sensors

MDPI

Article

Decentralized Policy-Hidden Fine-Grained Redaction in Blockchain-Based IoT Systems

Hongchen Guo [1], Xiaolong Tao [2], Mingyang Zhao [2], Tong Wu [2,*], Chuan Zhang [2], Jingfeng Xue [1] and Liehuang Zhu [2]

[1] School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China; guohongchen@bit.edu.cn (H.G.); xuejf@bit.edu.cn (J.X.)
[2] School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing 100081, China; ac_xiaolongt@163.com (X.T.) mingyangz@bit.edu.cn (M.Z.); chuanz@bit.edu.cn (C.Z.); liehuangz@bit.edu.cn (L.Z.)
* Correspondence: tracy_tongw@163.com

Abstract: Currently, decentralized redactable blockchains have been widely applied in IoT systems for secure and controllable data management. Unfortunately, existing works ignore policy privacy (i.e., the content of users' redaction policies), causing severe privacy leakage threats to users since users' policies usually contain large amounts of private information (e.g., health conditions and geographical locations) and limiting the applications in IoT systems. To bridge this research gap, we propose PFRB, a policy-hidden fine-grained redactable blockchain in decentralized blockchain-based IoT systems. PFRB follows the decentralized settings and fine-grained chameleon hash-based redaction in existing redactable blockchains. In addition, PFRB hides users' policies during policy matching such that apart from successful policy matching, users' policy contents cannot be inferred and valid redactions cannot be executed. Some main technical challenges include determining how to hide policy contents and support policy matching. Inspired by Newton's interpolation formula-based secret sharing, PFRB converts policy contents into polynomial parameters and utilizes multi-authority attribute-based encryption to further hide these parameters. Theoretical analysis proves the correctness and security against the chosen-plaintext attack. Extensive experiments on the FISCO blockchain platform and IoT devices show that PFRB achieves competitive efficiency over current redactable blockchains.

Keywords: blockchain-based IoT systems; fine-grained redaction; policy hiding; decentralization

Citation: Guo, H.; Tao, X.; Zhao, M.; Wu, T.; Zhang, C.; Xue, J.; Zhu, L. Decentralized Policy-Hidden Fine-Grained Redaction in Blockchain-Based IoT Systems. *Sensors* **2023**, *23*, 7105. https://doi.org/10.3390/s23167105

Academic Editor: Jian Li

Received: 9 July 2023
Revised: 31 July 2023
Accepted: 9 August 2023
Published: 11 August 2023

1. Introduction

The Internet of Things (IoT) is defined as connecting all objects through information-sensing devices such as radio frequency identification to the Internet, enabling intelligent recognition and management. The concept of IoT entails integrating sensors into various objects, including power grids and railways, enabling data collection and communication for seamless connectivity and interaction with the physical world [1–3]. However, due to the inherent decentralization of IoT device deployment [4,5], achieving secure data management poses challenges.

To address these challenges, blockchain has been widely applied in IoT systems as a decentralized data management platform, providing traceability and integrity to secure IoT systems [6–8]. According to a report by Morder Intelligence (https://www.mordorintelligence.com/industry-reports/blockchain-iot (accessed on 31 July 2023)), the market size of blockchain-based IoT systems is expected to grow from USD 568.51 million in 2023 to USD 3436.54 million by 2028. Despite these benefits, immutable blockchains present two limitations in current blockchain-based IoT systems. Firstly, the immutability of blockchain violates the right to be forgotten in the General Data Protection Regulation (GDPR) (https://gdpr-info.eu/art-17-gdpr/ (accessed on 31 July 2023)), severely limiting the development and implementation of blockchain-based IoT systems. GDPR mandates

that users should be able to erase their personal data (e.g., health and transportation data) from data management systems. However, with immutable blockchains, once data are appended to the blockchain, no modifications can be made [9]. This conflict between immutability and GDPR could result in high fines for blockchain-based IoT systems. Secondly, IoT devices are vulnerable to network attacks and can be used to spread improper information (e.g., indecent surveillance videos) in blockchain-based IoT systems. If this improper information cannot be redacted, it may negatively impact the blockchain-based IoT application ecology [10–12].

To address these limitations, some redactable blockchains have been proposed, adopting chameleon hashes [13–18]. Specifically, in 2017, Ateniese et al. [13] introduced the first redactable blockchain by using the chameleon hash technique. Their work extended researchers' insight into immutable blockchains and made blockchains compliant with data regulation (e.g., GDPR). Unfortunately, their work only supports all-or-nothing redaction privileges, i.e., they either cannot redact any data or can redact all data. Clearly, it is infeasible in practice since real-world blockchain applications contain a large number of devices that own different attributes. To achieve fine-grained redaction, Derler et al. [13], Ma et al. [18], and Xu et al. [19] utilized attribute-based fine-grained access control and chameleon hash to introduce the policy-based chameleon hash technique (PCH), subsequently using PCH to achieve fine-grained redactable blockchains. Following PCH, Mao et al. [18] further introduced a decentralized policy-based chameleon hash technique (DPCH) and used it to design a decentralized fine-grained redactable blockchain. However, in existing fine-grained redactable blockchains, users' policies are public to all users, making them unsuitable for some policy-sensitive IoT systems, such as IoT-based smart healthcare and smart transportation [20]. In these IoT systems, users' policies contain sensitive private information, e.g., users' health conditions and geographical locations [21]. For instance, in an IoT-based smart healthcare application, users use their IoT devices (e.g., smartwatches) to record their health data and utilize blockchain-based IoT systems to manage the data. In the application, users' policies usually contain information about users' health conditions, such as sensitive health details [22]. Obviously, if this private information is leaked, users may face discrimination and even potential attacks [23]. Therefore, there is a great need to design decentralized policy-hidden redactable blockchains in blockchain-based IoT systems. We make a comparison between our proposed solution and existing works, as shown in Table 1. We address the issue of policy disclosure in a decentralized, fine-grained setting.

Table 1. Comparison between PFRB and existing redactable blockchain schemes.

	Fine-Grained	**Decentralization**	**Policy-Hidden**
AMVA17 [13]	×	×	×
DSSS19 [16]	✓	×	×
DMT19 [14]	×	✓	×
TLL20 [24]	✓	×	×
PVM21 [25]	✓	×	×
JSZ21 [26]	×	×	×
XNMX21 [27]	✓	×	×
XNMX22 [15]	✓	×	×
RBDS22 [18]	✓	✓	×
Ours	✓	✓	✓

To solve the above problem, we propose a policy-hidden fine-grained redactable blockchain scheme (named PFRB) in decentralized blockchain-based IoT systems. In PFRB, users can enjoy blockchain services while hiding their policies. A technical challenge in designing PFRB is how to hide policy contents and support policy matching. PFRB

draws inspiration from Newton's interpolation formula-based secret sharing and gracefully converts policy contents into polynomial parameters. Particularly, we summarize the main contributions as follows:

- We propose a policy-hidden fine-grained redactable blockchain scheme (named PFRB) for blockchain-based IoT systems. With decentralized settings, PFRB enables users to achieve fine-grained data redaction without compromising policy privacy.
- PFRB leverages multi-authorized attribute-based encryption and Newton's interpolation formula-based secret sharing to construct a decentralized secret sharing for policy hiding. Then, based on the constructed secret sharing, PFRB further enriches chameleon hashes to achieve decentralized policy-hidden fine-grained redactable blockchains.
- Security analysis proves the security of PFRB under the chosen-plaintext attack in the random oracle model. Experimental results show that PFRB has competitive efficiency over recent fine-grained redactable blockchain schemes.

Organization. The remainder of this paper is structured as follows. In the next section, we review existing works on redactable blockchains to introduce the research gap. In Section 3, we provide an introduction to the preliminary information, followed by an overview of the PFRB system in Section 4. Next, in Section 5, we provide insight into the detailed construction of PFRB. Section 6 presents a formal correctness analysis and security analysis. Performance evaluation of PFRB is shown in Section 7. Finally, concluding remarks are given in Section 8.

2. Related Work

This section provides a systematic review of the current literature on redactable blockchain techniques and introduces the research gap.

In 2017, Ateniese et al. [13] introduced the first redactable blockchain by using the chameleon hash technique. Their work extended researchers' insight into immutable blockchains and made blockchains compliant with the data regulation (e.g., GDPR and CCPA). Unfortunately, their work only supports all-or-nothing redaction privileges, i.e., they either cannot redact any data or can redact all data. Clearly, it is infeasible in practice since real-world blockchain applications contain a large number of devices that own different attributes.

