information

MDPI

Article

A Blockchain-Based Efficient and Verifiable Attribute-Based Proxy Re-Encryption Cloud Sharing Scheme

Tao Feng *, Dewei Wang and Renbin Gong

School of Computer and Communication, Lanzhou University of Technology, Lanzhou 730050, China
* Correspondence: fengt@lut.edu.cn

Abstract: When choosing a third-party cloud storage platform, the confidentiality of data should be the primary concern. To address the issue of one-to-many access control during data sharing, it is important to encrypt data with an access policy that enables fine-grained access. The attribute-based encryption scheme can be used for this purpose. Additionally, attribute-based proxy re-encryption (ABPRE) can generate a secret key using the delegatee's secret key and access policy to re-encrypt the ciphertext, allowing for one-to-many data sharing. However, this scheme still has some flaws, such as low efficiency, inability to update access rules, and private data leakage. To address these issues, we proposed a scheme that combines attribute-based encryption (ABE) and identity-based encryption (IBE) to achieve efficient data sharing and data correctness verification. We also integrated this scheme with blockchain technology to ensure tamper-proof and regulated data storage, addressing issues such as data tampering and lack of supervision on third-party servers. Finally, to demonstrate the security of our scheme, we evaluated the communication overhead and computation overhead. Our results showed that our scheme is more efficient than other schemes and is secure against chosen plaintext attacks with verifiable properties.

Keywords: blockchain; attribute-based encryption; proxy re-encryption; cryptography

Citation: Feng, T.; Wang, D.; Gong, R. A Blockchain-Based Efficient and Verifiable Attribute-Based Proxy Re-Encryption Cloud Sharing Scheme. *Information* **2023**, *14*, 281. https://doi.org/10.3390/info14050281

Academic Editor: Nelly Leligou

Received: 6 February 2023
Revised: 24 March 2023
Accepted: 4 May 2023
Published: 9 May 2023

1. Introduction

Attribute-based encryption (ABE) was proposed by Waters et al. [1] as a means of achieving fine-grained access control for outsourced data to protect its confidentiality. ABE has been widely adopted in many applications. The characteristic of ABE is that it uses an access policy for users with different attribute sets in the encryption stage, ensuring that only qualified users can successfully obtain and decrypt plaintext. However, one of the drawbacks of ABE is that the access policy can become outdated and lack flexibility. Furthermore, the complexity of ABE decryption increases linearly as the number of attributes increases, making it unaffordable for mobile devices with limited computational capabilities.

The ABE scheme encrypts plaintext using an access policy, ensuring that only users with legal attributes can successfully decrypt the ciphertext. However, if the delegatee lacks the legal attributes required to decrypt the ciphertext, they can only decrypt it by obtaining the secret key of the delegator, which creates a significant security risk. Moreover, the complexity of ABE decryption increases linearly with the number of attributes, which can be time-consuming, especially on mobile devices with limited computational capabilities.

To address the issues mentioned above, we proposed a solution that combines identity-based encryption (IBE) with ABE [2]. By deploying IBE on mobile devices and ABE on PC devices, we can achieve effective encryption and decryption. Additionally, IBE enables the conversion of ABE ciphertext to IBE ciphertext, allowing delegatees with limited computational capabilities to access data at a lower computation cost.

In this paper, the EV-ABPRE encryption scheme based on blockchain is suggested. The scheme combines CP-ABE with IBE to construct an encryption scheme that can verify

the correctness of re-encrypted ciphertext. Our scheme intends to use the re-encryption key to change the ABE ciphertext into the IBE ciphertext and to liberate the delegatee from the decryption, which requires massive computational capabilities. When EV-ABPRE is used to motivate the scene, the delegator deploys the CP-ABE scheme on the PC to enable fine-grained access to the outsourced data. The delegatee accesses the data from the deployed IBE scheme to implement the effective decryption process, and that delegatee cannot directly access the original ciphertext. The delegator obtains the unique identifier of the delegatee (such as the telephone number and e-mail address); then, the algorithm is executed to output a re-encryption key and returns it to the proxy nodes. The proxy nodes generate the re-encrypted ciphertext that the delegatee could decrypt using their own IBE key to obtain the plaintext.

Our research aims to improve the security of attribute-based proxy re-encryption (ABPRE) schemes, which are commonly used to protect the confidentiality of outsourced data. While the current ABPRE scheme is effective in safeguarding the data's confidentiality, it lacks the ability to verify whether the proxy nodes honestly re-encrypted the ciphertext. This presents a significant security problem, as a dishonest proxy node could potentially leak sensitive data. To address this problem, we propose an effective and verifiable attribute-based proxy re-encryption scheme (EV-ABPRE) based on blockchain technology. Our research objective is to develop a scheme that not only maintains data confidentiality but also provides a verifiable method for ensuring the honesty of the proxy nodes. We define two types of security definitions, semantic security and verifiability, and examine our scheme's efficacy in meeting these requirements.

2. Related Work

It is well-known that IBE is an efficient encryption scheme that can utilize any string recognized as a public key and generate an IBE secret key from the string. Boneh and Franklin et al. [3] first proposed an IBE scheme based on bilinear groups. The advantage of the IBE scheme is that it does not require certificates of the public-key system, unlike traditional public-key encryption. Therefore, there is no overhead for storing and managing certificates. This scheme is widely used because of its confidentiality and efficient key management.

Waters proposed an ABE scheme to enable one-to-many data sharing [4]. ABE can be divided into two types, namely ciphertext-policy ABE (CP-ABE) and key-policy ABE (KP-ABE), depending on where the access policy is embedded [1]. The access policy is integrated into the encryption process, and only delegatees with legal attributes can successfully decrypt the ciphertext. Goyal et al. [1] noted that a drawback of encrypted data is that it can only be selectively shared at a coarse-grained level. To achieve fine-grained data sharing, they proposed the key policy attribute-based encryption. Muhammad et al. [5] combined the attribute-based access control (ABAC) framework with cloud storage and suggested an adapted CP-ABE approach to achieve fine-grained access control. The CP-ABE approach is suitable for data sharing with multiple users, while the KP-ABE approach is suitable for data sharing with a single user.

Proxy re-encryption (PRE) was first proposed by Blaze et al. [6], wherein semi-trusted proxy nodes use a re-encryption key to re-encrypt the ciphertext, allowing the delegatee to decrypt it. This approach achieves secure data sharing between the delegator and the delegatee without revealing the plaintext [7]. However, G et al. [8] proposed a bidirectional PRE scheme that cannot guarantee the security of the delegator's data, and only supports one-to-one data sharing. To address these limitations, Chen et al. [9] proposed an electronic medical record system that combines blockchain and proxy re-encryption, providing a secure solution for sharing sensitive data.

The attribute-based proxy re-encryption (ABPRE) scheme was suggested by Liang et al. [10], who combined the ABE scheme with the PRE scheme. This scheme provides a unidirectional and multi-use ciphertext policy ABPRE (CP-ABPRE) by adding attribute-based counterparts to traditional proxy re-encryption, enabling users to carry out delegation

in access control environments. The scheme supports an AND-gate policy for multivalued and negative attributes, allowing the user identified by the attribute to specify the proxy. Luo et al. [11] proposed a new CP-ABPRE scheme that supports multi-valued attributes, negative attributes, and the AND-gate policy, while Liang et al. [12] improved the existing CP-ABPRE by pointing out its vulnerability to a chosen-plaintext attack (CPA) and showing its security against a chosen-ciphertext attack (CCA) using the random oracle model. Hong et al. [13] proposed an attribute-based data retrieval scheme with proxy re-encryption to achieve fine-grained access control and data retrieval over ciphertexts. However, the problem with all KP-ABPRE methods, according to Luo et al. [14], is that they are based on classical number-theoretic assumptions, which make them vulnerable to quantum attacks. To address this, Luo proposed the first KP-ABPRE scheme based on learning with errors. Yang et al. [15] developed a one-way, non-interactive, non-transitive, non-transferable, and verifiable attribute-based proxy re-encryption scheme to ensure that user permissions are updated dynamically. Finally, Hong et al. [16] pointed out the time-bounded security and key exposure protection issues in existing ABPRE schemes and proposed a system in which users' access privileges are time-bounded. However, the delegatee requires the computational capabilities in the decryption phase to be linearly related to the number of attributes. If the computation capabilities of the delegatee's device are limited, it will take a long time to decrypt the ciphertext, and the delegatee cannot ensure the validity of the returned re-encrypted ciphertext from proxy nodes.

The linear increase in decryption complexity in ABPRE with the number of attributes is a significant burden for users with limited computation capabilities. To address this, Hua et al. [2] proposed a CP-HAPRE scheme that combines CP-ABE with IBE. This approach reduces the delegatee's burden on computational capabilities during the generation of the re-encryption key, which is generated by the unique identifier of the delegatee and the secret key of the delegator. By using this method, the complexity of generating the re-encryption key is independent of the number of attributes, making it more efficient and practical for users with limited computational capabilities.

While the previously discussed schemes require semi-trusted proxy nodes to facilitate data sharing, dishonest proxy nodes may use previously encrypted ciphertexts, even those generated at random, to minimize computation costs [17]. This can be extremely detrimental to accurate data if the plaintext differs from the correct plaintext. To address this issue, Lin et al. [18] proposed a general unidirectional single-hop ABPRE construction that introduces a commitment scheme and key derivation function to verify whether the proxy nodes have correctly re-encrypted the ciphertext. Ge et al. [19] proposed a verifiable and fair attribute-based proxy re-encryption (VF-ABPRE) scheme to support bidirectional verification operations for the proxy nodes and the delegatee, with proven confidentiality, verifiability, and fairness. However, the scheme is not suitable for delegatees with limited computational capabilities.

Blockchain is a decentralized and tamper-proof distributed ledger that provides anti-forgery features. To ensure data confidentiality, encryption algorithms are necessary, and cryptography technology can ensure secure data transmission. Zuo et al. [20] proposed a scheme that combines blockchain technology with CP-ABE, providing a secure and efficient cloud sharing scheme for the discrete logarithm problem and the decision q-parallel BDHE. Eltayieb et al. [21] proposed certificateless proxy re-encryption as an effective access control mechanism for protecting access to outsourced data. Zhang et al. [22] suggested a blockchain proxy re-encryption scheme with keyword search and attribute-based encryption, achieving better collusion resistance by using node classification and separating ciphertext storage.

