces shorter proxy keys and re-encrypted ciphertexts than the other schemes. Moreover, its original ciphertext size is comparable to those of IBP [27] and IBPRE [34], and smaller than those of the remaining schemes. Nevertheless, IBP [27] and IBPRE [34] fail to provide resistance against ASAs or support epoch-based revocation, and incur higher communication and computational overhead compared with the proposed scheme. Overall, the proposed scheme achieves stronger security guarantees with lower communication and computational overhead, making it well-suited for resource-constrained and dynamic environments.

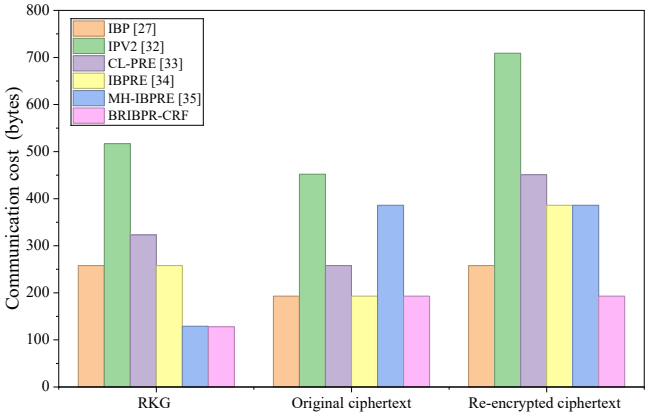

Figure 4. Comparison of communication cost.

For energy consumption analysis, a scalar multiplication consumes approximately 19.1 mJ, an exponentiation consumes 21.6 mJ, and a bilinear pairing consumes 45.6 mJ [24, 25, 41, 42]. Based on the computational costs summarized in Table 3, the IBP [27], IPV2 [32], CL-PRE [33], IBPRE [34], MH-IBPRE [35], and the proposed BRIBPR-CRF schemes consume approximately 1215.2 mJ, 1417.6 mJ, 947.3 mJ, 753.6 mJ, 1182.4 mJ, and 460.7 mJ, respectively, for a complete proxy re-encryption process. As a result, the proposed BRIBPR-CRF scheme exhibits substantially lower energy consumption than existing schemes. Figure 5 further illustrates the energy consumption comparison, clearly showing that the proposed BRIBPR-CRF scheme achieves the lowest energy consumption among the compared schemes and is therefore suitable for resource-constrained and dynamic environments.

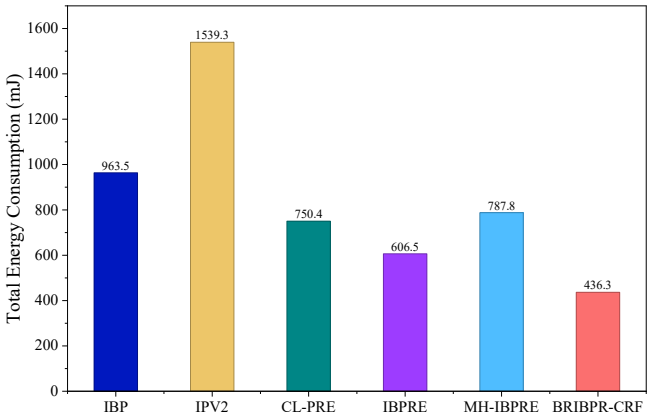

Figure 5. Comparison of total energy consumption.

6.1 End-to-End Latency and Blockchain Overhead Analysis

In addition to the computational cost of cryptographic operations, it is necessary to evaluate the practical delay introduced by the blockchain-assisted architecture. Therefore, we further analyze the end-to-end latency of secure data sharing by considering not only encryption and proxy re-encryption costs, but also the delay caused by blockchain transaction submission, smart-contract execution, and confirmation. Let the total end-to-end latency of the proposed scheme be defined as

$$T_{\mathrm{E2E}} = T_{\mathrm{Enc}} + T_{\mathrm{RKG}} + T_{\mathrm{RE}} + T_{\mathrm{BC}} + T_{\mathrm{Dec}},$$