To address the above limitation, Derler et al. [16] proposed a policy-based chameleon hash (PCH) and used PCH to implement fine-grained redactable blockchains. PCH combines chameleon hashes, ephemeral trapdoors, and linear secret sharing matrix-based attribute-based encryption to associate transactions with access policies, enabling rewriting only when editors' attributes satisfy the policy. Followed by Derler et al.'s work, Tian et al. [24] and Xu et al. [19] extent redactable blockchains with accountability by introducing digital signatures, respectively. Due to the utilization of centralized attribute-based encryption schemes, the above schemes can only be applied in consortium blockchains, and cannot support generally decentralized settings. However, in practical blockchain applications, especially distributed IoT systems, decentralized systems are more general. To achieve decentralized redactable blockchains, Ma et al. [18] introduced the first decentralized policy-based chameleon hash (DPCH) by linear secret sharing matrix-based multi-authority attribute-based encryption and used DPCH to achieve decentralized fine-grained blockchain redaction.

However, redaction policies in the above redactable blockchains are open to all participants, which is infeasible in practical blockchain-based IoT systems since policies in IoT systems usually contain users' private information. For instance, in IoT-based smart healthcare applications, patients use their smart swatches to collect their health conditions (such as heart rate and personal temperature) and specify policies to allow their private doctors to edit. Evidently, these policies usually contain sensitive information regarding patients' health conditions and geographical locations. Once this private information is leaked, adversaries can launch attacks on these patients such as robbery and discrimination.

Thus, the protection of policy privacy in decentralized redactable blockchains represents an urgent matter.

3. Preliminary

In this section, we introduce the building blocks of PFRB, i.e., multi-authority attribute-based encryption, Newton's interpolation formula-based secret sharing, and chameleon hash.

3.1. Multi-Authority Attribute-Based Encryption

A multi-authority attribute-based encryption (MA-ABE) [28] system consists of arbitrary numbers of attribute authorities and users. A set of global public parameters is defined in the system. Users can select an attribute authority and obtain their corresponding decryption key. The authorization authority performs the appropriate attribute key generation algorithm and returns the result to the user. The encryption process uses the global public parameters and a set of attributes to generate the ciphertext. The decryption process uses the decryption key for the attribute set to perform decryption.

Definition 1 (MA-ABE): A multi-authority attribute-based encryption ABE_{MC} involves three types of entities: authorities, data owners, and data users. It includes five algorithms:

- Global Setup $(\lambda) \rightarrow (GP)$: This algorithm accepts a secure parameter λ as input and produces a public global parameter GP as output.
- Authority Setup $(GP) \rightarrow (PK, SK)$: In this step, the algorithm takes the public global parameter GP as input and generates a public key PK and a secret key SK as output. It is crucial to keep the secret key SK confidential, while the public key PK is intended for publication.
- Encryption $(M, (A, \rho), GP, \{PK\}) \rightarrow (CT)$: The algorithm accepts several inputs, including a message M, an $n \times \ell$ access matrix A with ρ mapping its rows to attributes, the global parameter GP, and the public keys of the relevant authorities PK. It then produces a ciphertext CT as output.
- KeyGen $(ID, i, SK, GP) \rightarrow (K_{i,ID})$: The algorithm generates a key $K_{i,ID}$ for attribute i associated with an authority using the inputs: a global identifier ID, the attribute i, the secret key SK, and the public global parameter GP.
- Decryption $(CT, \{K_{i,ID}\}, GP) \rightarrow (M)$: The algorithm decrypts the ciphertext CT using the input parameters: the key $K_{i,ID}$ for ID and attribute i, as well as the global parameter GP. The result of the decryption process is the message M.

3.2. Newton's Interpolation Formula-Based Secret Sharing

In this paper, Newton's interpolation formula is primarily used for key recovery. Also, due to the introduction of polynomial-oriented secret sharing, PFRB achieves higher efficiency than traditional linear secret sharing matrix-based works. The transaction issuer hides the key within the zeroth term of a polynomial, ensuring that only users who meet the policy requirements can reconstruct the polynomial and access the hidden key.

- Secret Generation: Assume that there are $(n+1)$ points represented as (x_0, y_0), (x_1, y_1), $\ldots$, (x_n, y_n). Here, x_i is called the interpolation point, and y_i is called the interpolation value. Given an interpolation polynomial $f(x)$, for each $i = 0, 1, 2, \ldots, n$, y_i is represented as $y_i = f(x_i)$. The Newton's basis $n_i(x)$ is defined as follows:

$$n_i(x) = \begin{cases} 1, & \text{if } i = 0, \\ \prod_{j=0}^{i-1}(x - x_j), & \text{otherwise.} \end{cases}$$

Based on $n_i(x)$, the Newton's interpolation polynomial $Q_n(x)$ can be defined as follows:

$$Q_n(x) = K_0 + K_1(x - x_0) + \cdots + K_n \prod_{i=0}^{n-1}(x - x_i) = \sum_{i=0}^{n} K_i n_i(x).$$

Specifically, based on x_0, $Q_n(x)$ can be estimated as follows:

$$Q_n(x_0) = \sum_{i=0}^{n} K_i n_i(x) = K_0 = f(x_0) = f[x_0].$$

where $f[x_0]$ represents the zeroth order divided difference. Similarly, based on x_1, the Newton's interpolation polynomial $Q_n(x)$ is estimated as follows:

$$Q_n(x_1) = K_1(x - x_0) + K_0 = K_1(x - x_0) + f[x_0] = f[x_1].$$

Thus, K_1 can be estimated as follows:

$$K_1 = f[x_0, x_1] = \frac{f[x_1] - f[x_0]}{x_1 - x_0}.$$

where $f[x_0, x_1]$ represents the first order divided difference. Without loss of generality, K_i can be defined as follows:

$$K_i = \frac{f[x_1, x_2, \ldots, x_i] - f[x_0, x_1, \ldots, x_{i-1}]}{x_i - x_0}.$$

where $f[x_1, x_2, \ldots, x_i]$ denotes the i-th order divided difference, respectively. For more details, the reader can refer to previous literature.

- Secret Construction: We can reconstruct the secret with Newton's parameters as follows:

$$s = Q_n(0) = \sum_{i=0}^{n} K_i n_i(0).$$

3.3. Chameleon Hash

A chameleon hash $\mathcal{CH}$ typically encompasses the following five algorithms:

- Setup$(1^\lambda) \rightarrow pp$: The probabilistic setup algorithm takes a security parameter λ as input and generates a public parameter pp as output. The public parameter pp is used in subsequent algorithms and protocols to ensure the security and functionality of the system.
- KeyGen$(pp) \rightarrow (pk, sk)$: The probabilistic key generation algorithm takes the public parameter pp as input and generates a public-secret key pair (pk, sk) as output. The public key pk is used for encryption or other public operations, while the secret key sk is kept confidential and used for decryption or other sensitive operations.
- Hash$(pk, m) \rightarrow (h, r)$: The probabilistic hash algorithm takes the public key pk and a message $m \in \mathcal{M}$ as input. It then produces an output of 1 if the tuple (h, r) is considered valid according to the algorithm's criteria. If the tuple is not valid, the output will be 0.
- Verify$(pk, m, h, r) \rightarrow \{0, 1\}$: The deterministic verification algorithm takes the public key pk, message m, hash value h, and randomness value r as input. It then determines whether the tuple (h, r) is valid according to the defined criteria. If the tuple is valid, the algorithm outputs 1. Otherwise, if the tuple is not valid, the output will be 0.
- Adapt$(sk, m, m', h, r) \rightarrow r'$: The deterministic adaptation algorithm takes the secret key sk, message $m \in \mathcal{M}$, hash value h, and randomness value r as input. It then generates an adapted randomness value r' as output.

4. System Overview

In this section, we present the system model, brief definition, and security model of PFRB.

4.1. System Model

As shown in Figure 1, the PFRB (Privacy-Preserving Redactable Blockchain) system involves four types of entities:

- Authorities: The authorities are all trusted. One of them initializes the system, and they can all generate attribute-value pairs.
- Transaction Owner: The transaction owner is also trusted and wants to place a deal or some data on the blockchain. They hash the data and attempt to add the transaction to the blockchain.
- Transaction Modifier: The transaction modifier is a user who wants to modify a transaction in the blockchain. They retrieve the attribute-value pairs and try to match the transaction to modify it.
- Blockchain Participants: The blockchain participants are users of the redactable blockchain. They verify each transaction published by the transaction owners or the transaction modifiers.