3. Preliminaries

This section is mainly used to introduce the cryptography knowledge used in this scheme.

3.1. Bilinear Maps

$\mathbb{G}_1$ and $\mathbb{G}_T$ are two cyclic groups of prime order p. Let g be a generator of $\mathbb{G}_1$ and $e: \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$ be a map with the following properties:

Bilinearity: For $a, b \in \mathbb{Z}_p$, $u, v \in \mathbb{G}_1$, both $e(u^a, v^b) = e(u^b, v^a) = e(u, v)^{ab}$;

Non-degenerative: $e(g, g) \neq 1$. If the group operations in $\mathbb{G}_1$ and bilinear map $e: \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$ can be computed efficiently, then $\mathbb{G}_1$ is a bilinear group;

Computability: There is an efficient algorithm for any $g \in \mathbb{G}_1$, and all can be calculated $e(g, g)$.

3.2. LSSS [23]

Definition 1. *Let $\mathbb{P} = \{P_1, P_2, \ldots, P_n\}$ be a collection of a series of participants; a secret sharing scheme $\prod$ in $\mathbb{P}$ is linear when the following two conditions are satisfied:*

1. *Each participant has a secret about s what constitutes a vector on $\mathbb{Z}_p$;*
2. *There exists a matrix $\mathbb{A}$ called the sharing-generate matrix for $\prod$; $\mathbb{A}$ is an $l \times n$ matrix, and $\rho(i)$ is an injective function that maps each row of $\mathbb{A}$ to an attribute set, $i \in l$. Randomly select vector $\vec{v} = (s, r_2, \ldots, r_n)$; s is a secret that needs to be shared. So $\mathbb{A}_i \vec{v}$, $i \in l$, is the ith party of $\rho(i)$.*

3.3. q-Parallel BDHE Assumption

$\mathbb{G}_1$ is a cyclic group of prime order p. Let g be a generator of $\mathbb{G}_1$. Select elements $a, s, b_1, \ldots, b_n \in \mathbb{Z}_p$, given vector $g, g^s, g^a, g^{a^q}, g^{a^{q+1}}, g^{a^{2q}}, g^{sb_j}, g^{a/b_j}, \ldots, g^{a^{q+2}/b_j}, \ldots, g^{a^{2q}/b_j}$, of which $\forall 1 \leq j \leq q$, $g^{sab_k/b_j}, \ldots, g^{sa^q b_k/b_j}$, of which $\forall j \leq 1, q \geq k, j \neq k$.

The decisional q-parallel BDHE assumption immediately assumes that a probability polynomial time (PPT) algorithm that outputs $e = e(g, g)^{sa^{q+1}} \in \mathbb{G}_1$ with a non-negligible probability ϵ does not exist.

4. Modeling EV-ABPRE

EV-ABPRE combines two distinct encryption schemes, CP-ABE and IBE, where the private key Generator (PKG) generates and distributes secret keys for both. Given the typical use of a mobile device by the delegatee with limited computational capabilities and storage, data are encrypted using the CP-ABE scheme. Moreover, since the delegatee uses a mobile device, the IBE scheme is employed, which is better suited for such environments with limited computational capabilities and storage space.

The delegator is responsible for encrypting the plaintext using the CP-ABE scheme, followed by uploading the resulting ciphertext to the ciphertext chain-T. Subsequently, the delegator uploads the access policy, ciphertext storage address, and metadata to index chain-I. Once the delegatee successfully authenticates and sends its ID to the delegator, the latter generates the re-encryption key and uploads it to the proxy nodes. Finally, the system returns the verification result to the delegatee.

The ciphertext chain-T stores the ciphertext uploaded by the delegator, and the entire network verifies that the data have been successfully written into the block during storage. However, if the amount of data is large, it can cause a single point of failure, leading to a waste of storage space. To address this issue, SUN et al. [24] proposed a chain structure proposal that has been extended using the chord algorithm. This approach offers improved fault tolerance and scalability by enabling nodes to efficiently locate and retrieve data, even in large-scale decentralized networks.

The proxy nodes play a crucial role in the re-encryption process by retrieving the ciphertext from ciphertext chain-T, re-encrypting it, and then forwarding it to the delegatee. Throughout this entire process, the proxy node has no access to the plaintext, ensuring its confidentiality.

The private key generator (PKG) is responsible for generating both the public key and secret key for the CP-ABE scheme and the IBE scheme.

The access policy, storage address, and metadata information are recorded and stored in the index chain-I.

4.1. Scheme Definition

EV-ABPRE is a cryptographic system composed of two encryption schemes—CP-ABE and IBE—along with the necessary re-encryption algorithms. The EV-ABPRE algorithm is the following:

$\text{Setup}(1^{\lambda}) \rightarrow (\text{PP}_{\text{CP}}, \text{PP}_{\text{IBE}}, \text{MK})$: The PKG takes as input security parameters λ and executes the Setup algorithm, which returns public parameters PP_{CP} and PP_{IBE} and master key MK.

$\text{KeyGen}_{\text{IBE}}(\text{PP}_{\text{IBE}}, \text{MK}, \text{ID}) \rightarrow (\text{SK}_{\text{ID}})$: The PKG takes as input public parameters PP_{IBE}, master key MK, and the delegatee's ID and then executes the $\text{KeyGen}_{\text{IBE}}$ algorithm, which returns the IBE secret key SK_{ID}.

$\text{Encrypt}_{\text{CP}}(\text{m}, (\mathbb{A}, \rho), \text{PP}_{\text{CP}}) \rightarrow \text{CT}$: The delegator takes as input plaintext m, access policy $(\mathbb{A}, \rho)$ and public parameters PP_{CP} and then executes the $\text{Encrypt}_{\text{CP}}$ algorithm, which returns the ciphertext CT.

$\text{KeyGen}_{\text{CP}}(\text{PP}_{\text{CP}}, \text{MK}, \text{S}) \rightarrow (\text{SK}_{\text{S}})$: The PKG takes as input public parameters PP_{CP} of CP-ABE, master key MK, and attribute sets S. Then, it executes the $\text{KeyGen}_{\text{CP}}$ algorithm, which returns secret key SK_{S}.

$\text{ReKeyGen}(\text{PP}_{\text{CP}}, \text{SK}_{\text{S}}, \text{ID}) \rightarrow \text{rk}_{\text{S}\rightarrow\text{ID}}$: The delegator takes as input public parameters PP_{CP}, ABE secret key SK_{S}, and the delegatee's ID and then executes the ReKeyGen algorithm, which returns re-encryption key $\text{rk}_{\text{S}\rightarrow\text{ID}}$.

$\text{ReEncrypt}(\text{rk}_{\text{S}\rightarrow\text{ID}}, \text{CT}, \text{PP}_{\text{CP}}) \rightarrow \text{CT}'$: The proxy nodes take as input the delegator's re-encryption key $\text{rk}_{\text{S}\rightarrow\text{ID}}$, ciphertext CT from the ciphertext chain-T, and public parameters PP_{CP}. They then execute the ReEncrypt algorithm, which returns the re-encrypted ciphertext.

$\text{DecRe}(\text{CT}', \text{SK}_{\text{ID}}) \rightarrow \text{m}/\perp$: The delegatee takes as input re-encrypted ciphertext CT' and their secret key SK_{ID} and then executes the DecRe algorithm. If successful, this returns the plaintext. Otherwise, it returns the false symbol $\perp$.

$\text{Claim}(\text{SK}_{\text{S}}, \text{CT}')$: The delegator takes as input re-encrypted ciphertext CT' and secret key SK_{S} and then verifies whether the semi-trusted proxy nodes re-encrypted the ciphertext honestly. The algorithm returns a Boolean value of true or false depending on the outcome of the verification (Figure 1).

4.2. Scheme Definition

In the EV-ABPRE scheme, because the CP-ABE and IBE schemes are integrated into the entire process, the original ciphertext and re-encrypted ciphertext are defined, respectively.

4.2.1. Semantic Security

Regarding the choice of the security model, we have adopted the selective model, which requires the adversary to submit the challenge policy before the security game [4].

Original ciphertext semantically secure: The scheme is considered semantically secure with respect to the original ciphertext if adversary $\mathcal{A}$ has only a negligible advantage in the game, according to the selective model.

Init: Adversary $\mathcal{A}$ chooses an access policy $(\mathbb{A}^*, \rho^*)$ and ID^*, of which $\mathbb{A}^*$ is an $\text{l} \times \text{n}$ matrix.

Setup: In this phase, challenger $\mathcal{B}$ executes the algorithm Setup, which returns public parameter PP.