where T_{Enc} denotes the sender-side encryption time, T_{RKG} denotes the proxy key generation time, T_{RE} denotes the proxy re-encryption time, T_{BC} denotes the blockchain-related delay, and T_{Dec} denotes the receiver-side decryption time. The blockchain-related delay can be further expressed as

$$T_{\mathrm{BC}} = T_{\mathrm{sub}} + T_{\mathrm{exec}} + T_{\mathrm{con}},$$

where T_{sub} is the transaction submission delay, T_{exec} is the smart-contract execution time, and T_{con} is the block confirmation latency. To reduce blockchain storage cost and improve scalability, the proposed BRIBPR-CRF scheme stores only lightweight metadata on-chain, such as ciphertext indices, identity bindings, revocation status, policy references, and integrity hashes, while the encrypted data itself is maintained off-chain. This design reduces on-chain overhead

while preserving traceability, auditability, and tamper resistance.

For performance evaluation, the end-to-end latency can be measured by deploying the smart contract in a local or consortium blockchain environment and recording the average delay over multiple runs. The main contract functions include sender registration, receiver registration, revocation update, ciphertext metadata submission, and access verification. The corresponding latency components include transaction submission delay, smart-contract execution time, block confirmation latency, and the total end-to-end latency. The above metrics provide a more realistic evaluation of the practical efficiency of the proposed scheme. In particular, while the blockchain layer introduces additional delay compared with cryptographic schemes, it removes dependence on a single centralized controller for metadata management and revocation logging, thereby improving transparency, availability, and auditability. Therefore, the blockchain-assisted design does not merely rely on theoretical advantages, but can be quantitatively evaluated in terms of its latency and on-chain execution overhead.

7 Conclusion

In this paper, an efficient and secure data-sharing BRIBPR-CRF scheme for decentralized environments is proposed. The scheme integrates CRFs into the key generation, encryption, and proxy key algorithms to effectively mitigate ASA attacks and data exfiltration threats even in the presence of compromised cryptographic algorithms. An epoch-based revocation mechanism is further introduced to enable efficient user revocation without requiring global key updates. In addition, smart contracts deployed on the blockchain replace semi-trusted proxy servers and eliminate single points of failure by performing re-encryption operations. Formal security analysis demonstrates that BRIBPR-CRF achieves IND-sID-CPA security and resilience against ASAs under standard cryptographic assumptions in the ROM. Experimental results indicate that the proposed scheme significantly reduces communication overhead, computation time, and energy consumption compared with existing approaches. These advantages make BRIBPR-CRF more suitable for secure, scalable data sharing in decentralized, semi-trusted environments. For future work, we plan to integrate BRIBPR-CRF with edge intelligence and 5G/6G networks, along with lightweight cryptographic optimizations, to support large-scale IoT and cloud–edge systems.

Data Availability Statement

Data will be made available on request.

Funding

This work was supported in part by the National Foreign Expert Program under Grant Y20250135; in part by the National Key Research and Development Program of China under Grant 2022YFB2701400; in part by the Sichuan Science and Technology Support Plan under Grant 2024NSFTD0005.

Conflicts of Interest

The authors declare no conflicts of interest.

AI Use Statement

The authors declare that no generative AI was used in the preparation of this manuscript.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Chen, J., Wang, M., Cao, Z., Dong, X., & Sun, L. (2025). Secure and controllable cloud–edge collaborative data sharing scheme for wireless body area networks in IIoT. *Computers & Security, 153*, 104389. [CrossRef]

[2] Bayat, M., Jamali, M. A. J., Abbasi, M., Anari, B., & Akbarpour, S. (2025). Enhancing secure IoT data sharing through dynamic Q-learning and blockchain at the edge. *Scientific Reports, 15*(1), 39153. [CrossRef]

[3] Shang, F., & Deng, X. (2025). A data sharing scheme based on blockchain for privacy protection certification of Internet of Vehicles. *Vehicular Communications, 51*, 100864. [CrossRef]