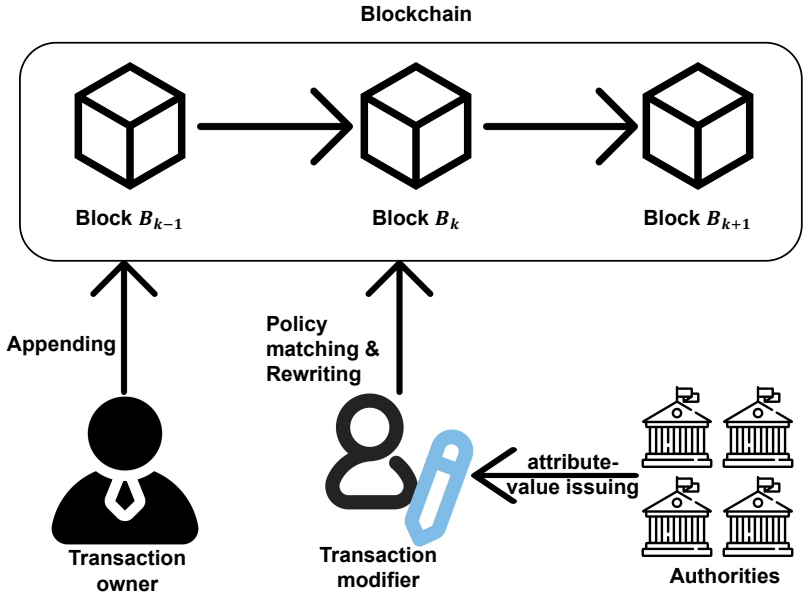

Figure 1. PFRB system model.

4.2. Definition of PFRB

We next provide a brief definition of PFRB and summarize the notations in Table 2.

- Setup $(1^\lambda) \rightarrow (pp_s, pk_s, sk_s)$: Given a security parameter, the Setup algorithm outputs a public parameter pp_s, public key pk_s, and secret key sk_s. Then, the authority publishes pp_s and pk_s to all users.
- RKGen $(msk, \rho) \rightarrow (dk_\rho, \{\Delta_{i,R_j}\})$: The RKGen algorithm takes msk and ρ as input, where ρ is the attribute set of the modifier. The algorithm outputs the decryption key dk_ρ and the Lagrange coefficients of ρ, $\{\Delta_{i,R_j}\}$.
- ModSetup $(sk_s, id) \rightarrow (sk_{id}, \sigma_{id})$: The ModSetup algorithm takes the secret key sk_s and the global identifier id as input. It outputs the modifier's secret key sk_{id} and the modifier's signature σ_{id}.
- AuthSetup $(\theta) \rightarrow (pk_\theta, sk_\theta)$: The AuthSetup algorithm takes the authority θ as input and outputs the authority's public key pk_θ and secret key sk_θ.
- ModKeyGen $(pk_s, id, \sigma_{id}, sk_\theta, A) \rightarrow sk_{id,A} / \perp$: The ModKeyGen algorithm takes the public key pk_s, the modifier's global identifier id, the modifier's signature σ_{id}, the authority's secret key sk_θ, and an attribute A as input. It generates the secret key $sk_{id,A}$ for the modifier's attribute A if the request is legal; otherwise, it outputs nothing.
- Hash $(pk_s, \{pk_\theta\}, m, R) \rightarrow (pk^{etd}, h, r, c)$: The Hash algorithm is designed to take the following inputs: the public key pk_s, a group of authorities' public keys pk_θ, the message

m to be encrypted, and the policy R of the target receiver. It generates four outputs: a public key pk^{etd} (a public component of the ephemeral trapdoor), a hash value h, a randomness value r, and a ciphertext c. The ciphertext c plays the crucial role of securely sealing the secret component sk^{etd}, guaranteeing its confidentiality.

- Verify $(pk_s, pk^{etd}, m, h, r) \rightarrow \{0,1\}$: The Verify algorithm can be executed by any entity within the system. It accepts the following inputs: the public key pk_s, the public component pk^{etd} of the ephemeral trapdoor, the message $m \in M$, the hash value h, and the randomness value r. The algorithm then determines whether the tuple (h, r) is valid according to its defined criteria. If the tuple is deemed valid, the algorithm outputs 1. However, if the tuple is found to be invalid, the output will be 0.
- Adapt $(sk_{id}, \{sk_{id,A}\}, c, m, m', h, r) \rightarrow \mathrm{r}'$: The Adapt algorithm is executed by the transaction modifier. It takes inputs such as the secret component sk^{etd}, a set of secret keys $sk_{id,A}$, the ciphertext c, messages m and m', the hash value h, and the randomness value r. The output of the algorithm is a new randomness value r'.

Table 2. Notations used in PFRB.

Notations	Descriptions	Notations	Descriptions
(p, G, G_0, e, g)	description of bilinear group	λ	security parameter
(n, p, q, e, d)	description of RSA parameter	$H_{ID,G}, H_{U,G}, H_0, H_{b,Z_p}$	description of hash function
$e(g, g)$	description of the bilinear function	pp_s	public parameter
pk_s	public key	sk_s	secret key
msk	master secret key	ρ	an attribute set of modifier
dp_ρ	the decryption key	Δ_{i,R_j}	Lagrange coefficient of ρ
id	global identifier	sk_{id}	modifier's secret key
σ	modifier's signature	θ	an authority
(pk_θ, sk_θ)	authority's own public key the secret key	A	an attribute
$sk_{id,A}$	the secret key of the modifier's attribute A	sk^{etd}	the secret component of the ephemeral trapdoor
R	the target receiver of the transaction	pk^{etd}	public component of an ephemeral trapdoor

4.3. Security Model

In our scheme, we assume that all authorities and the data owner are trusted entities, and communications between them are secure. The transaction owner generates mutable transactions honestly, and the authorities preserve the secret key honestly. However, other entities, such as chain participants, can act as adversaries and collaborate to launch the chosen-plaintext attack. The security of PFRB is defined as the indistinguishability and the collision resistance under the chosen-plaintext attack in the random oracle model as follows.

- Setup: The challenger runs the Setup algorithm and shares the public parameters PK with the adversary.
- Phase 1: The challenger allows the adversary to request private keys from the encryption oracle $\mathcal{O}_E$ by their attributes $S_1, \ldots, S_{q1}$.
- Challenge: The adversary selects and uploads two messages, M_0 and M_1, of equal length. The adversary also presents a challenge access structure, denoted as A, which none of the previously generated attribute sets can satisfy. The challenger randomly chooses a coin flip outcome, encrypts either M_0 or M_1 under the challenge access structure A, and provides the resulting ciphertext CT^* to the adversary.
- Phase 2: Phase 1 is repeated, but with the additional constraint that none of the sets of attributes $S_{q1+1}, S_{q1+2}, \ldots, S_q$ satisfy the access structure associated with the given challenge. This restriction ensures that the adversary cannot find any new sets of attributes that fulfill the challenge access structure.
- Guess: Based on the above experiment, the adversary outputs a guess, b_0, of b.

We say that the adversary $\mathcal{A}$ wins the above game if the guess $b^{'}$ equals b. Specifically, PFRB is secure against the chosen-plaintext attack (CPA) if any probabilistic polynomial-time adversary $\mathcal{A}$ only has a negligible advantage to win the game as follows.

$$\mathsf{Adv}_{\mathcal{A}}^{CPA}(\lambda) = |\mathsf{Pr}[b' = b|\mathcal{A}^{\mathcal{O}_E}(M_0, M_1, CT^*)]|.$$

5. Proposed Scheme

In this section, we present detailed construction of PFRB. The workflow of PFRB is shown in Figure 2. There are eight algorithms: Setup, RkGen, ModSetup, AuthSetup, ModKeyGen, Hash, Verify and Adapt.