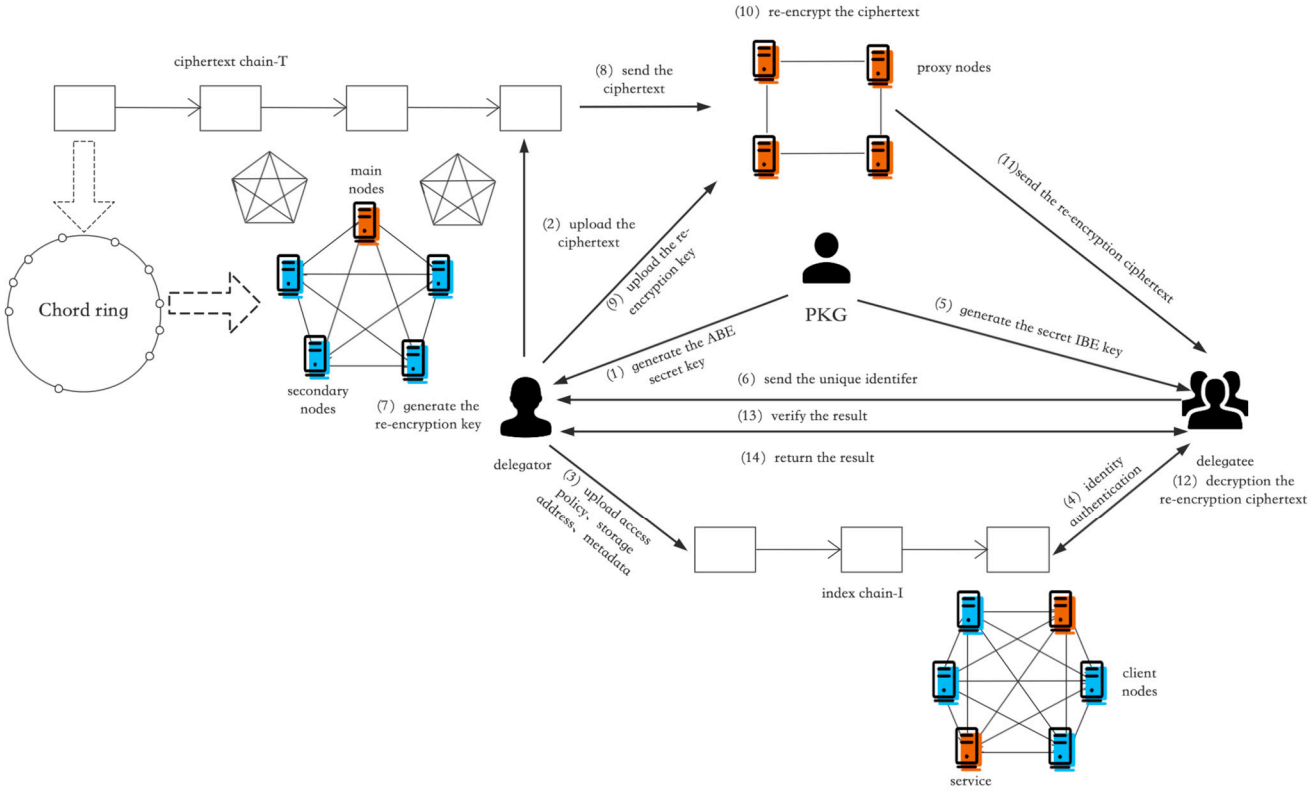

Figure 1. EV-ABPRE model.

Query Phase 1:

$\mathcal{O}_{SK_{ID}}(ID_i)$: The adversary $\mathcal{A}$ submits a query for the IBE key. If $ID_i \neq ID^*$, the challenger $\mathcal{B}$ executes the $KeyGen_{IBE}$ algorithm, which generates and returns IBE secret key SK_{ID_i} to adversary $\mathcal{A}$. Otherwise, false symbol $\perp$ is returned.

$\mathcal{O}_{SK_S}(S_i)$: Adversary $\mathcal{A}$ submits a query for the ABE key. If $S_i \notin \mathbb{A}^*$, challenger $\mathcal{B}$ executes the $KeyGen_{CP}$ algorithm, which generates and returns ABE secret key SK_{S_i} to adversary $\mathcal{A}$. Otherwise, false symbol $\perp$ is returned.

$\mathcal{O}_{RK}(S_i, ID_i)$: Adversary $\mathcal{A}$ submits a query for the re-encryption key, and challenger $\mathcal{B}$ executes the ReKeyGen algorithm, which generates and returns re-encryption key $rk_{S_i \to ID_i}$ to adversary $\mathcal{A}$.

$\mathcal{O}_{re}(CT, S_i, ID_i)$: Adversary $\mathcal{A}$ submits a query for the re-encryption result, and challenger $\mathcal{B}$ executes $KeyGen_{IBE}$, ReKeyGen and ReEncrypt algorithms, which generate and return re-encrypted ciphertext CT' to adversary $\mathcal{A}$.

Challenge Phase:

Adversary $\mathcal{A}$ submits the access policy $(\mathbb{A}^*, \rho^*)$ and two plaintexts m_0 and m_1 of equal length to challenger $\mathcal{B}$. Challenger $\mathcal{B}$ selects a plaintext randomly and then executes the $Encrypt_{CP}(m_\sigma, (\mathbb{A}^*, \rho^*), PP_{CP})$ algorithm, which returns ciphertext CT to adversary $\mathcal{A}$.

Query Phase 2:

Phase 1 queries are repeated while removing any queries that are not allowed.

Guess Phase:

Adversary $\mathcal{A}$ outputs its guess $\sigma' \in \{0, 1\}$, and the advantage of an adversary, $\mathcal{A}$, relative to winning the game is defined as follows.

$$Adv_{\mathcal{A}}^{Sem-Or} = \left| Pr[\sigma = \sigma'] - \frac{1}{2} \right| \quad (1)$$

If adversary $\mathcal{A}$ has a negligible advantage in the following game, then the scheme's original ciphertext is semantically secure under the selective model.

The re-encrypted ciphertext is semantically secure: The scheme is considered semantically secure with respect to the re-encrypted ciphertext if adversary $\mathcal{A}$ only has a negligible advantage in the game, according to the selective model.

Init: Adversary $\mathcal{A}$ chooses an access policy $(\mathbb{A}^*, \rho^*)$ and ID^*, of which $\mathbb{A}^*$ is an $l \times n$ matrix.

Setup: In this phase, challenger $\mathcal{B}$ executes the algorithm Setup, which returns public parameter PP.

Query Phase 1:

$\mathcal{O}_{SK_{ID}} ID_i$: Adversary $\mathcal{A}$ submits a query for the IBE key. If $ID_i \neq ID^*$, challenger $\mathcal{B}$ executes the $KeyGen_{IBE}$ algorithm, which generates and returns the IBE secret key SK_{ID_i} to adversary $\mathcal{A}$. Otherwise, false symbol $\perp$ is returned.

$\mathcal{O}_{SK_S}(S_i)$: Adversary $\mathcal{A}$ submits a query for the ABE key. If $S_i \notin \mathbb{A}^*$, challenger $\mathcal{B}$ executes the $KeyGen_{CP}$ algorithm, which generates and returns the ABE secret key SK_{S_i} to adversary $\mathcal{A}$. Otherwise, false symbol $\perp$ is returned.

$\mathcal{O}_{RK}(S_i, ID_i)$: Adversary $\mathcal{A}$ submits a query for the re-encryption key. If $S_i \in \mathbb{A}^*$, challenger $\mathcal{B}$ generates re-encryption key $rk_{S_i \to ID_i}$ randomly. Otherwise, $\mathcal{B}$ executes the ReKeyGen algorithm, which generates and returns the re-encryption key $rk_{S_i \to ID_i}$ to adversary $\mathcal{A}$.

Challenge Phase:

Adversary $\mathcal{A}$ submits identity ID^*, access policy $(\mathbb{A}^*, \rho^*)$, and two plaintexts m_0 and m_1 of equal length to challenger $\mathcal{B}$. Next, $\mathcal{A}$ executes the ReKeyGen algorithm to obtain the re-encrypted ciphertext. Challenger $\mathcal{B}$ then randomly selects a plaintext and executes the $Encrypt_{CP}(m_\sigma, (\mathbb{A}^*, \rho^*), PP_{CP})$ algorithm to obtain ciphertext CT, followed by the $ReEncrypt(rk_{S^* \to ID^*}, CT, PP)$ algorithm, which returns re-encrypted ciphertext CT' to $\mathcal{A}$.

Query Phase 2:

Phase 1 queries are repeated while removing any queries that are not allowed.

Guess Phase:

Adversary $\mathcal{A}$ outputs a guess, $\sigma' \in \{0, 1\}$, and adversary $\mathcal{A}$ wins the game if $\sigma = \sigma'$.

The advantage of adversary $\mathcal{A}$ in winning the game is defined as

$$Adv_{\mathcal{A}}^{Sem-Re} = \left| Pr[\sigma = \sigma'] - \frac{1}{2} \right| \tag{2}$$

Definition 2. *The EV-ABPRE scheme is CPA-secure under the selective model if all PPT adversaries, their advantage* $Adv_{\mathcal{A}}^{Sem-Re}$ *and* $Adv_{\mathcal{A}}^{Sem-Or}$ *are negligible.*

4.2.2. Verifiability

Init: Adversary $\mathcal{A}$ chooses an access policy $(\mathbb{A}^*, \rho^*)$ and ID^*, of which $\mathbb{A}^*$ is an $l \times n$ matrix.

Setup : In this phase, challenger $\mathcal{B}$ executes the algorithm Setup, which returns public parameter PP.

Query phase 1:

$\mathcal{O}_{SK_{ID_i}}(ID_i)$: Adversary $\mathcal{A}$ submits a query for the IBE key. If $ID_i \neq ID^*$, challenger $\mathcal{B}$ executes the $KeyGen_{IBE}$ algorithm, which generates and returns the IBE secret key SK_{ID_i} to adversary $\mathcal{A}$. Otherwise, false symbol $\perp$ is returned.

$\mathcal{O}_{RK}(S_i, ID_i)$: Adversary $\mathcal{A}$ submits a query for the re-encryption key. If $S_i \in \mathbb{A}^*$, it generates re-encryption key $rk_{S_i \to ID_i}$ randomly. Otherwise, challenger $\mathcal{B}$ executes the ReKeyGen algorithm, which generates and returns the re-encryption key $rk_{S_i \to ID_i}$ to adversary $\mathcal{A}$.

$\mathcal{O}_{Claim}(SK_{S_i}, CT')$: Adversary $\mathcal{A}$ submits a query for the re-encrypted ciphertext verification, and challenger $\mathcal{B}$ returns the verification result.

Challenge phase:

Adversary $\mathcal{A}$ submits access policy $(\mathbb{A}^*, \rho^*)$ and plaintext m^* to challenger $\mathcal{B}$. Challenger $\mathcal{B}$ executes $\text{Encrypt}_{CP}(m^*, (\mathbb{A}^*, \rho^*), PP) \rightarrow CT$ and then returns ciphertext CT to adversary $\mathcal{A}$.

Query Phase 2:

Phase 1 queries are repeated while removing any queries that are not allowed.

Guess phase:

Adversary $\mathcal{A}$ outputs attribute set $S^*, S^* \in \mathbb{A}^*$, and re-encrypted ciphertext CT' if $\text{DecRe}\left(CT^{*\prime}, SK_{S^*}\right) \neq m^*$.

The advantage of adversary $\mathcal{A}$ in winning the game is defined as

$$\text{Adv}_{\mathcal{A}}^{Ver} = \left|\Pr[\text{A wins}|\text{Ver}] \right. \tag{3}$$

Definition 3. *The EV-ABPRE scheme is verifiable under the selective model if all PPT adversaries' advantage* $Adv_{\mathcal{A}}^{Ver}$ *is negligible.*

5. Our Construction

5.1. The EV-ABPRE Construction

$\text{Setup}\left(1^{\lambda}\right) \rightarrow (PP_{CP}, PP_{IBE}, MK)$: The PKG generates a bilinear pairing tuple $PP_{CP}(p, g, \mathbb{G}_1, \mathbb{G}_T, e)$ and then randomly selects element $\alpha, \beta \in \mathbb{Z}_p, u, h, w, v, f \in \mathbb{G}_1$, selecting encoding function $F : \mathbb{G}_T \rightarrow \mathbb{G}_1$. $PP_{CP} = \left\{g, u, h, w, v, f, e(g,g)^{\alpha}, F\right\}, PP_{IBE} = \left\{u, h, g, e(g,g)^{\alpha}, F\right\}$ and the master key $MK = \{\alpha\}$.