[4] Zhang, Q., Yuan, L., Xie, T., & Chen, H. (2024). Auditable and dynamic access control scheme with behavior and identity tracing. *Computer Networks, 251*, 110623. [CrossRef]

[5] Blaze, M., Bleumer, G., & Strauss, M. (1998, May). Divertible protocols and atomic proxy cryptography. In *International conference on the theory and applications of cryptographic techniques* (pp. 127-144). Berlin, Heidelberg: Springer Berlin Heidelberg. [CrossRef]

[6] Wang, F., Cui, J., Zhang, Q., He, D., Gu, C., & Zhong, H. (2023). Lightweight and secure data sharing based on proxy re-encryption for blockchain-enabled industrial internet of things. *IEEE Internet of Things Journal, 11*(8), 14115-14126. [CrossRef]

[7] Tang, Q., & Yung, M. (2017, October). Cliptography: Post-snowden cryptography. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security* (pp. 2615-2616). [CrossRef]

[8] Bellare, M., Paterson, K. G., & Rogaway, P. (2014, August). Security of symmetric encryption against mass surveillance. In *Annual cryptology conference* (pp. 1-19). Berlin, Heidelberg: Springer Berlin Heidelberg. [CrossRef]

[9] Mironov, I., & Stephens-Davidowitz, N. (2015, April). Cryptographic reverse firewalls. In *Annual international conference on the theory and applications of cryptographic techniques* (pp. 657-686). Berlin, Heidelberg: Springer Berlin Heidelberg. [CrossRef]

[10] Zhang, Q., Fu, Y., Cui, J., He, D., & Zhong, H. (2024). Efficient fine-grained data sharing based on proxy re-encryption in iiot. *IEEE Transactions on Dependable and Secure Computing, 21*(6), 5797–5809. [CrossRef]

[11] Lv, Y., Li, X., Wang, Y., Chen, K., Hou, Z., & Feng, R. (2024). Cross-chain sharing of personal health records: Heterogeneous and interoperable blockchains. In *2024 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (pp. 3588–3591). IEEE. [CrossRef]

[12] Zhang, J., Su, S., Zhong, H., Cui, J., & He, D. (2023). Identity-based broadcast proxy re-encryption for flexible data sharing in VANETs. *IEEE Transactions on Information Forensics and Security, 18*, 4830–4842. [CrossRef]

[13] Green, M., & Ateniese, G. (2007, June). Identity-based proxy re-encryption. In *International Conference on Applied Cryptography and Network Security* (pp. 288-306). Berlin, Heidelberg: Springer Berlin Heidelberg. [CrossRef]

[14] Cohen, A. (2019, April). What about bob? The inadequacy of CPA security for proxy reencryption. In *IACR International Workshop on Public Key Cryptography* (pp. 287-316). Cham: Springer International Publishing. [CrossRef]

[15] Susilo, W., Dutta, P., Duong, D. H., & Roy, P. S. (2021, October). Lattice-based HRA-secure attribute-based proxy re-encryption in standard model. In *European Symposium on Research in Computer Security* (pp. 169-191). Cham: Springer International Publishing. [CrossRef]

[16] Zhao, F., Weng, J., Xie, W., Li, M., & Weng, J. (2024). HRA-secure attribute-based threshold proxy re-encryption from lattices. *Information Sciences, 655*, 119900. [CrossRef]

[17] Ge, C., Susilo, W., Liu, Z., Baek, J., Luo, X., & Fang, L. (2023, July). Attribute-based proxy re-encryption with direct revocation mechanism for data sharing in clouds. In *Proceedings of the ACM Turing Award Celebration Conference-China 2023* (pp. 164-165). [CrossRef]

[18] Luo, F., Al-Kuwari, S., Susilo, W., & Duong, D. H. (2020). Chosen-ciphertext secure homomorphic proxy re-encryption. *IEEE Transactions on Cloud Computing, 10*(4), 2398–2408. [CrossRef]

[19] Manzoor, A., Liyanage, M., Braeke, A., Kanhere, S. S., & Ylianttila, M. (2019). Blockchain based proxy re-encryption scheme for secure IoT data sharing. In *2019 IEEE international conference on blockchain and cryptocurrency (ICBC)* (pp. 99–103). IEEE. [CrossRef]