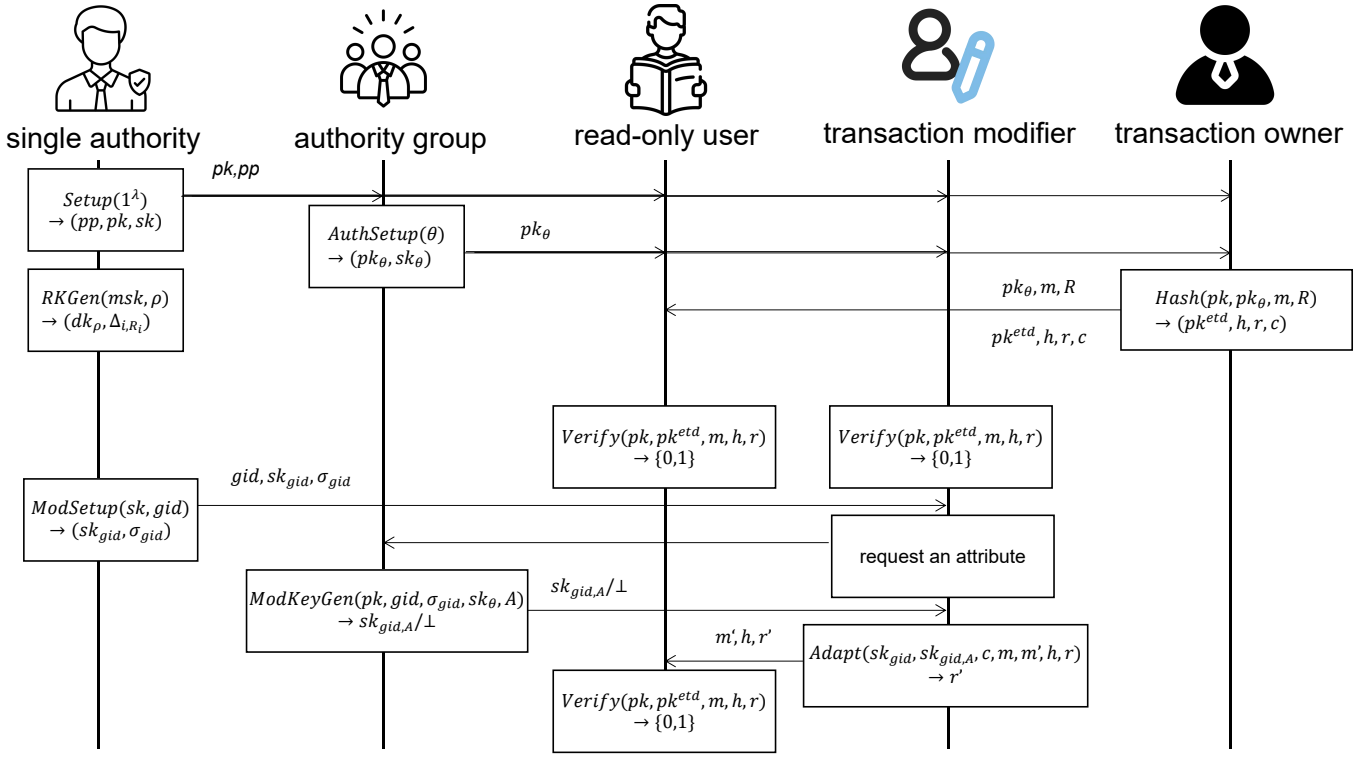

Figure 2. Detailed workflow in PFRB.

- Setup $(1^\lambda) \rightarrow (pp_s, pk_s, sk_s, mpk, msk)$:
  - Given a security parameter λ, generate the bilinear group description (p, G, G_0, e, g).
  - Running $RSAKeyGen(1^\lambda)$, and then get the first set of RSA parameter (n_0, p_0, q_0, e_0, d_0)
  - Choose three random exponents $\alpha \in Z_p, \beta, \gamma \in G$, a mapping function, and four hash functions as follows.

$$H_{b,Z_p} : \{0,1\} \rightarrow Z_p,$$
$$H_{ID,G} : \{0,1\} \times ID \rightarrow G,$$
$$H_0 : \{0,1\}^* \rightarrow Z_n^*,$$
$$H_{U,G} : U \rightarrow G,$$
$$F_m : U \rightarrow U_\theta.$$

  - Set $Y_\beta = e(g,g)^\beta, Y_\gamma = e(g,g)^\gamma$. Choose n, d and $2n+2$ random values $t_1, \ldots, t_{n+1}$, $d_1, \ldots, d_{n+1} \in Z_p$ and set $T_i = g^{t_i}, D_i = g^{d_i}$ for each i from 1 to $n+1$.
  - Choose a symmetric encoding method $F_{en} : \{0,1\}^* \rightarrow G_0$, and the corresponding decoding method $F_{en}^{-1} : G_0 \rightarrow \{0,1\}^*$.

– Calculate the system public parameter pp_s, master secret key msk, master public key mpk, system public key pk_s, and system secret key sk_s as:

$$pp_s = (1^\lambda, p, G, G_0, e, g, H_{b,Z_p}, H_{ID,G}, H_0, H_{U,G}, F_m, F_{en}, F_{en}^{-1}),$$
$$mpk = (Y_\beta, Y_\gamma, \{T_i\}, \{D_i\}), msk = (\beta, \gamma),$$
$$pk_s = (H_0, n_0, e_0, g^\alpha), sk_s = (d_0, \alpha).$$

The function $H_{ID,G}$ receives a bit $b \in \{0,1\}$ and an input $id \in ID$, producing a hash value in G. In our system, the attributes are named using the format "[attribute-id]@[authority-id]". To extract only the authority ID from the attribute name, we use a mapping function called F_m. This function helps us retrieve the authority ID while ignoring the attribute ID.

- RKGen $(msk, \rho) \to (dk_\rho, \{\Delta_{i,R_j}\})$: ρ is an attribute set of modifier.
  - Randomly generate a d-1 degree polynomial q_2 with $q_2(0) = \gamma$
  - Calculate the values of $dk_i = g^{\frac{q_2(i)}{d_i}}$ for each i in ρ, and store them as $dk_\rho = \{dk_i\}$.
  - Compute a set of Lagrange coefficients Δ_{i,R_j}, where each Lagrange coefficient might satisfies the policy, and R_j belongs to ρ.
- ModSetup $(sk_s, id) \to (sk_{id}, \sigma_{id})$:
  - The authorities compute their secret key $sk_{id} = d_0$ and their signature $\sigma_{id} = H_{ID,G}(1, id)^\alpha$. Return sk_{id} and σ_{id}.
- AuthSetup $(\theta) \to (pk_\theta, sk_\theta)$:
  - The authority Chooses two random values $a_\theta, b_\theta \in Z_p$. Calculate the secret key sk_θ as (a_θ, b_θ) and public key pk_θ as $(e(g,g)^{a_\theta}, g^{b_\theta})$. Return pk_θ and sk_θ.
- ModKeyGen $(pk_s, id, \sigma_{id}, sk_\theta, A) \to sk_{id,A} / \perp$
  - If $e(g, \sigma) \neq e(g^\alpha, H_{ID,G}(1, id))$, return $\perp$.
  - Generate a random value $t \in Z_p$. Compute $sk_{id,A,0} = g^{a_\theta} H_{ID,G}(0, id)^{b_\theta} H_{U,G}(A)^t$ and $sk_{id,A,1} = g^t$.
  - Output the corresponding secret key of the attribute A as follows.

$$sk_{id,A} = (id, A, sk_{id,A,0}, sk_{id,A,1}).$$

- $Hash(pk_s, \{pk_\theta\}, m, R) \to (pk^{etd}, h, r, c)$: R is a policy of target receiver.
  - Run the RSA key generator RSAKeyGen(1^λ) and generate another set of RSA parameter$(n_1, p_1, q_1, e_1, d_1)$.
  - Choose $H_1 : \{0,1\}^* \to Z_{n_1}^*, r_0 \in Z_{n_0}^*$ and $r_1 \in Z_{n_1}^*$. Then compute two hash values $h_0 = H_0(m) r_0^{e_0}$ and $h_1 = H_1(m) r_1^{e_1}$
  - Choose a random sequence $r_t \in \{0,1\}^\lambda$. Then, run $SE.KeyGen(1^\lambda)$ to generate a key k. Subsequently, utilize k to generate a ciphertext $c_t \leftarrow SE.Enc(k, d_1)$.
  - Run the symmetric encryption algorithm and compute $k_c \leftarrow F_{en}(k, r_t)$ and $z \leftarrow H_{b,Z_p}(r_t, G_n(x))$. Calculate $c_0 = k_c e(g,g)^z$. Choose a set of random numbers $u_r, t_r, r_{1,r}, r_{2,r} \in Z_p$. Compute

$$T_r = g^{t_r}, U_r = g^{u_r}, R_{1,r} = g^{r_{1,r}}, R_{2,r} = g^{r_{2,r}}.$$

Compute $P_{i,r} = D_{i,r}^{r_{1,r}}$, $E_{i,r} = T_{i,r}^{r_{2,r}}$ where $i \in R$.