$\text{KeyGen}_{IBE}(PP_{IBE}, MK, ID) \rightarrow (SK_{ID})$: PKG selects random numbers $r \in \mathbb{Z}_p$ and outputs $SK_{ID} = \left(g^{\alpha}\left(u^{ID}h\right)^{r}, g^{r}\right)$.

$\text{Encrypt}_{CP}(m, (\mathbb{A}, \rho), PP_{CP}) \rightarrow CT$: The delegator encrypts plaintext m with LSSS access policy $(\mathbb{A}, \rho)$, of which $\mathbb{A}$ is an $l \times n$ matrix and ρ is an injective function that maps the ith row of $\mathbb{A}$ to an attribute set $T = \left(t_{\rho(1)}, t_{\rho(2)}, \ldots, t_{\rho(l)}\right)$. Choosing a random vector $\vec{v} = (s, \widetilde{r_2}, \ldots, \widetilde{r_n}) \in \mathbb{Z}_p$, s is the secret to be shared. For each row of $\mathbb{A}$, $\lambda_i = A_i\vec{v}$ is computed. Elements $k, y_2, \ldots, y_l, z_i (i \in S) \in \mathbb{Z}_p$ are selected randomly. Then, $C = m \cdot e(g,g)^{\alpha s}, C_0 = g^s, C_{i,1} = w^{\lambda_i} v^{y_i}, C_{i,2} = u^{t_{\rho(i)}}, C_{i,3} = g^{y_i}, C_{i,4} = g^{z_{\rho(i)}}, C_5 = f^s, CT = \left(C, C_0, \{C_{i,1}, C_{i,2}, C_{i,3}, C_{i,4}\}_{i=1}^{l}, C_5\right)$ is computed.

$\text{KeyGen}_{CP}(PP_{CP}, MK, S) \rightarrow (SK_S)$: PKG takes as input public parameters PP_{CP}, master key MK, and attribute set $S = \left(att_1, att_2, \ldots att_{|S|}\right)$; PKG randomly selects $r_1, r_2, \ldots, r_{|S|} \in \mathbb{Z}_p, r' = r_1 + r_2 + \ldots + r_{|S|} \in \mathbb{Z}_p$ and computes

$$K_0 = g^{\alpha} w^{r'}, K_1 = g^{r'}, K_{i,2} = g^{r_i}, K_{i,3} = u^{att_i} v^{-r'}, K_{j,4} = g^{\frac{\lambda_i}{z_{\rho(i)}}}$$

$$SK_S = \left(K_0, K_1, \{K_{i,2}K_{i,3}\}_{i=1}^{|S|}, \{K_{j,4}\}_{j=1}^{l}\right)$$

$\text{ReKeyGen}(PP_{CP}, SK_S, ID) \rightarrow rk_{S \rightarrow ID}$: The delegator takes as input PP_{CP}, $SK_S = \left(K_0, K_1, \{K_{i,2}K_{i,3}\}_{i=1}^{|S|}, \{K_{j,4}\}_{j=1}^{l}\right)$,ID, and the delegator selects a random number, $\widetilde{r}, \widetilde{t} \in \mathbb{Z}_p$. Then, it computes

$$rk_o = K_0 f^{\widetilde{r}}, rk_1 = K_1, rk_{i,2} = K_{i,2}, rk_{i,3} = K_{i,3}, rk_4 = F\left(e(g,g)^{\alpha\,\widetilde{t}}\right) g^{\widetilde{r}}, rk_5 = \left(u^{ID}h\right)^{\widetilde{t}}, rk_6 = g^{\widetilde{t}}$$

$$rk_{S \rightarrow ID} = \left(rk_o, rk_1, \{rk_{i,2}, rk_{i,3}\}_{i=1}^{|S|}, rk_4, rk_5, rk_6\right)$$

$\text{ReEncrypt}(rk_{S \rightarrow ID}, CT, PP_{CP}) \rightarrow CT'$: The proxy nodes takes as input $rk_{S \rightarrow ID} = \left(rk_o, rk_1, \{rk_{i,2}, rk_{i,3}\}_{i=1}^{|S|}, rk_4, rk_5, rk_6\right)$, $CT = \left(C, C_0, \{C_{i,1}, C_{i,2}, C_{i,3}, C_{i,4}\}_{i=1}^{l}, C_5\right)$. If the ciphertext CT and access policy $(\mathbb{A}, \rho)$ is associated and attribute set S satisfies $\mathbb{A}$, then

$I = \{i : \rho(i) \in S\}$ and $\{\omega_i \in \mathbb{Z}_p\}_{i\in I}$ exists, resulting in $\sum_{i\in l} \omega_i A_i = (1, 0, \ldots, 0)$; thus, the proxy nodes computes

$$B = \frac{e(C_0, rk_0)}{\prod_{i\in I}\left(e(C_{i,1}, rk_1)\cdot e(C_{i,2}, rk_2)\cdot e\left(C_{i,3}, rk_{j,3}\right)\right)^{\omega_i}} \tag{4}$$

$$C' = C/B, C'_0 = rk_4, C'_1 = rk_5, C'_2 = rk_6, C'_3 = C_5, C'_{i,4} = C_{i,4}, CT' = \left(C', C'_0, C'_1, C'_2, C'_3, \{\mathbf{C'_{i,4}}\}_{\mathbf{i}=1}^{\mathbf{l}}\right)$$

$\mathrm{DecRe}\left(CT', SK_{ID}\right) \rightarrow m$: The delegatee takes as input
$CT\prime = \left(C', C'_0, C'_1, C'_2, C'_3, \{C'_{i,4}\}_{i=1}^{l}\right)$, $SK_{ID} = \left(g^{\alpha}\left(u^{ID}h\right)^{r}, g^{r}\right)$ and then calculates

$$\frac{e\left(K_{ID,0}, C'_2\right)}{e\left(K_{ID,1}, C'_1\right)} = e(g,g)^{\alpha\,\tilde{t}} \tag{5}$$

$$g^{\tilde{t}} = \frac{C'_0}{F\left(e(g,g)^{\alpha\,\tilde{t}}\right)} \tag{6}$$

$$m = C'\cdot e\left(g^{\tilde{t}}, C'_3\right) \tag{7}$$

$\mathrm{Claim}\left(SK_S, CT'\right)$: The delegator inputs $SK_S = \left(K_0, K_1, \{K_{i,2}K_{i,3}\}_{\mathbf{i}=1}^{|\mathbf{S}|}, \{K_{j,4}\}_{\mathbf{j}=1}^{\mathbf{l}}\right)$,
$CT' = \left(C', C'_0, C'_1, C'_2, C'_3, \{\mathbf{C'_{i,4}}\}_{\mathbf{i}=1}^{\mathbf{l}}\right)$.

We verify whether $e\left(K_{j,4}, C'_{i,4}\right)$ is equal to $e(g,g)^{\lambda_i}$. If it is equal to it, true is returned; otherwise, false is returned.

A_i is the i row of matrix A, $A_i = (a_{i,1}, a_{i,2}, \ldots, a_{i,n})$.

$$e(g,g)^{\lambda_i} = e(g,g)^{A_i\vec{v}} = e(g,g)^{a_{i,1}s + a_{i,2}r_2 + \ldots + a_{i,n}r_n} = \left(e(g,g)^{s}\right)^{a_{i,1}}\prod_{k=2}^{n}\left(e(g,g)^{r_k}\right)^{a_{i,k}}$$

5.2. Correctness

Compute the correctness of the re-encrypted ciphertext if attribute set S of re-encryption key $rk_{S\rightarrow ID}$ satisfies access policy $(\mathbb{A}, \rho)$; the following is computed.

$$\begin{aligned} B &= \frac{e\left(g^{s}, g^{\alpha}w^{r'}f^{\tilde{r}}\right)}{\prod_{i\in I}\left(e\left(w^{\lambda_i}v^{y_i}, g^{r'}\right)\cdot e\left(u^{t_{\rho(i)}}, g^{r_i}\right)\cdot e\left(g^{y_i}, u^{t_{\rho(i)}}v^{-r'}\right)\right)^{\omega_i}} \\ &= e(g^{s}, g^{\alpha})e\left(g^{s}, w^{r'}\right)/e(w,g)^{r\sum_{i\in I}\omega_i\lambda_i} \\ &= e(g,g)^{\alpha s}e\left(g^{s}, f^{\tilde{r}}\right) \end{aligned} \tag{8}$$

$$C' = \frac{m}{e\left(g^{s}, f^{\tilde{r}}\right)} = m\cdot e(g,g)^{\alpha s}/\left(e(g,g)^{\alpha s}e\left(g^{s}, f^{\tilde{r}}\right)\right) = m/e\left(g^{s}, f^{\tilde{r}}\right) \tag{9}$$

$$\frac{e\left(K_{ID,0}, C'_2\right)}{e\left(K_{ID,1}, C'_1\right)} = \frac{e\left(g^{\alpha}\left(u^{ID}h\right)^{r}, g^{\tilde{t}}\right)}{e\left(g^{r}, \left(u^{ID}h\right)^{\tilde{t}}\right)} = e(g,g)^{\alpha\tilde{t}} \tag{10}$$

$$\frac{C'_0}{F\left(e(g,g)^{\alpha\tilde{t}}\right)} = \frac{F\left(e(g,g)^{\alpha\tilde{t}}\right)g^{\tilde{r}}}{F\left(e(g,g)^{\alpha\tilde{t}}\right)} = g^{\tilde{r}} \tag{11}$$

$$m = C'\cdot e\left(g^{\tilde{t}}, C'_3\right) = \frac{m}{e\left(g^{s}, f^{\tilde{r}}\right)}\,e\left(g^{\tilde{t}}, f^{s}\right) \tag{12}$$

5.3. Semantic Security

Theorem 1. *EV-ABPRE is semantically secure under the q-parallel BDHE assumption.*

Proof. Suppose there is a PPT adversary, $\mathcal{A}$, that has a non-negligible advantage, ϵ, in breaking the original ciphertext semantic security of the EV-ABPRE scheme. In this case, we can construct simulator $\mathcal{B}$, which can solve the q-parallel BDHE assumption with the same advantage, ϵ. Specifically, $\mathcal{B}$ generates a q-parallel instance $\left(\vec{v}, T\right)$ with the goal of determining whether T is equal to $e(g,g)^{s\ \beta^{q+1}}$ or whether it is randomly selected from $\mathbb{G}_T$. □

Lemma 1. *EV-ABPRE is originally ciphertext secure under the q-parallel BDHE assumption.*

Query Phase 1:

Init: Adversary $\mathcal{A}$ outputs access policy $(\mathbb{A}^*, \rho^*)$ and ID^*, of which $\mathbb{A}^*$ is an $l \times n$ matrix, and challenger $\mathcal{B}$ creates three forms that are initially empty $\mathcal{L}_{RK}$ = (S_i,ID_i,$rk_{S\to ID}$), $\mathcal{L}_{SK_{ID}}$=(ID_i,SK_{ID_i}), and $\mathcal{L}_{SK_S}$ = (S_i,SK_{S_i}).