[20] Jin, C., Chen, Z., Qin, W., Sun, K., Chen, G., & Chen, L. (2024). A Blockchain-Based Proxy Re-Encryption Scheme With Cryptographic Reverse Firewall for IoV. *International Journal of Network Management, 34*(6), e2305. [CrossRef]

[21] Lin, Z., Zhou, J., Cao, Z., Dong, X., & Choo, K. K. R. (2023). Generalized autonomous path proxy re-encryption scheme to support branch functionality. *IEEE Transactions on Information Forensics and Security, 18*, 5387-5400. [CrossRef]

[22] Zhou, Y., Guo, J., & Li, F. (2020). Certificateless public key encryption with cryptographic reverse firewalls. *Journal of Systems Architecture, 109*, 101754. [CrossRef]

[23] Ma, H., Zhang, R., Yang, G., Song, Z., Sun, S., & Xiao, Y. (2018). Concessive online/offline attribute based encryption with cryptographic reverse firewalls—Secure and efficient fine-grained access control on corrupted machines. In *European symposium on research in computer security* (pp. 507–526). Springer. [CrossRef]

[24] Li, F., & Xiong, P. (2013). Practical secure communication for integrating wireless sensor networks into the internet of things. *IEEE Sensors Journal, 13*(10), 3677–3684. [CrossRef]

[25] Hundera, N. W., Aftab, M. U., Mesfin, D., Dioubi, F., Xu, H., & Zhu, X. (2024). An efficient heterogeneous online/offline anonymous certificateless signcryption with proxy re-encryption for Internet of Vehicles. *Vehicular Communications, 49*, 100811. [CrossRef]

[26] Hundera, N. W., Mei, Q., Xiong, H., & Geressu, D. M. (2020). A secure and efficient identity-based proxy signcryption in cloud data sharing. *KSII Transactions on Internet & Information Systems, 14*(1). [CrossRef]

[27] Zhou, Y., Zhao, L., Jin, Y., & Li, F. (2022). Backdoor-resistant identity-based proxy re-encryption for cloud-assisted wireless body area networks. *Information Sciences, 604*, 80–96. [CrossRef]

[28] Hundera, N. W., Jin, C., Geressu, D. M., Aftab, M. U., Olanrewaju, O. A., & Xiong, H. (2022). Proxy-based public-key cryptosystem for secure and efficient IoT-based cloud data sharing in the smart city. *Multimedia Tools and Applications, 81*(21), 29673–29697. [CrossRef]

[29] Li, Y., Chen, R., & Rahmani, R. (2023, August). Secure data sharing in internet of vehicles based on blockchain and attribute-based encryption. In *2023 IEEE International Conference on Smart Internet of Things (SmartIoT)* (pp. 56-63). IEEE. [CrossRef]

[30] Sanchol, P., & Fugkeaw, S. (2023). A fully outsourced

attribute-based signcryption scheme supporting privacy-preserving policy update in mobile cloud computing. *IEEE Access*, *11*, 145915-145930. [CrossRef]

[31] Ateniese, G., Fu, K., Green, M., & Hohenberger, S. (2006). Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Transactions on Information and System Security (TISSEC)*, *9*(1), 1–30. [CrossRef]

[32] Sun, M., Ge, C., Fang, L., & Wang, J. (2018). A proxy broadcast re-encryption for cloud data sharing. *Multimedia Tools and Applications*, *77*(9), 10455–10469. [CrossRef]

[33] Xu, L., Wu, X., & Zhang, X. (2012). CL-PRE: a certificateless proxy re-encryption scheme for secure data sharing with public cloud. In *Proceedings of the 7th ACM symposium on information, computer and communications security* (pp. 87–88). [CrossRef]

[34] Wang, Z. (2018). Leakage resilient ID-based proxy re-encryption scheme for access control in fog computing. *Future Generation Computer Systems*, *87*, 679–685. [CrossRef]

[35] Luo, S., Shen, Q., & Chen, Z. (2011, November). Fully secure unidirectional identity-based proxy re-encryption. In *International Conference on Information Security and Cryptology* (pp. 109-126). Berlin, Heidelberg: Springer Berlin Heidelberg. [CrossRef]

[36] Lynn, B. (2007). Pbc library-pairing-based cryptography. *http://crypto. stanford. edu/pbc/*.