  - Compute

$$K_{1,r} = e(R_{1,r}, T_r) Y_\gamma^{r_{1,r}},$$
$$K_{2,r} = e(R_{2,r}, U_r) Y_\beta^{r_{2,r}},$$
$$V_r = z \oplus H(e(R_{1,r}, T_r)) \oplus H(e(R_{2,r}, U_r)).$$

– Return the public key pk^{etd}, random value r, hash value h, and ciphertext c as follows.

$$pk^{etd} = (H_1, n_1, e_1), h = (h_0, h_1), r = (r_0, r_1),$$
$$c = (c_0, T_r, U_r, \{P_{i,r}\}, \{E_{i,r}\}, W_r, \{K_{1,r}\}, \{K_{2,r}\}, V_r).$$

- Verify $(pk_s, pk^{etd}, m, h, r) \rightarrow \{0, 1\}$:Parse $h = (h_0, h_1)$ and $r = (r_0, r_1)$
  - Return 1 if $h_0 = H_0(m)r_0^{e_0}\ mod\ n_0$ and $h_1 = H_1(m)r_1^{e_1}\ mod\ n_1$; otherwise, return 0;
- Adapt$(sk_{id}, \{sk_{id,A}\}, c, m, m', h, r) \rightarrow r'$:
  - If m equals m', then output $r' = r$.
  - Enumerate all sets of combinations of attributes and compute

$$g_{T,1} = K_1 / \prod_{i,R_j} e(P_i, dk_i)^{\Delta_{i,R_j}(0)}, with\ R_j \subset R, |R_j| \geq d,$$
$$g_{T,2} = K_2 / e(\prod_{i,S} E_{i,r}, W_r),$$
$$z = V \oplus H(g_{T,1}) \oplus H(g_{T,2}),$$
$$k_c = c_0 / e(g, g)^z, (k, r_t) \leftarrow F_{en}^{-1}(k_c).$$

  - Run $d_1 \leftarrow SE.Dec(c_t, k)$. Compute

$$r'_0 = (h_0(H_0(m')^{-1}))^{d_0} mod\ n_0,$$
$$r'_1 = (h_1(H_1(m')^{-1}))^{d_1} mod\ n_1.$$

  - Return the randomness $r' = (r'_0, r'_1)$.

6. Theoretical Analysis

In this section, we theoretically analyze the correctness and security of PFRB. Then, we discuss some promising applications of PFRB.

6.1. Correctness Analysis

In this section, we will provide detailed proof of the correctness of the proposed scheme in this paper. The scheme presented in this paper builds upon Ma et al's scheme (i.e., RBDS22) [18] while incorporating additional improvements. It is worth noting that if RBDS22 is correct and the calculations of $K_{1,r}$ and $K_{2,r}$ in this scheme are correct, then the proposed scheme in this paper is also correct. We will now present the specific proof of the correctness of $K_{1,r}$ and $K_{2,r}$ as follows:

In the Hash part, we have $K_{1,r} = e(R_{1,r}, T_r)Y_\gamma^{r_{1,r}}$,$K_{2,r} = e(R_{2,r}, U_r)Y_\beta^{r_{2,r}}$. $K_{1,r}$ can be transformed as follows:

$$K_{1,r} = e(R_{1,r}, T_r)Y_\gamma^{r_{1,r}} = e(R_{1,r}, T_r)e(g, g)^{\gamma, r_{1,r}}.$$

Similarly, $K_{2,r}$ can be transformed as follows:

$$K_{2,r} = e(R_{2,r}, U_r)Y_\beta^{r_{2,r}} = e(R_{2,r}, U_r)e(g, g)^{\beta, r_{2,r}}.$$

In the Adapt part, we have: $g_{T,1} = K_{1,r} / \prod_{i,R_j} e(P_{i,r}, dk_i)^{\Delta i, R_j(0)}$, $g_{T,2} = K_{2,r} / e(\prod_{i,s} E_{i,r}, W_r)$.

Transform these two parts as follows:

$$K_{1,r} = g_{T,1} \cdot \prod_{i,R_j} e(P_{i,r}, dki)^{\Delta i,R_j(0)} = g_{T,1} \cdot \prod_{i,R_j} e(P_{i,r}, dki^{\Delta i,R_j(0)})$$
$$= g_{T,1} \cdot \prod_{i,R_j} e(D_{i,r}^{r_{1,r}}, dki^{\Delta i,R_j(0)}) = g_{T,1} \cdot \prod_{i,R_j} e(g^{d_{i,r}}, dki^{\Delta i,R_j(0)})$$
$$= g_{T,1} \cdot \prod_{i,R_j} e(g^{d_{i,r}}, g^{\frac{q_2(i)}{d_i}\Delta_{i,R_j}(0)}) = g_{T,1} \cdot e(g,g)^{\gamma, r_{1,r}}.$$

Similarly, we have $K_{2,r} = g_{T,2} \cdot e(g,g)^{\beta, r_{2,r}}$.

When assuming that the two instances of $K_{1,r}$ and $K_{2,r}$ are equal, we can deduce that $e(R_{1,r}, T_r) = g_{T,1}$ and $e(R_{2,r}, U_r) = g_{T,2}$. In this case, the modifier can obtain the desired decryption key z from V as follows: $z = V \oplus H(g_{T,1}) \oplus H(g_{T,2})$.

From the above, it is evident that when the calculations of $K_{1,r}$ and $K_{2,r}$ are correct, the modifier can successfully recover the key z, thereby reducing the correctness of this paper's scheme to the correctness of RBDS22. As RBDS22 is known to be correct, it follows that this paper's scheme is also correct.

6.2. Security Analysis

Theorem 1. *PFRB is secure against the chosen-plaintext attack.*

Proof. To prove the security of the encryption method against the chosen-plaintext attack, we need to demonstrate that the adversary cannot distinguish the generated ciphertext, even if it can choose a set of plaintexts and observe their encryption forms.

Considering the message need to be encrypted m, the challenger first generates a randomness r_t. Calculates the K_c by running $F_{en}(m, r_t)$. then let $z = H_{b,Z_p}(r_t, G_n(x))$, where $G_n(x)$ is an invariant polynomial which $\mathcal{A}$ cannot get. Computing $c_0 = k_c e(g,g)^z$. Return c_0 to the adversary $\mathcal{A}$.

Recall the above encryption process, we know that $e()$, g, F_{en}, $H_{b,Z_p}()$, λ are fixed. message m are values chosen by $\mathcal{A}$, and r_t are one-time random value chosen by $\mathcal{C}$. It is obvious that C_0 is a random value associated with r_t, which means although the adversary $\mathcal{A}$ can continuously submit requests to $\mathcal{C}$ and receive their corresponding ciphertexts, the ciphertexts appear random to $\mathcal{A}$. Therefore, $\mathcal{A}$ is incapable of distinguishing plaintexts from ciphertexts and it can only randomly guess b' from 0,1. Hence, we can know that

$$|\mathsf{Adv}_{\mathcal{A}}[b' = b] - \frac{1}{2}| \leq \epsilon,$$

where ϵ is negligible. Thus, PFRB is of CPA security, and Theorem 1 is proven. □

Also, PFRB can resist user collusion attacks if the soundness of Newton's interpolation formula-based secret sharing is hard. As shown in the detailed construction, PFRB relies on Newton's interpolation formula-based secret sharing to achieve controllable redaction. Specifically, Newton's interpolation formula-based secret sharing is used to specify policies and generate secret keys. namely, the user collusion resistance of PFRB can be reduced to the soundness of Newton's interpolation formula-based secret sharing. As proven in prior works, Newton's interpolation formula-based secret sharing holds soundness. Thus, PFRB is secure against the user collusion attack.

6.3. Application Discussion

The emergence of decentralized policy-hidden fine-grained redactable blockchain (PFRB) technology has opened new avenues for secure data management and privacy preservation. In this section, we explore the potential benefits of integrating PFRB with smart healthcare, smart industry, and artificial intelligence (AI) systems.

6.3.1. Application of PFRB in Smart Healthcare

Currently, with policy privacy, PFRB can be widely applied in smart healthcare applications to collect and manage medical data. For instance, PFRB ensures that nobody can infer users' private information from users' redaction policies. This benefit can impel users' enthusiasm for using portable devices (e.g., mobile phones and smart swatches) to collect data associated with their health condition, such as heart rate and temperature [29]. Clearly, these data hold high value for medical data analytic applications.

6.3.2. Application of PFRB in Smart Industry

In the industry environment, due to the harsh environment and heavy data collection tasks, IoT devices have been deployed to replace humans' work and securely transmit their data through blockchains [30]. In this case, policies from IoT devices usually contain large amounts of commercially sensitive data, such as data types and factory addresses [31]. Protecting this information is crucial to increase the wide implementation of blockchain-based IoT systems in the smart industry. Matching these practical requirements, PFRB is a promising solution to blockchain-based IoT systems in the smart industry.

6.3.3. Application of PFRB in AI

Recently, some cases have been proposed to prove the practical feasibility of combining the advantages of blockchains, AI, and IoT systems, such as swarm learning and machine/deep-learning-based IoT systems [32]. Specifically, IoT systems are responsible for collecting data for AI models, and blockchains are responsible for managing data and training AI models. The trained models can then provide rich services for IoT devices, such as fault detection and inference [33]. Similarly, in these promising applications, redaction policies from IoT devices contain much private information of IoT owners, such as policies for personal temperatures in medical analytics containing users' health conditions. PRFB addresses the policy leakage problem and can further impel the development of combining blockchains, IoT systems, and AI in real-world applications.

7. Performance Evaluation

In this section, we conduct experiments on the FISCO blockchain platform to evaluate the practical efficiency of PFRB.