Setup : Challenger $\mathcal{B}$ randomly selects $n \in \mathbb{Z}_p$ and sets $e(g,g)^{\alpha} = e(g,g)^{n} e\left(g^{\beta}, g^{\beta^q}\right)$, so we are aware that $\alpha = n + \beta^{q+1}$; it then sets $PP = \left\{g, u, h, w, v, e(g,g)^{\alpha}\right\}$. Then, element γ is randomly selected, and $f = g^{\gamma}$ is computed. Encoding function $F : \mathbb{G}_T \to \mathbb{G}_1$ is selected. Finally, public parameter $PP = (g, u, h, w, f, v, e(g,g)^{\alpha}, F)$ is returned.

$\mathcal{O}_{SK_{ID_i}}(ID_i)$: Adversary $\mathcal{A}$ submits a query for the IBE key. If $ID_i = ID^*$, challenger $\mathcal{B}$ returns false symbol $\perp$ to adversary $\mathcal{A}$. Otherwise, $\mathcal{B}$ selects a vector $\vec{\theta} = (\theta_1, \theta_2, \ldots, \theta_n)$, of which $\theta_1 = -1$ and for all i where $\rho^*(i) \in S$, we know $\vec{\theta} \cdot \mathbb{A}_i^* = 0$; then, the following is computed.

$$K_{ID,1} = g^{r'} \prod_{i=1}^{n^*} \left(g^{\beta^{q+1-i}}\right)^{\theta_i} \triangleq g^{r} \tag{13}$$

From the above, $r = r' + \sum_{i=1}^{n^*} \theta_i \cdot \beta^{q+1-i}$.

In that case,

$$\begin{aligned} K_{ID,0} &= g^{\alpha} \left(u^{ID_i} h\right)^{r} = g^{\beta^{q+1}+n} \cdot \left(u^{ID_i} h\right)^{r' + \sum_{i=1}^{n^*} \theta_i \beta^{q+1-i}} \\ &= g^{m} \left(u^{ID_i} h\right)^{r'} \prod_{i=2}^{n^*} \left(g^{\beta^{q+1-i}}\right)^{\theta_i} \end{aligned} \tag{14}$$

This returns SK_{ID_i} to adversary $\mathcal{A}$ and stores it in form $\mathcal{L}_{SK_{ID}}$.

$\mathcal{O}_{SK_S}(S_i)$: Adversary $\mathcal{A}$ submits a query for the ABE key. If $S_i \in \mathbb{A}^*$, challenger $\mathcal{B}$ returns false symbol $\perp$ to the adversary. Otherwise, challenger $\mathcal{B}$ executes the $KeyGen_{CP}$ algorithm, which generates and returns the ABE secret key SK_{S_i} to adversary $\mathcal{A}$ and stores it in form $\mathcal{L}_{SK_S}$.

$\mathcal{O}_{RK}(S_i, ID_i)$: Adversary $\mathcal{A}$ submits a query for the re-encryption key. If (S_i, ID_i) already exists in form $\mathcal{L}_{RK}$, challenger $\mathcal{B}$ returns $rk_{S\to ID}$ to adversary $\mathcal{A}$. Otherwise, $\mathcal{B}$ first executes the $KeyGen_{CP}$ algorithm, which generates a CP-ABE secret key SK_{S_i}. Then, $\mathcal{B}$ executes the ReKeyGen algorithm, which generates and returns re-encryption key $rk_{S_i\to ID_i}$ to adversary $\mathcal{A}$ and stores it in $\mathcal{L}_{RK}$.

$\mathcal{O}_{re}(CT, S_i, ID_i)$: Adversary $\mathcal{A}$ submits a query for the re-encryption result, challenger $\mathcal{B}$ will run algorithms $KeyGen_{IBE}$, ReKeyGen and ReEncrypt. the challenger $\mathcal{B}$ executes the $KeyGen_{IBE}$, ReKeyGen and ReEncrypt algorithm, which generates and returns the re-encrypted ciphertext CT′ to the adversary $\mathcal{A}$.

Challenge phase:

Adversary $\mathcal{A}$ submits access policy $(\mathbb{A}^*, \rho^*)$ and two plaintexts m_0 and m_1 of equal length to challenger $\mathcal{B}$. Challenger $\mathcal{B}$ then randomly selects a plaintext and executes the

$\text{Encrypt}_{CP}(m_\sigma, (\mathbb{A}^*, \rho^*), PP)$ algorithm to obtain ciphertext CT. Definition $X = \{i : \rho(i) = x\}$ is as follows.

$$u^{t_{\rho(x)}} = g^{t_x} \prod_{i \in X} g^{\beta A^*_{i,1}} g^{\beta^2 A^*_{i,2}} \ldots\ldots g^{\beta^{n^*} A^*_{i,n}} \tag{15}$$

$$C = m_\sigma Te(g,g)^{ns},\ C_0 = g^s$$

$$C_{i,1} = w^{\lambda_i} v^{y_i} = \prod_{j=1}^{l^*} v^{y_j} \prod_{i=2}^{n^*} w^{\tilde{r}_i A^*_{j,i}} \tag{16}$$

$$C_{i,2} = \left(u^{t_{\rho(i)}} h\right)^{-y_i} = g^{t_i} \prod_{j=1}^{l^*} \prod_{x=1}^{n^*} \left(g^{\beta^x A^*_{j,x}} h\right)^{-y_j} \tag{17}$$

$$C_{i,3} = \prod_{j=1}^{l^*} g^{y_j} \tag{18}$$

$$C_4 = f^s$$

$$CT = \left(C, C_0, \{C_{i,1}, C_{i,2}, C_{i,3}\}_{i=1}^{l}, C_4\right)$$

Ciphertext CT is returned to adversary $\mathcal{A}$.

Query phase 2:

Phase 1 queries are repeated while removing any queries that are not allowed.

Guess phase:

Adversary $\mathcal{A}$ outputs its guess, $\sigma' \in \{0,1\}$. If $\sigma = \sigma'$, this means $T = e(g,g)^{s\ \beta^{q+1}}$, and $\mathcal{A}$ wins in the game, with $C = m_\sigma Te(g,g)^{ns} = me(g,g)^{s\ \beta^{q+1}} e(g,g)^{ns} = me(g,g)^{s(n+\beta^{q+1})} = me(g,g)^{\alpha s}$. When $T = e(g,g)^{r\ \alpha^{q+1}}$, this means $\Pr[\sigma = \sigma'] = 1/2 + \epsilon$; therefore, $\text{Adv}_{\mathcal{A}}^{\text{Sem-Or}} = \Pr[\sigma = \sigma'] - 1/2 = \epsilon$, which means that $\mathcal{B}$ has a non-negligible advantage when solving the q-parallel BDHE assumption.

Proof. Suppose that there is a PPT adversary, $\mathcal{A}$, that has a non-negligible advantage, ϵ, in breaking the re-encrypted ciphertext semantic security of the EV-ABPRE scheme. In this case, we can construct simulator $\mathcal{B}$, which can solve the q-parallel BDHE assumption with the same advantage, ϵ. □

Lemma 2. *EV-ABPRE is re-encrypted and ciphertext secure under the q-parallel BDHE assumption.*

Init : Adversary $\mathcal{A}$ outputs access policy $(\mathbb{A}^*, \rho^*)$ and ID^*, of which $\mathbb{A}^*$ is an $l \times n$ matrix, and challenger $\mathcal{B}$ creates two forms that are initially empty: $\mathcal{L}_{RK}$=(S_i,ID_i,$rk_{S \to ID}$) and $\mathcal{L}_{SK_{ID}}$=(ID_i,SK_{ID_i}).

Setup : Challenger $\mathcal{B}$ randomly selects $n \in \mathbb{Z}_p$. Then, element γ is randomly selected, and $f = g^\gamma$ is computed. Encoding function $F : \mathbb{G}_T \to \mathbb{G}_1$ is selected. Finally, public parameter $PP = (g, u, h, w, f, v, e(g,g)^\alpha, F)$ is outputted.

$\mathcal{O}_{SK_{ID_i}}(ID_i)$: Adversary $\mathcal{A}$ submits a query for the IBE key. If $ID_i = ID^*$, challenger $\mathcal{B}$ returns false symbol $\perp$. Otherwise, challenger $\mathcal{B}$ executes the KeyGen_{IBE} algorithm, which generates and returns the IBE secret key SK_{ID_i} to the adversary $\mathcal{A}$ and stores it in form $\mathcal{L}_{SK_{ID}}$.

$\mathcal{O}_{SK_S}(S_i)$: Adversary $\mathcal{A}$ submits a query for the ABE key. If $S_i \notin \mathbb{A}^*$, challenger $\mathcal{B}$ executes the KeyGen_{CP} algorithm, which generates and returns the ABE secret key SK_{S_i} to adversary $\mathcal{A}$. Otherwise, false symbol $\perp$ is returned.