[37] Daemen, J., & Rijmen, V. (2002). Related block ciphers. In *The Design of Rijndael: AES—The Advanced Encryption Standard* (pp. 161-173). Berlin, Heidelberg: Springer Berlin Heidelberg. [CrossRef]

[38] Lo, N.-W., & Tsai, J.-L. (2015). An efficient conditional privacy-preserving authentication scheme for vehicular sensor networks without pairings. *IEEE Transactions on Intelligent Transportation Systems*, *17*(5), 1319–1328. [CrossRef]

[39] Fan, K., Ren, Y., Wang, Y., Li, H., & Yang, Y. (2018). Blockchain-based efficient privacy preserving and data sharing scheme of content-centric network in 5G. *IET Communications*, *12*(5), 527–532. [CrossRef]

[40] Buterin, V. (2014). A next-generation smart contract and decentralized application platform. *white paper*, *3*(37), 2-1.

[41] Wu, T. Y., Wu, H., Tang, M., Kumari, S., & Chen, C. M. (2025). CD-AKA-IoV: A Provably Secure Cross-Domain Authentication and Key Agreement Protocol for Internet of Vehicle. *Computers, Materials & Continua*, *85*(1). [CrossRef]

[42] Chinnadurai, G., & Nagarajan, S. (2026). Development of advanced sensor materials and encryption techniques for secure Wireless Body Area Networks (WBANs) in healthcare applications. *Matéria (Rio de Janeiro)*, *31*, e20250390. [CrossRef]

Negalign Wake Hundera received the B.S. degree from the Faculty of Engineering and Technology, Jimma University, Jimma, Ethiopia, in 2009, the master's degree from the School of Software Engineering, University of Electronic Science and Technology of China, Chengdu, China, in 2016, and the Ph.D. degree from the School of Software Engineering, University of Electronic Science and Technology of China, Chengdu, China, in 2021. He is currently a Postdoctoral Researcher with the School of Software Engineering, University of Electronic Science and Technology of China. His current research interests include cryptography and network security. (Email: nigaccna21@uestc.edu.cn)

Rashad Elhabob received the B.S. degree from the Faculty of Computer Science and Information Technology, Karary University, Khartoum, Sudan, in 2010, the master's degree from the Faculty of Mathematical Science, University of Khartoum, Khartoum, in 2014, and the Ph.D. degree from the School of Software Engineering, University of Electronic Science and Technology of China, Chengdu, China, in 2019. He is currently a Postdoctoral Researcher with the School of Software Engineering, University of Electronic Science and Technology of China. His current research interests include cryptography and network security. (Email: rashad@uestc.edu.cn)

Deepak Adhikari received the Ph.D. degree from the University of Electronic Science and Technology of China (UESTC), Chengdu, China. He is currently a Postdoctoral Research Fellow with UESTC. His research interests include cybersecurity, privacy preservation, the Internet of Things (IoT), federated reinforcement learning, artificial intelligence, and secure and autonomous decision-making in distributed systems. He has received multiple national research grants, including funding from the National Foreign Talent Individual Program and the National Natural Science Foundation of China (NSFC), and has served as Principal Investigator and participant in projects on neural network security, attack detection, privacy-preserving mechanisms, and fault detection. He serves as an Editor for the ICCK Transactions on Advanced Computing Systems and Machine Intelligence and has published more than 30 articles in peer-reviewed journals and international conferences. (Email: deepakadhikari@uestc.edu.cn)

Hu Xiong received the Ph.D. degree from the School of Computer Science and Engineering, University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2009. He is currently a Full Professor with the School of Information and Software Engineering, UESTC. His research interests include applied cryptography and cyberspace security. (Email: xionghu.uestc@gmail.com)