7.1. Experimental Settings

Configuration: The experiment was conducted on a personal computer running Windows 10 (x64) with an Intel i7 8550U processor clocked at 1.80GHz and 8GB of memory. The implementation was done using the JPBC library in Java 8, and the MNT224 curve, known for its type-III properties and offering a 96-bit security level, was selected for pairing operations. Additionally, a 2048-bit RSA group was used for the Chameleon hash, providing a security level of 112 bits. Our scheme was deployed on the FISCO BCOS platform, which is an open-source platform customized for the financial industry. It is built upon the BCOS platform and incorporates module upgrades and functionality customization. The "Arbitration Chain" in FISCO leverages blockchain decentralization, tamper resistance, and trustworthiness. Real-time preserved data are securely stored on the blockchain using distributed data storage and encryption algorithms, ensuring the authenticity, legality, and relevance of the evidence.

Parameter design: The data owner randomly selects a news headline and uploads it to the service node as a transaction. In our experiment, we assume that each user can choose up to 100 attributes, such as gender, age, region, education level, occupation, etc., for access permissions. The access control policies for transactions are determined based on information such as the time and location of the news, and up to 100 policies can be set. It is important to note that the traditional scheme lacks policy protection and is vulnerable to security issues such as privacy leaks. This paper presents the experimental perfor-

mance of key generation, hashing, and adaptation and chooses recent related works [16,18] as comparisons.

Dataset: The implementation of our proposed solution utilizes the MNH9 dataset, which is derived from a real-world open dataset provided by the Australian Broadcasting Corporation. The MNH9 dataset consists of millions of news headlines.

7.2. Experimental Results

In this paper, we present a superior solution that excels in key generation, hashing, verification, and adaptation compared to traditional methods.

Figure 3 illustrates a significant improvement in the key generation aspect. For all three approaches, the required time increases linearly with the number of attributes, and both our proposed scheme and RBDS22 outperform the traditional approach by requiring considerably less time. This noteworthy enhancement is primarily attributed to the higher efficiency of our scheme, which leverages traditional linear secret sharing matrices at the underlying level, resulting in a more stable and efficient process.

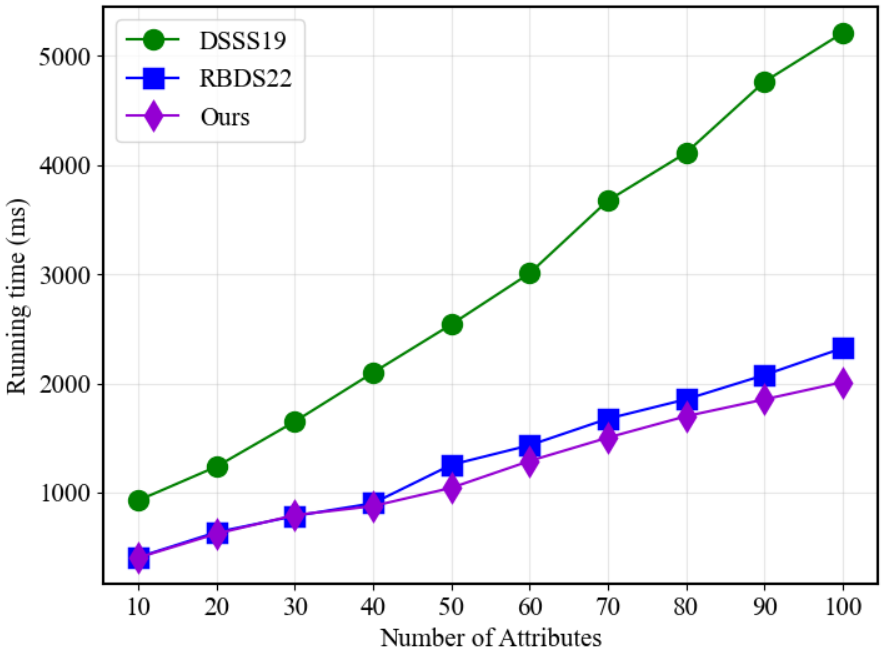

Figure 3. Key generation performance.

Moving on to Figure 4, the most noticeable efficiency improvement of the proposed solution is in the hashing part when compared to the traditional scheme. The traditional approach's required time increases exponentially with the number of policies, while the proposed solution and RBDS22 demonstrate linear growth. This advantage can be attributed to the use of polynomial functions generated from point values, which have proven to be significantly more efficient than using traditional linear secret sharing matrices schemes.

Figure 5 reveals that the adaptation algorithm part, which employs Newton's interpolation to reconstruct polynomials, presents challenges due to the enumeration of attribute-value pairs owned by the user. While the required time for both the traditional scheme and the proposed solution increases exponentially with the number of policies, the proposed scheme still outperforms the traditional approach significantly. Moreover, RBDS22 shows linear growth in the required time, and the proposed solution performs better than RBDS22, especially when the number of policies is low.

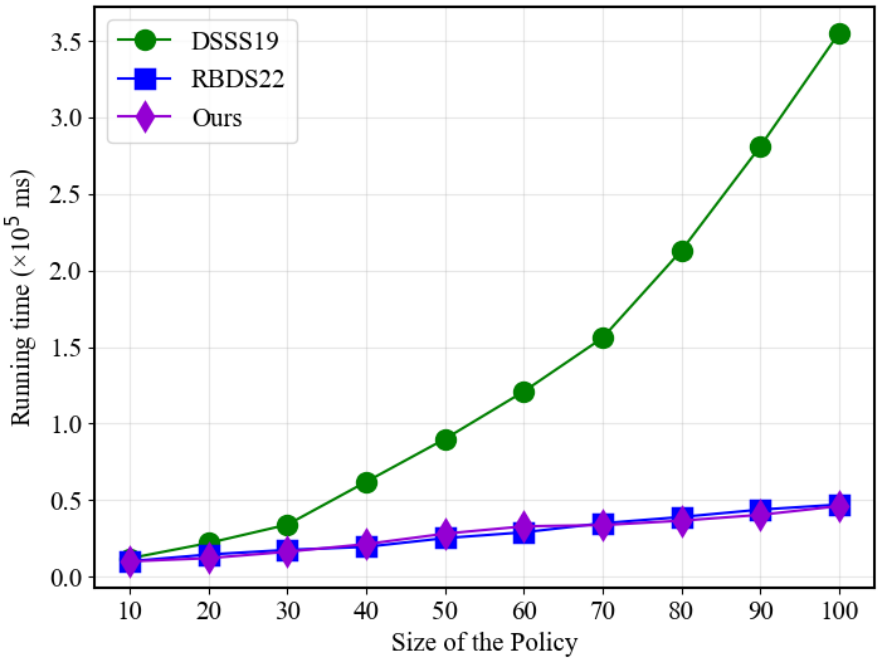

Figure 4. Hash performance with different policy sizes.

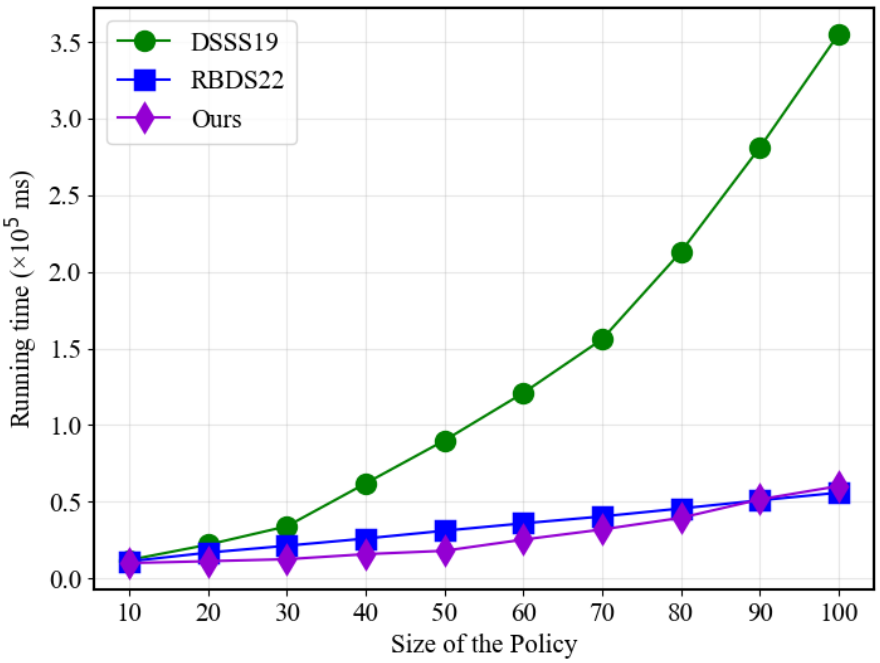

Figure 5. Adaption performance with different policy sizes.