$\mathcal{O}_{RK}(S_i, ID_i)$: Adversary $\mathcal{A}$ submits a query for the re-encryption key. If (S_i, ID_i) already exists in form $\mathcal{L}_{RK}$, challenger $\mathcal{B}$ returns the $rk_{S \to ID}$ to adversary $\mathcal{A}$. Otherwise, if $S_i \notin \mathbb{A}^*$, challenger $\mathcal{B}$ executes the KeyGen_{CP} algorithm, which returns the ABE se-

cret key SK_{S_i}; then, it randomly selects $t\prime, s' \in \mathbb{Z}_p$ and computes $rk_0 = K_0 \cdot f^{t'}, rk_1 = K_1, \left\{rk_{j,2} = K_{j,2}, rk_{j,3} = K_{j,3}\right\}_{j=1}^{|S|}, rk_4 = F\left(e(g,g)^{\alpha s'}\right) g^{t'}, rk_5 = \left(u^{ID_i} h\right)^s, rk_6 = g^{s'}$, returning $rk_{S_i \to ID_i}$ to adversary $\mathcal{A}$ and storing it in $\mathcal{L}_{RK}$. Otherwise, $S_i \in \mathbb{A}^*$, a random re-encryption key, is randomly generated by challenger $\mathcal{B}$. It selects $t\prime, s' \in \mathbb{Z}_p$ randomly, $r_1, r_2, \ldots, r_{|S|}, rk_0 \in \mathbb{G}_1$, $r' = r_1 + r_2 + \ldots + r_{|S|}$, and then computes

$$rk_1 = g^{r'}\left\{rk_{j,2} = g^{r_j}, rk_{j,3} = \left(u^{att_j} h\right)^{r_j} v^{-r'}\right\}_{j=1}^{|S|}, rk_4 = F\left(e(g,g)^{\alpha s'}\right) g^{t'}, rk_5 = \left(u^{ID} h\right)^{s'}, rk_6 = g^{s'}.$$

$rk_{S_i \to ID_i}$ is returned to adversary $\mathcal{A}$ and is then stored in $\mathcal{L}_{RK}$.

Challenge phase:

Adversary $\mathcal{A}$ submits identity ID^*, access policy $(\mathbb{A}^*, \rho^*)$, and two plaintexts m_0 and m_1 of equal length to challenger $\mathcal{B}$. Next, $\mathcal{A}$ executes the ReKeyGen algorithm to obtain the re-encrypted ciphertext. Challenger $\mathcal{B}$ then randomly selects a plaintext and executes the $\text{Encrypt}_{CP}(m_\sigma, (\mathbb{A}^*, \rho^*), PP)$ algorithm to obtain ciphertext CT, followed by the $\text{ReEncrypt}(rk_{S^* \to ID^*}, CT, PP)$ algorithm, which generates and returns re-encrypted ciphertext CT' to adversary $\mathcal{A}$.

$$C_3' = C_0^{\gamma} = f^s, CT^{*\prime} = \left(C', C_0', C_1', C_2', C_3'\right).$$

Query phase 2:

Phase 1 queries are repeated while removing any queries that are not allowed.

Guess phase:

We first show that adversary $\mathcal{A}$ has the ability to distinguish between a random re-encryption key and a well-formed re-encryption key. When $S_i \in \mathbb{A}^*$, challenger $\mathcal{B}$ selects a $rk_0 \in \mathbb{G}_1$. There must be a random number, $t'' \in \mathbb{Z}_p$, which results in $rk_4 = g^{\alpha} w^{r'} f^{t''}$ for rk_0. From the above, $rk_6 = F\left(e(g,g)^{\alpha\, s\prime}\right) g^{t''}$. The random re-encryption key can be written as $rk'_{S_i \to ID_i} = \left(g^{\alpha} w^{r'} f^{t''}, g^{r'}, \left\{g^{r_j}, u^{att_j} v^{-r'}\right\}_{j=1}^{1}, F\left(e(g,g)^{\alpha\, s\prime}\right) g^{t'}, \left(u^{ID} h\right)^{s'}, g^{s'}\right)$.

The well-formed re-encryption key can be written as

$$rk'_{S_i \to ID_i} = \left(g^{\alpha} w^{r'} f^{\,t\,''}, g^{r'}, \left\{g^{r_j}, u^{att_j} v^{-r'}\right\}_{j=1}^{1}, F\left(e(g,g)^{\alpha s'}\right) g^{t''}, \left(u^{ID} h\right)^{s'}, g^{s'}\right)$$

Therefore, adversary $\mathcal{A}$ requires a clear distinction between randomly generated $F\left(e(g,g)^{\alpha s'}\right) g^{t''}$ and well-formed $F\left(e(g,g)^{\alpha s'}\right) g^{t'}$. These two parts are encryptions of the IBE for $g^{t'}$ and $g^{t''}$. Therefore, adversary $\mathcal{A}$ distinguishes the random re-encryption key and the re-encryption key generated according to the algorithm with the same distribution as the IBE scheme, $\Pr[\sigma = \sigma'] = \frac{1}{2} + \epsilon$; therefore, $Adv_{\mathcal{A}}^{Sem-Re} = \Pr[\sigma = \sigma'] - \frac{1}{2} = \epsilon$, and challenger $\mathcal{B}$ has a non-negligible advantage in solving the q-parallel BDHE assumption.

5.4. Verifiability

Proof. Suppose there is a PPT adversary, $\mathcal{A}$, that has a non-negligible advantage, ϵ, in breaking the verifiability of the EV-ABPRE scheme. In this case, we can construct simulator $\mathcal{B}$, which can solve the q-parallel BDHE assumption with the same advantage, ϵ. □

Lemma 3. *EV-ABPRE is verifiable under the discrete logarithm assumption.*

Init : Adversary $\mathcal{A}$ outputs access policy $(\mathbb{A}^*, \rho^*)$ and ID^*, of which $\mathbb{A}^*$ is an $l \times n$ matrix, and challenger $\mathcal{B}$ creates two forms that are initially empty: $\mathcal{L}_{RK}$=(S_i,ID_i,$rk_{S \to ID}$) and $\mathcal{L}_{SK_{ID}}$=(ID_i,SK_{ID_i}).

Setup : Challenger $\mathcal{B}$ randomly selects $n \in \mathbb{Z}_p$. Then, it randomly selects element γ and computes $f = g^{\gamma}$. Encoding function $F : \mathbb{G}_T \to \mathbb{G}_1$ is selected. Finally, public parameter $PP = (g, u, h, w, f, v, e(g,g)^{\alpha}, F)$ is outputted.

Query phase 1:

$\mathcal{O}_{SK_{ID_i}}(ID_i)$: Adversary $\mathcal{A}$ submits a query for the IBE key. If $ID_i = ID^*$, challenger $\mathcal{B}$ returns false symbol $\perp$. Otherwise, challenger $\mathcal{B}$ executes the $KeyGen_{IBE}$ algorithm, which generates and returns the IBE secret key SK_{ID_i} to adversary $\mathcal{A}$, and stores it in $\mathcal{L}_{SK_{ID}}$.

$\mathcal{O}_{RK}(S_i, ID_i)$: Adversary $\mathcal{A}$ submits a query for the re-encryption key. If (S_i, ID_i) already exists in form $\mathcal{L}_{RK}$, challenger $\mathcal{B}$ returns $rk_{S\to ID}$ to the adversary. Otherwise, $\mathcal{B}$ first executes the $KeyGen_{CP}$ algorithm, which generates a CP-ABE secret key SK_{S_i}. Then, challenger $\mathcal{B}$ executes the ReKeyGen algorithm, which generates and returns re-encryption key $rk_{S_i\to ID_i}$ to adversary $\mathcal{A}$ and stores it in $\mathcal{L}_{RK}$.

$\mathcal{O}_{Claim}(SK_S, CT')$: Adversary $\mathcal{A}$ submits a query for the re-encrypted ciphertext verification to check whether the re-encrypted ciphertext is valid. If it is valid, true is returned to adversary $\mathcal{A}$. Otherwise, false is returned.

Challenge phase:

Adversary $\mathcal{A}$ submits access policy $(\mathbb{A}^*, \rho^*)$ and plaintext m^* to challenger $\mathcal{B}$. Challenger $\mathcal{B}$ executes $Encrypt_{CP}(m^*, (\mathbb{A}^*, \rho^*), PP) \to CT$ and then returns ciphertext CT to adversary $\mathcal{A}$.

Query phase 2: Repeats the query of phase 1, removing queries that are not allowed.

Guess phase:

Adversary $\mathcal{A}$ outputs an IBE key, SK_{ID^*}, as well as re-encrypted ciphertext $CT^{*\prime}$. If $DecRe(CT^{*\prime}, SK_{ID^*}) \neq m^*$, $\mathcal{A}$ wins the game, and the advantage of $\mathcal{A}$ in winning this game is defined as $Adv_{\mathcal{A}}^{Ver} = \left|Pr[\mathcal{A}\ wins \middle| Ver\right]$. For all PPT adversaries, if its advantage, $Adv_{\mathcal{A}}^{Ver}$, is negligible, this means that the scheme is verifiable.

6. Performance Evaluation

To further evaluate the scheme, we selected several similar ones and compared them in three aspects: functionality, communication overhead, computational overhead, and security. By conducting simulation experiments, we analyzed the advantages and disadvantages of the scheme and similar ones in terms of computing costs.

6.1. Functionality Comparison

The functionality comparison between our scheme and other secure data sharing schemes is shown in Table 1. Proxy re-encryption is used in all schemes in [12,19,25,26], but it cannot ensure that the original ciphertext is tamper-proof without the participation of the blockchain, and the schemes in [12,25,26] cannot verify whether or not the re-encrypted ciphertext is honestly re-encrypted by the proxy nodes. The scheme ensures the security of the data storage and sharing process based on the blockchain. Fine-grained access control is offered via attribute-based proxy re-encryption, and the delegator can confirm the validity of the ciphertext. In conclusion, our scheme offers more advantages in comparison in terms of functionality.

Table 1. Functionality comparison.

Scheme	Attribute Sets	Blockchain	ABE	PRE	Verifiable	Security
[12]	√	×	√	√	√	CPA
[19]	√	×	√	√	√	Semantic security
[25]	√	×	√	√	×	CPA
[26]	√	√	√	√	×	CPA
Our scheme	√	√	√	√	√	Semantic security

Table 1 presents a comparison of the functionalities of our secure data sharing scheme with similar schemes. While all schemes in [12,19,25,26] use proxy re-encryption, they cannot guarantee the original ciphertext's tamper-proof nature without the involvement of the blockchain. Additionally, schemes in [25,26] fail to verify whether the proxy nodes have honestly re-encrypted the ciphertext. Our scheme ensures secure data storage and

sharing via the blockchain and offers fine-grained access control using attribute-based proxy re-encryption. The delegatee can also verify the re-encrypted ciphertext's validity. Hence, our scheme has a more comprehensive set of functionalities than the others, making it more advantageous.

6.2. Communication Overhead

The communication overhead of our scheme was evaluated by analyzing the length of the secret key, public parameter, ciphertext, and re-encryption key, as shown in Tables 2 and 3. Our results demonstrate that our scheme has low communication overhead, especially when compared to other similar schemes. For instance, the length of the secret key and ciphertext in our scheme is shorter than that of [19,25]. Furthermore, our scheme has smaller-sized public parameters and re-encryption keys than [12,26]. These results indicate that our scheme is more efficient in terms of communication overhead.

Table 2. Delegator's communication overhead.

Scheme	Secret Key	Ciphertext	Public Parameters
[12]	$3\|\mathbb{G}_1\|$	$(2+2n)\|\mathbb{G}_1\|$	$3\|\mathbb{G}_1\|+\|\mathbb{G}_T\|$
[19]	$(3+n)\|\mathbb{G}_1\|$	$(2+3n)\|\mathbb{G}_1\|+\|\mathbb{G}_T\|$	$(5+n)\|\mathbb{G}_1\|+\|\mathbb{G}_T\|$
[25]	$(2+2n)\|\mathbb{G}_1\|$	$(2+n)\|\mathbb{G}_1\|+\|\mathbb{G}_T\|$	$3\|\mathbb{G}_1\|+\|\mathbb{G}_T\|$
[26]	$(2+n)\|\mathbb{G}_1\|$	$(2+2n)\|\mathbb{G}_1\|+\|\mathbb{G}_T\|$	$3\|\mathbb{G}_1\|$
Our scheme	$(3+4n)\|\mathbb{G}_1\|$	$(2+5n)\|\mathbb{G}_1\|+\|\mathbb{G}_T\|$	$6\|\mathbb{G}_1\|+\|\mathbb{G}_T\|$

Table 3. Delegatee's communication overhead.

Scheme	Secret Key	Re-Encrypted Ciphertext	Public Parameters
[12]	$3\|\mathbb{G}_1\|$	$(1+2n)\|\mathbb{G}_1\|+2\|\mathbb{G}_T\|$	$3\|\mathbb{G}_1\|+\|\mathbb{G}_T\|$
[19]	$(3+n)\|\mathbb{G}_1\|$	$(1+3n)\|\mathbb{G}_1\|+3\|\mathbb{G}_T\|$	$(5+n)\|\mathbb{G}_1\|+\|\mathbb{G}_T\|$
[25]	$(2+2n)\|\mathbb{G}_1\|$	$\|\mathbb{G}_1\|+\|\mathbb{G}_T\|$	$3\|\mathbb{G}_1\|+\|\mathbb{G}_T\|$
[26]	$(3+n)\|\mathbb{G}_1\|\|$	$(2+2n)\|\mathbb{G}_1\|+\|\mathbb{G}_T\|$	$3\|\mathbb{G}_1\|$
Our scheme	$4\|\mathbb{G}_1\|$	$(5+n)\|\mathbb{G}_1\|+\|\mathbb{G}_T\|$	$3\|\mathbb{G}_1\|+\|\mathbb{G}_T\|$

For the sake of simplicity, we use n to denote the number of attributes and the bit length of the elements. As demonstrated in Tables 2 and 3, only [12] and our scheme achieved a constant-length secret key for the delegatee. Taking into account all aspects, our scheme is still more efficient in terms of the delegatee's local storage space.

6.3. Computational Overhead

We consider only the most expensive exponentiation and pairing operations, where E represents an exponential operation on group $\mathbb{G}_1$, P represents a bilinear operation, and x represents the number of attributes. Comparing our scheme with [12,14] in terms of computational overhead, we observe that our scheme requires more computational capabilities for encrypting plaintext and generating re-encrypted plaintext. However, in our scheme, the computational capabilities required for generating the re-encryption key and decrypting the re-encrypted ciphertext are constant, whereas in scheme [12,14], the computational capabilities required are linearly related to the number of attributes. This significantly reduces the computation stress of the delegator and the delegatee in our algorithm for generating the re-encryption key and decrypting the re-encrypted ciphertext (Table 4).

Table 4. Computational overhead.

Scheme	Enc	ReKeyGen	ReEnc	DecRe
[12]	$(3+2x)E$	$(4+2x)E$	$(2+2x)P+xE$	$(1+2x)P+xE$
[19]	$(3+3x)E$	$(7+4x)E$	$(2+2x)P+xE$	$(1+2x)P+xE$
Our scheme	$(3+5x)E$	6E	$(1+3x)P+xE$	3P

6.4. Security Properties

- Collusion Resistance: By executing the ReKeyGen algorithm, it outputs $rk_{S\to ID} = \left(rk_o, rk_1, \{rk_{i,2}, rk_{i,3}\}_{i=1}^{|S|}, rk_4, rk_5, rk_6\right)$. We are aware that $rk_o = K_0\tilde{f^r}, rk_1 = K_1, rk_{i,2} = K_{i,2}, rk_{i,3} = K_{i,3}, rk_4 = F\left(e(g,g)^{\alpha\tilde{t}}\right)\tilde{g^r}, rk_5 = \left(u^{ID}h\right)^{\tilde{t}}, rk_6 = \tilde{g^t}$, if the semi-trusted proxy conspires with the delegatee; that is, the re-encryption key and the delegatee's ID are known, and it is easy for $\mathcal{A}$ to recover $\tilde{g^r}$ from $rk_{S\to ID}$ since $rk_{S\to ID}$ contains the IBE encryption of $\tilde{g^r}$ under ID. If $\mathcal{A}$ wants to obtain the private key of the delegator, it needs to find a way to obtain rk_o. However, $\mathcal{A}$ cannot obtain blinding factor part $\tilde{f^r}$.
- Verifiability: Existing cloud storage solutions lack trusted third parties, which creates a risk of malicious data deletion by delegatees. To address this issue, our scheme leverages the decentralized nature of the blockchain to provide a trusted environment for verifiable schemes. By utilizing the tamper-proof and traceability properties of the blockchain, we store the ciphertext on ciphertext chain-T. After receiving the re-encrypted ciphertext, the trustee sends ciphertext sub-item $C'_{i,4}$ to the client, and the delegator obtains verification results using the verification algorithm. The blockchain's traceability property increases the cost of dishonest re-encryption by semi-trusted agents. This approach effectively avoids the risk of the malicious tampering of data by a semi-trusted proxy or an illegal delegatee.
- Extensibility: The blockchain is a decentralized ledger. The data on the chain are generated by consensus, traceable, and cannot be deleted. To ensure that uploaded data can be checked quickly, and the scheme stores the complete ciphertext in ciphertext chain-T. The access policy, storage address, and metadata information are stored in index chain-I for easy verification and traceability. We want to ensure that we address the issue of the limited block storage capacity and prevent a single-point failure due to a large amount of data, which would result in a waste of storage space. Therefore, we combined the chord algorithm with ciphertext chain-T to extend the chain structure.
- Privacy: Our scheme provides protection for both data content privacy and delegatee identity privacy. Specifically, the ciphertext encrypted by ABE is stored in ciphertext chain-T, while index chain-I only stores the storage address and access policy. This approach enables fine-grained control and secure data sharing while protecting the delegatee's identity privacy via the use of a unique identity identifier for interactions.

6.5. Simulation Experiment

To evaluate the computational efficiency of our scheme, we compared it with schemes [12,19] and measured the performance of each based on computational overhead data. To simulate EV-ABPRE and scheme [12,19], we utilized the C++ programming language and the pairing-based cryptography library. The experiment was conducted in a virtual environment (Parallels Desktop) using a Windows 11 operating system, an Apple M1 CPU clocked at 3.2 GHz, and 8 GB of RAM. We computed the computational overhead of the three schemes in the encryption, re-encryption key generation, re-encryption, and decryption phases.

Figure 2a shows that the computational capabilities required for the encryption phase increase linearly with the number of attributes. To ensure data security, the encryption process requires an increased number of pairing operations, resulting in higher computational demands for our scheme compared to the schemes in [12,19].