Figure 6 demonstrates that in the Verify algorithm part, the response time of all three schemes increases linearly with the number of requests, with no significant difference in the time used. This similarity is mainly due to the similarity in the calculation operations performed in the Verify algorithm part of all three schemes. Therefore, the time used by all three schemes is similar in this regard.

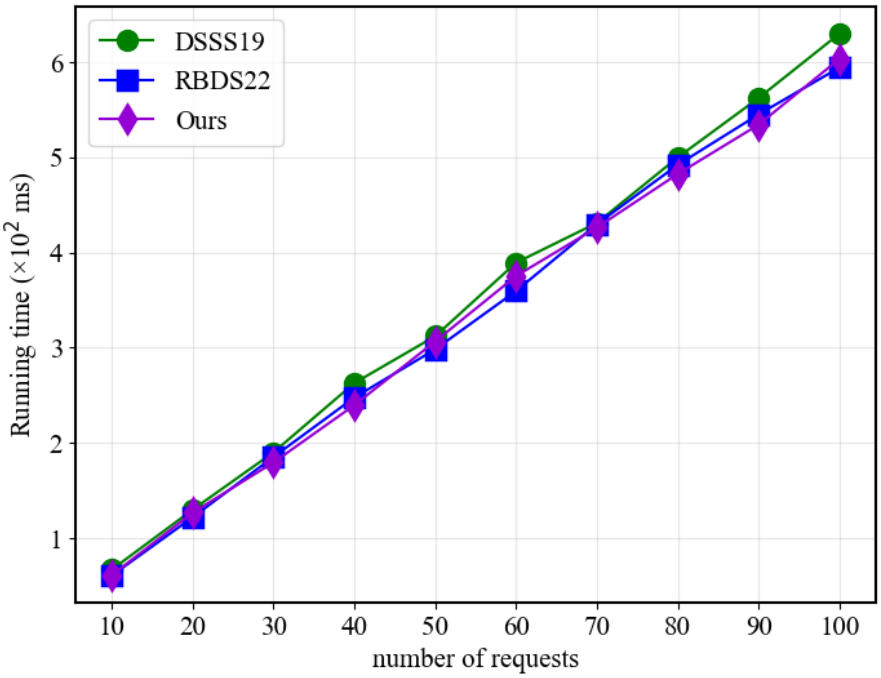

Figure 6. Verify performance with different request numbers.

Figures 7 and 8 display that when the number of policies is fixed, the response time of all three schemes in the hashing and adaptation algorithm parts increases linearly with the number of requests. However, the proposed scheme requires significantly less time than the traditional scheme and slightly less time than RBDS22 when the number of policies is relatively small. This advantage can be attributed to the proposed scheme's utilization of MA-ABE and Newton's interpolation, which enables it to achieve higher efficiency compared to traditional ABE and linear secret sharing matrices.

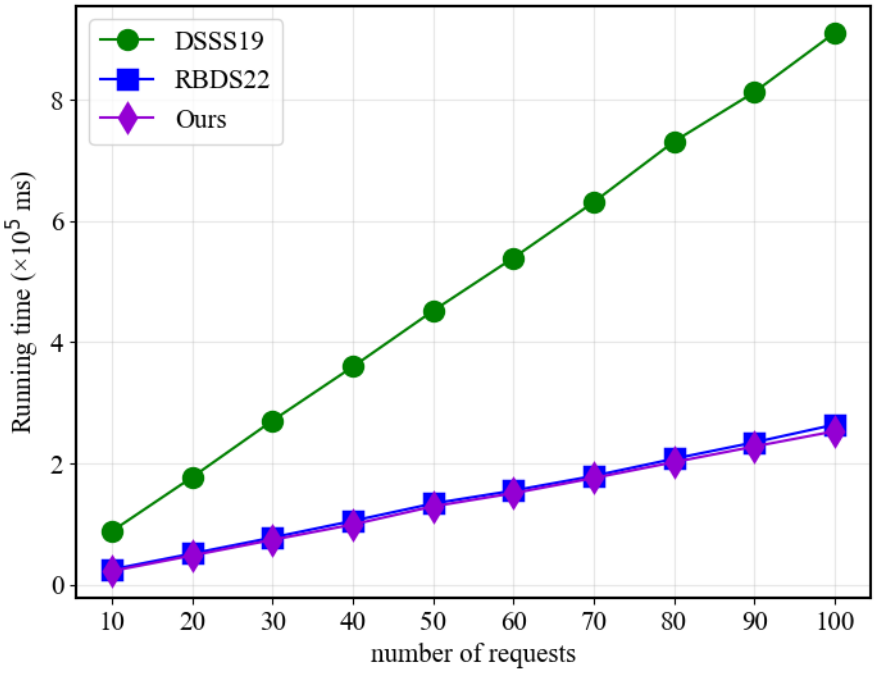

Figure 7. Hash performance with different user numbers.

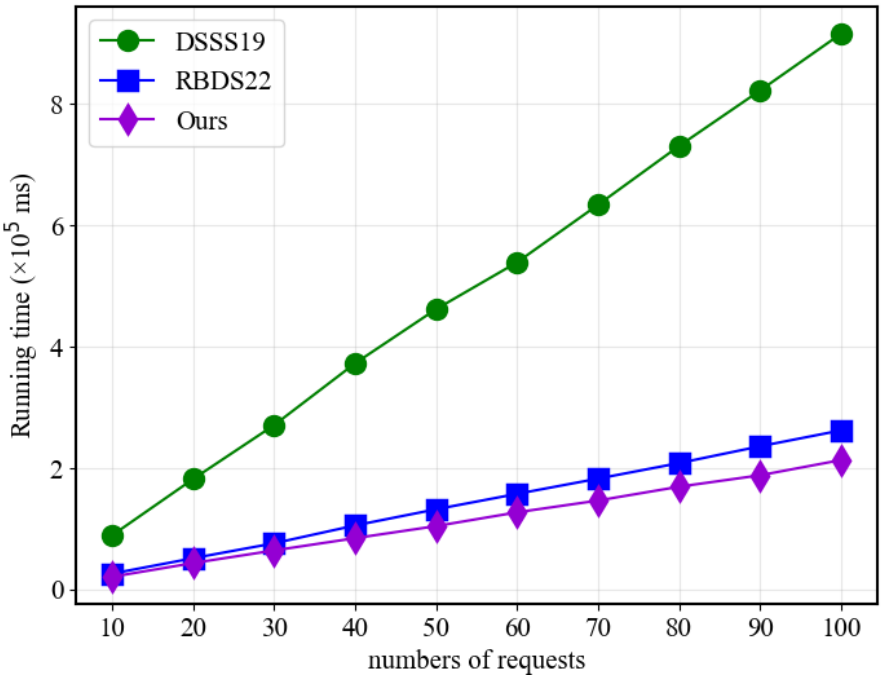

Figure 8. Adaption performance with different user numbers.

8. Conclusions

In this paper, we propose a policy-hidden fine-grained redactable blockchain (named PFRB) in decentralized blockchain-based IoT systems. Considering existing redactable blockchains, PFRB supports decentralized settings and fine-grained chameleon hash-based redaction. In addition, PFRB ensures that apart from successful policy matching, anyone cannot infer users' policy contents and execute any valid redaction. PFRB draws inspiration from Newton's interpolation formula-based secret sharing to convert policy contents into polynomial parameters. PFRB then utilizes multi-authority attribute-based encryption to hide these parameters further. Theoretical analysis proves that PRFB is secure against the chosen-plaintext attack. Extensive experiments on the FISCO blockchain platform and IoT devices show that PFRB achieves competitive efficiency over current redactable blockchains. For future work, we will focus on providing more comprehensive privacy protection (e.g., data privacy) and richer functionalities (such as accountability and revocation) in decentralized IoT systems. In addition, we will also focus on achieving richer data analytic mechanisms by combining various machine learning schemes.

Author Contributions: Conceptualization, H.G.; Formal analysis, M.Z.; Funding acquisition, C.Z. and J.X.; Methodology, H.G. and X.T.; Project administration, L.Z.; Software, X.T.; Supervision, T.W., C.Z., J.X. and L.Z.; Validation, T.W.; Writing—original draft, H.G., X.T. and M.Z.; Writing—review and editing, H.G., X.T., M.Z., T.W. and C.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by National Natural Science Foundation of China (Grant Nos. 62202051, 62102027, and 62172042), China Postdoctoral Science Foundation (Grant Nos. 2021M700435 and 2021TQ0042), Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies (Grant No. 2022B1212010005), Shandong Provincial Key Research and Development Program (Grant No. 2021CXGC010106), Major Scientific and Technological Innovation Projects of Shandong Province (2020CXGC010116), and Beijing Institute of Technology Research Fund Program for Young Scholars.