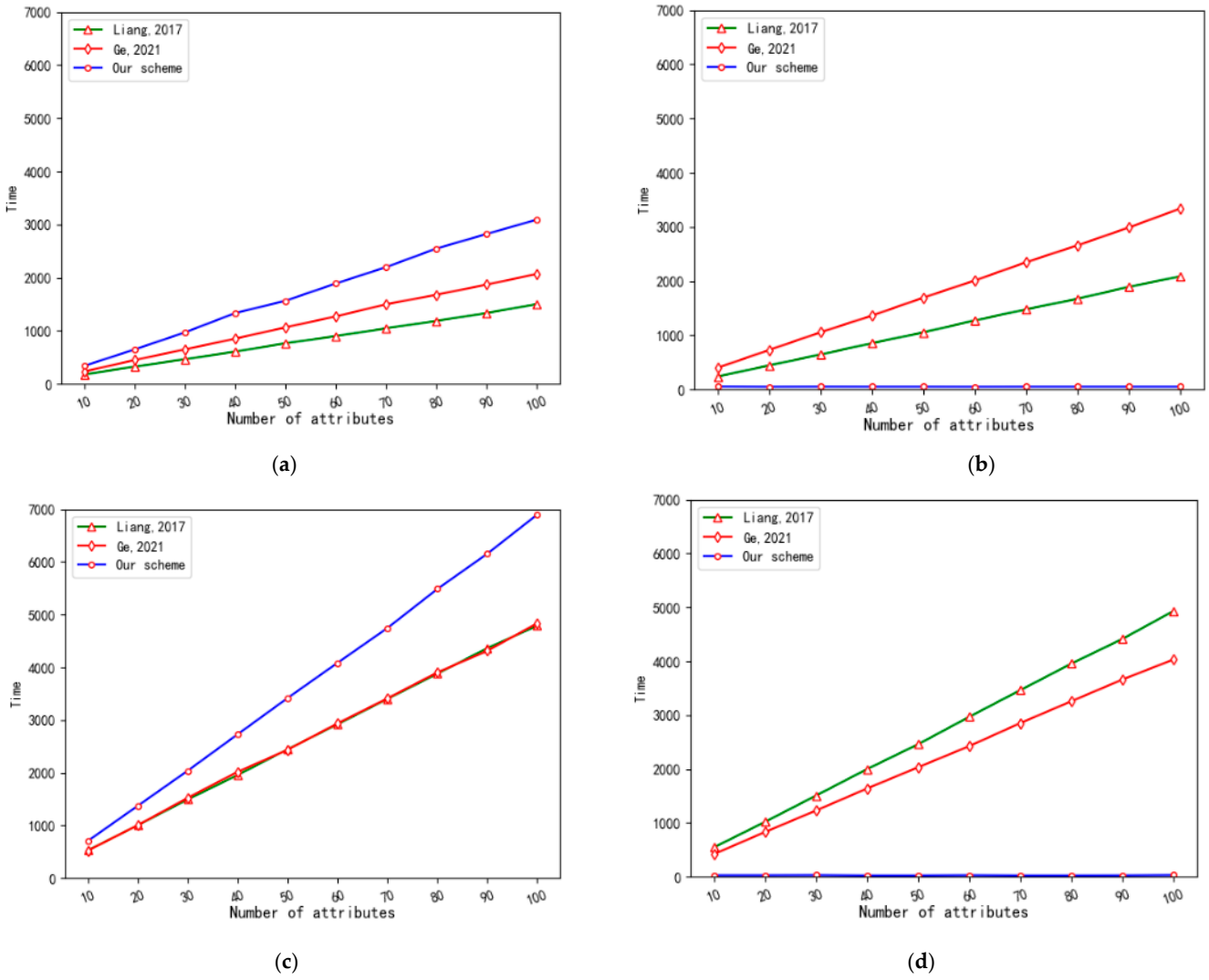

Figure 2. Computation time of our proposed EV-ABPRE scheme. (**a**) Enc overhead comparison; (**b**) ReKeyGen overhead comparison; (**c**) ReEnc overhead comparison; (**d**) DecRe overhead comparison [12,19].

Figure 2b shows that as the number of attributes increases in the schemes in [12,19], the delegator's required computational capabilities to generate the re-encryption key also increase. In contrast, the computational capabilities consumed by the delegator to generate the re-encryption key in our scheme remain constant. This advantage becomes more apparent as the number of attributes increases.

Figure 2c shows that as the number of attributes increases, the proxy nodes' requirement for computational capabilities to generate the re-encrypted ciphertext also increases. Our scheme's increased pairing operations ensure data security, but it also requires more computational capabilities compared to the schemes in [12,19].

Figure 2d shows that as the number of attributes increases in the schemes in [12,19], the delegatee's required computational capabilities to decrypt the re-encrypted ciphertext also increase. However, the computational capabilities consumed by the delegatee to decrypt the re-encrypted ciphertext in our scheme remain constant. This benefit becomes clearer as the number of attributes increases.

7. Conclusions

Our scheme combines blockchain and EV-ABPRE. On the basis of blockchain, an efficient and verifiable attribute-based proxy re-encryption cloud sharing scheme is suggested,

which realizes fine-grained and secure data sharing by using two blockchains and the encryption scheme proposed in this paper. Our scheme solves the obsolete access policy in the ABE scheme and reduces the computation costs of the delegatee during decryptions. In comparison to the traditional ABPRE scheme, it adds the function of verifying whether the ciphertext is honestly encrypted by proxy nodes. In terms of functionality, the comparison with various encryption systems, communication overhead, and computation overhead shows that the scheme has the advantage of less computational capabilities and storage space while simultaneously improving security, but it needs to add a verification ciphertext segment to the encryption process.

However, the PKG in this scheme is too large in the entire system, and this can easily become the performance bottleneck of the system. In future research, we hope to achieve efficient and safe cloud data sharing by combining the multi-attribute authorization center with this scheme. In addition, in the verifiable attribute-based proxy re-encryption scheme, the authenticity of data can be greatly protected, but compared with other similar schemes, it is more sophisticated than others in the encryption phase. The computational power consumption of EV-ABPRE will also be further improved in the encryption phase, which is also a problem we will solve in future research.

Author Contributions: T.F. participated in the feasibility discussion, analysis of the paper scheme, and the proofreading of the paper; D.W. was responsible for the overall design, performance analysis, and paper writing; R.G. supervised the formulation of the scheme and reviewed and revised the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Natural Science Foundation of China (Grant No. 62162039 and 61762060).

Data Availability Statement: The data used to support the findings of this study are included within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Goyal, V.; Pandey, O.; Sahai, A.; Waters, B. Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data. In Proceedings of the 13th ACM conference on Computer and Communications Security, Alexandria, VA, USA, 30 October–3 November 2006.
2. Deng, H.; Qin, Z.; Wu, Q.; Guan, Z.; Zhou, Y. Flexible attribute-based proxy re-encryption for efficient data sharing. *Inf. Sci.* **2020**, *511*, 94–113. [CrossRef]
3. Boneh, D. Identity-based encryption from the Weil pairing. In *Advances in Crytology, Crypto 2001*; Springer: Berlin/Heidelberg, Germany, 2001.
4. Waters, B. Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization. In Proceedings of the 14th International Conference on Practice and Theory in Public Key Cryptography, Taormina, Italy, 6–9 March 2011.
5. Norhidayah, M.; Jasni, M.Z. Access Control: Ciphertext Policy-Attribute Based Encryption in Cloud Computing. *J. Phys. Conf. Ser.* **2021**, *1830*, 012019.
6. Blaze, M.; Bleumer, G.; Strauss, M. *Divertible Protocols and Atomic Proxy Cryptography*; Springer: Berlin/Heidelberg, Germany, 1998.
7. Lang, X.; Wei, L.; Wang, X.; Wu, X. Cryptographic access control scheme for cloud storage based on proxy re-encryption. *J. Comput. Appl.* **2014**, *34*, 724.
8. Ateniese, G.; Fu, K.; Green, M.; Hohenberger, S. Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Trans. Inf. Syst. Secur.* **2006**, *9*, 1–30. [CrossRef]
9. Chen, W.; Zhu, S.; Li, J.; Wu, J.; Chen, C.; Deng, Y. Authorized Shared Electronic Medical Record System with Proxy Re-Encryption and Blockchain Technology. *Sensors* **2021**, *21*, 7765. [CrossRef] [PubMed]
10. Liang, X.; Cao, Z.; Huang, L.; Shao, J. Attribute based proxy re-encryption with delegating capabilities. In Proceedings of the International Symposium on Information, Computer, and Communications Security, Sydney, Australia, 10–12 March 2009.
11. Song, L.; Hu, J.; Zhong, C. Ciphertext Policy Attribute-Based Proxy Re-encryption. In Proceedings of the Information & Communications Security-international Conference, Barcelona, Spain, 15–17 December 2010.
12. Liang, K.; Fang, L.; Wong, D.S.; Susilo, W. A ciphertext-policy attribute-based proxy re-encryption scheme for data sharing in public clouds. *Concurr. Comput.* **2015**, *27*, 2004–2027. [CrossRef]
13. Hong, H.; Liu, X.; Sun, Z. A Fine-Grained Attribute Based Data Retrieval with Proxy Re-Encryption Scheme for Data Outsourcing Systems. *Mob. Netw. Appl.* **2018**, *26*, 2509–2514. [CrossRef]

14. Luo, F.; Al-Kuwari, S.; Wang, F.; Chen, K. Attribute-based proxy re-encryption from standard lattices. *Theor. Comput. Sci.* **2021**, *865*, 52–62. [CrossRef]
15. Yang, G.; Guo, R.; Zhuang, C.; Wang, X. Dynamically Updatable Attribute Based Proxy Re-encryption Scheme in Cloud. *J. Cyber Secur.* **2022**, *7*, 43–55.
16. Hong, H.; Sun, Z. Sharing your privileges securely: A key-insulated attribute based proxy re-encryption scheme for IoT. *World Wide Web* **2018**, *21*, 595–607. [CrossRef]
17. Lai, J.; Deng, R.H.; Guan, C.; Weng, J. Attribute-Based Encryption with Verifiable Outsourced Decryption. *IEEE Trans. Inf. Forensics Secur.* **2013**, *8*, 1343–1354.
18. Lin, S.; Zhang, R.; Wang, M. Verifiable attribute-based proxy re-encryption for secure public cloud data sharing. *Secur. Commun. Netw.* **2016**, *9*, 1748–1758. [CrossRef]
19. Ge, C.; Susilo, W.; Baek, J.; Liu, Z.; Xia, J.; Fang, L. A Verifiable and Fair Attribute-based Proxy Re-encryption Scheme for Data Sharing in Clouds. *IEEE Trans. Dependable Secur. Comput.* **2021**, *19*, 2907–2919. [CrossRef]
20. Zuo, Y.; Kang, Z.; Xu, J.; Chen, Z. BCAS: A blockchain-based ciphertext-policy attribute-based encryption scheme for cloud data security sharing. *Int. J. Distrib. Sens. Netw.* **2021**, *17*, 1550147721999616. [CrossRef]
21. Eltayieb, N.; Sun, L.; Wang, K.; Li, F. A Certificateless Proxy Re-encryption Scheme for Cloud-Based Blockchain. In *Frontiers in Cyber Security, Proceedings of the Second International Conference on Frontiers in Cyber Security, FCS 2019, Xi'an, China, 15–17 November 2019*; Springer: Berlin/Heidelberg, Germany, 2019.
22. Zhang, X.; Sun, L. Attribute Proxy Re-encryption for Ciphertext Storage Sharing Scheme on Blockchain. *J. Syst. Simul.* **2020**, *32*, 1009–1020.
23. Beimel, A. Secure Schemes for Secret Sharing and Key Distribution. Ph.D. Thesis, Technion-Israel Institute of Technology, Haifa, Israel, 1996.
24. Sun, Z.; Zhang, X.; Xiang, F.; Chen, L. Survey of Storage Scalability on Blockchain. *J. Softw.* **2021**, *32*, 1–20.
25. Tiwari, D.; Gangadharan, G.R. SecCloudSharing: Secure data sharing in public cloud using ciphertext-policy attribute-based proxy re-encryption with revocation. *Int. J. Commun. Syst.* **2017**, *31*, e3494. [CrossRef]
26. Zhai, S.; Tong, T.; Bai, X. Blockchain-based attribute proxy re-encryption data sharing scheme. *Comput. Eng. Appl.* **2023**, *59*, 270–279.