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

References

1. Chunka, C.; Banerjee, S.; Sachin Kumar, G. A secure communication using multifactor authentication and key agreement techniques in internet of medical things for COVID-19 patients. *Concurr. Comput. Pract. Exp.* **2023**, *35*, e7602. [CrossRef]
2. Ahmad, U.; Chaudhary, J.; Ahmad, M.; Naz, A.A. Survey on internet of things (IoT) for different industry environments. *Ann. Emerg. Technol. Comput. (AETiC)* **2019**, *3*, 28–43. [CrossRef]
3. Sinha, B.B.; Dhanalakshmi, R. Recent advancements and challenges of Internet of Things in smart agriculture: A survey. *Future Gener. Comput. Syst.* **2022**, *126*, 169–184. [CrossRef]
4. Liu, Y.; Hao, X.; Ren, W.; Xiong, R.; Zhu, T.; Choo, K.K.R.; Min, G. A Blockchain-Based Decentralized, Fair and Authenticated Information Sharing Scheme in Zero Trust Internet-of-Things. *IEEE Trans. Comput.* **2022**, *72*, 501–512. [CrossRef]
5. Kouicem, D.E.; Imine, Y.; Bouabdallah, A.; Lakhlef, H. Decentralized Blockchain-Based Trust Management Protocol for the Internet of Things. *IEEE Trans. Dependable Secur. Comput.* **2022**, *19*, 1292–1306. [CrossRef]
6. Mathur, S.; Kalla, A.; Gür, G.; Bohra, M.; Liyanage, M. A Survey on Role of Blockchain for IoT: Applications and Technical Aspects. *Comput. Netw.* **2023**, *227*, 109726. [CrossRef]
7. Hao, X.; Ren, W.; Fei, Y.; Zhu, T.; Choo, K.K.R. A blockchain-based cross-domain and autonomous access control scheme for internet of things. *IEEE Trans. Serv. Comput.* **2022**, *16*, 773–786. [CrossRef]
8. Bothra, P.; Karmakar, R.; Bhattacharya, S.; De, S. How can applications of blockchain and artificial intelligence improve performance of Internet of Things?–A survey. *Comput. Netw.* **2023**, *224*, 109634. [CrossRef]
9. Zhang, C.; Zhao, M.; Zhu, L.; Zhang, W.; Wu, T.; Ni, J. FRUIT: A blockchain-based efficient and privacy-preserving quality-aware incentive scheme. *IEEE J. Sel. Areas Commun.* **2022**, *40*, 3343–3357. [CrossRef]
10. Moonie, H. Man's "Right to Be Forgotten" Case Stalls After He Is Found on the Bitcoin Blockchain. 2016. Available online: https://medium.com/@hankmoonie/mans-right-to-beforgotten-case-stalls-after-he-is-found-on-the-bitcoin-blockchain-1a32c4fc0963 (accessed on 31 July 2023).
11. Tian, G.; Wei, J.; Kutylowski, M.; Susilo, W.; Huang, X.; Chen, X. VRBC: A Verifiable Redactable Blockchain With Efficient Query and Integrity Auditing. *IEEE Trans. Comput.* **2023**, *72*, 1928–1942. [CrossRef]
12. Shen, J.; Chen, X.; Liu, Z.; Susilo, W. Verifiable and Redactable Blockchains With Fully Editing Operations. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 3787–3802. [CrossRef]
13. Ateniese, G.; Magri, B.; Venturi, D.; Andrade, E. Redactable blockchain–or–rewriting history in bitcoin and friends. In Proceedings of the 2017 IEEE European Symposium on Security and Privacy (EuroS&P), Paris, France, 26–28 April 2017; pp. 111–126.
14. Deuber, D.; Magri, B.; Thyagarajan, S.A.K. Redactable blockchain in the permissionless setting. In Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP), Francisco, CA, USA, 19–23 May 2019; pp. 124–138.
15. Xu, S.; Ning, J.; Ma, J.; Huang, X.; Deng, R.H. K-time modifiable and epoch-based redactable blockchain. *IEEE Trans. Inf. Forensics Secur.* **2021**, *16*, 4507–4520. [CrossRef]
16. Derler, D.; Samelin, K.; Slamanig, D.; Striecks, C. Fine-Grained and Controlled Rewriting in Blockchains: Chameleon-Hashing Gone Attribute-Based. *Cryptol. ePrint Arch.* **2019**.
17. Jia, M.; Chen, J.; He, K.; Du, R.; Zheng, L.; Lai, M.; Wang, D.; Liu, F. Redactable Blockchain From Decentralized Chameleon Hash Functions. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 2771–2783. [CrossRef]
18. Ma, J.; Xu, S.; Ning, J.; Huang, X.; Deng, R.H. Redactable blockchain in decentralized setting. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 1227–1242. [CrossRef]
19. Xu, S.; Huang, X.; Yuan, J.; Li, Y.; Deng, R.H. Accountable and Fine-Grained Controllable Rewriting in Blockchains. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 101–116. [CrossRef]
20. Zhang, C.; Zhao, M.; Zhu, L.; Wu, T.; Liu, X. Enabling Efficient and Strong Privacy-Preserving Truth Discovery in Mobile Crowdsensing. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 3569–3581. [CrossRef]
21. Hu, C.; Zhang, C.; Lei, D.; Wu, T.; Liu, X.; Zhu, L. Achieving Privacy-Preserving and Verifiable Support Vector Machine Training in the Cloud. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 3476–3491. [CrossRef]
22. Zhang, C.; Zhao, M.; Xu, Y.; Wu, T.; Li, Y.; Zhu, L.; Wang, H. Achieving fuzzy matching data sharing for secure cloud-edge communication. *China Commun.* **2022**, *19*, 257–276. [CrossRef]
23. Zhang, C.; Hu, C.; Wu, T.; Zhu, L.; Liu, X. Achieving Efficient and Privacy-Preserving Neural Network Training and Prediction in Cloud Environments. In *IEEE Transactions on Dependable and Secure Computing*; Early Access; IEEE: New York City, NY, USA, 2022.
24. Tian, Y.; Li, N.; Li, Y.; Szalachowski, P.; Zhou, J. Policy-based chameleon hash for blockchain rewriting with black-box accountability. In Proceedings of the Annual Computer Security Applications Conference, Austin, TX, USA, 7–11 December 2020; pp. 813–828.
25. Panwar, G.; Vishwanathan, R.; Misra, S. ReTRACe: Revocable and traceable blockchain rewrites using attribute-based cryptosystems. In Proceedings of the 26th ACM Symposium on Access Control Models and Technologies, Virtual, 16–18 June 2021; pp. 103–114.
26. Jia, Y.; Sun, S.F.; Zhang, Y.; Liu, Z.; Gu, D. Redactable blockchain supporting supervision and self-management. In Proceedings of the Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security, Virtual, 7–11 June 2021; pp. 844–858.

27. Xu, S.; Ning, J.; Ma, J.; Xu, G.; Yuan, J.; Deng, R.H. Revocable policy-based chameleon hash. In Proceedings of the Computer Security–ESORICS 2021: 26th European Symposium on Research in Computer Security, Darmstadt, Germany, 4–8 October 2021; pp. 327–347.
28. Chase, M. Multi-authority attribute based encryption. In Proceedings of the Theory of Cryptography: 4th Theory of Cryptography Conference, TCC 2007, Amsterdam, The Netherlands, 21–24 February 2007; pp. 515–534.
29. Chae, Y.; Wang, S.; Kim, S.M. Exploiting WiFi Guard Band for Safeguarded ZigBee. In Proceedings of the Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems, SenSys, Shenzhen, China, 4–7 November 2018; pp. 172–184.
30. Wang, S.; Kim, S.M.; He, T. Symbol-Level Cross-Technology Communication via Payload Encoding. In Proceedings of the 38th IEEE International Conference on Distributed Computing Systems, Vienna, Austria, 2–5 July 2018; pp. 500–510.
31. Wu, C.; Li, X.; Zuo, F.; Luo, L.; Du, X.; Di, J.; Zeng, Q. Use It-No Need to Shake It!: Accurate Implicit Authentication for Everyday Objects with Smart Sensing. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **2022**, *6*, 146:1–146:25. [CrossRef]
32. Zhang, J.; Li, Y.; Xiao, W. Integrated Multiple Kernel Learning for Device-Free Localization in Cluttered Environments Using Spatiotemporal Information. *IEEE Internet Things J.* **2021**, *8*, 4749–4761. [CrossRef]
33. Zhang, J.; Li, Y.; Xiao, W.; Zhang, Z. Online Spatiotemporal Modeling for Robust and Lightweight Device-Free Localization in Nonstationary Environments. *IEEE Trans. Ind. Inform.* **2023**, *19*, 8528–8538. [CrossRef]