Article

An Anonymous Protocol with User Identification and Linking Capabilities for User Privacy in a Permissioned Blockchain

Gyeongjin Ra 1, Deahee Seo 2, Md Zakirul Alam Bhuiyan 3 and Imyeong Lee 1,*

1 Department of Computer Science and Engineering, Soonchunhyang University, Asan-si 31538, Korea; rababi@sch.ac.kr
2 Department of Computer Science, Kennesaw State University, Marietta, GA 30060, USA; dseo1@kennesaw.edu
3 Department of Computer Science and Information Sciences, Fordham University, Bronx, NY 10458, USA; m.bhuiyan.dr@ieee.org

* Correspondence: imylee@sch.ac.kr

Received: 27 May 2020; Accepted: 6 July 2020; Published: 22 July 2020

Abstract: A permissioned blockchain includes a user in the network after verifying the user’s identity, in contrast to Bitcoin, which is a public blockchain that allows network participation without third-party approval. The two types of permissioned blockchains are private blockchains, each consisting of one server and multiple users, and consortium blockchains, which consist of groups of private blockchains. However, a blockchain has privacy issues, such as user tracking and inference. Therefore, cryptography should be applied for user privacy in a blockchain. There is a lot of research on anonymous protocols for privacy in a blockchain. In this paper, we provide a scheme for user management, i.e., identification and authorization, in a permissioned blockchain. We also propose an anonymous protocol with user identification and transaction linking capabilities provided by the private server, strictly to solve privacy concerns.

Keywords: permissioned blockchain; privacy; pseudonym; anonymous credential system

1. Introduction

A blockchain is a distributed ledger in which all participants can record and share information with each other [1]. In addition, a blockchain is a structure that enables modification of the data by the owner of a private key. As a result, the owner of the private key can control the data. A blockchain is also capable of providing a public key to access and offer signature verification of data, without a central trusted third party (TTP) public key certificate. It can provide anonymity, as Bitcoin does, because it uses a pseudonym (public key) without revealing the actual identity information. Blockchains are increasing in popularity in that they provide decentralization, integrity, transparency, and privacy [2].

Unlike permissionless environments, e.g., Bitcoin, wherein the public network allows network participation without third-party approval, a permissioned blockchain is composed of a set of known, identified nodes that might not completely trust each other [3]. Additionally, permissioned models control who participates in the validation protocol; these nodes usually have constructed identities and form a consortium [4]. For example, consider an independent hospital that records a patient’s prescription information in a private blockchain [5]. To negotiate with other hospitals, which can share the records and prescriptions of patients, a consortium blockchain is preferred. Similarly, a consortium blockchain can allow customers to make banking transactions both within the bank and between multiple banks. Service providers in the permissioned blockchain need to identify and manage a user’s identity. In a permissioned blockchain, this happens through a membership service provider.
MSP). Repeated use of the same public key by a user may lead to a privacy problem due to inferring or tracking the specific user with the identifier recorded in the blockchain. If a member’s transaction in the private blockchain can be identified, the member can also be known on the consortium blockchain. Sensitive information, such as hospital records, must have greater security to protect the privacy of the user [6]. It is essential that permissioned blockchains guarantee privacy between consortium members while managing user authentication.

Privacy means the ability of an individual or group to conceal and selectively reveal their own information. Anonymity is derived from Greek, meaning “without a name” or “namelessness” [7,8]. Anonymity, as part of privacy, means that even if a message (transaction) is disclosed on the blockchain, the public key is not related to the individual’s off-chain ID (real identity) [8]. Even if one possesses the message, still, the identity of the individual cannot be identified or inferred [7,8]. In this connection, unlinkability and untraceability are essential. Unlinkability means one has “no idea how transactions are related,” and the untraceability means one has “no idea where a transaction came from.” Bitcoin, which uses financial transactions with the mask of public keys, provides unlinkability. However, the untraceability of the same user in the blockchain transaction is not guaranteed. Monero offers both unlinkability and untraceability through a ring signature and a one-time key [8]. On the other hand, there are also areas where blockchain is applied rather than currency. Yin et al. and Yang et al. used blockchain to improve privacy, anonymity, and efficiency in 5G networks and in the cloud [9,10]. Reference [9] secured reliability through the layer of the blockchain and strengthened anonymity using only the public key address. Reference [10] applied blockchain for efficiency and reliability in a 5G network. In addition, anonymous authentication is performed using a zero-knowledge certificate. However, for zero knowledge verification, the key generation and verification verification time are required separately. We try not to use zero-knowledge proofs. Additionally, complete anonymity, such as zero-knowledge proof, may not be able to control malicious users in the business model.

In a permissioned blockchain, the goals of unlinkability and untraceability need not always be fulfilled simultaneously. A permissioned blockchain needs the MSP to identify, manage, and restrict participants. Depending on policy, a permissioned blockchain may also require the tracking of malicious members or verification of the identity of a user upon a legitimate request. Only partial anonymity is needed to protect the user’s privacy while revealing the user’s identity and efficiently computing connections at the same time. In this paper, we propose an anonymous protocol that features improved user privacy in permissioned blockchains and fulfills the requirements of anonymity, linking, and user identification.

This paper consists of five sections: introduction, related work, preliminary remarks, proposed scheme, and analysis and conclusion. In Section 1, we introduce the blockchain and privacy problem and some design issues for the permissioned blockchain. In Section 2, we summarize the related works. In Section 3, we review the background and preliminary work for the proposed scheme. In Section 4, we cover the proposed scheme in depth—the motivation, contributions, overview, entities, and security requirements. We also describe the use of the workflow and the formula. In Section 5, we analyze and compare our scheme to other existing schemes. In Section 6, we set out our conclusion.

2. Related Work

In this section we review existing methods for improving privacy in blockchains. Privacy is a critical issue for blockchains. Since a public blockchain may be accessed by anyone, malicious users may be able to obtain personal data. A permissioned blockchain has a duty to users to protect their privacy. We look at the privacy considerations in [7]. We compare a variety of privacy schemes; the results are summarized in Tables 1 and 2 below. The most often considered choices for fulfilling privacy on the blockchain are as follows [7]:

1. Homomorphic encryption (HE): All transactions are encrypted in such a way that they can be easily investigated without revealing the real values [7]. HE is considered an inefficient method with numerous factors needing large computing resources and keys up to 25 GB in size [11].
Confidential transactions (CT), used to hide transaction values, are performed using additionally homomorphic encryption, with a lower calculation overhead [12].

(2) Mixing services: Transactions are mixed by third-party servers. One company, JoinMarket, used a method where an order mix request was broadcast via IRC (Internet Relay Chat) [13]. In July 2015, it was revealed that JoinMarket failed in its execution method and so allowed curious parties to perform analysis to link output and input address pairs [14]. Afterwards, Ruffing et al. proposed a coin-mixing scheme using unlinkable Bitcoin. In this scheme, the server returns a transaction that mixes users’ addresses, which the users then sign. This solved the server privacy problem and kept the advantages of the central mixing server [15].

(3) Application of privacy by design: Cryptography can be applied to some applications that need characteristics such as anonymity and untraceability—currency, for example. A cryptocurrency could also generate all transactions with "zero-knowledge"; for example, ZCash [16]. As noted previously, Monero offers unlinkability and untraceability through a ring signature and a one-time key [17].

| Scheme                  | Environments          | Main Technologies                                      | Features                                                                                      |
|-------------------------|-----------------------|--------------------------------------------------------|-----------------------------------------------------------------------------------------------|
| Mercer et al. [7]       | Public Blockchain    | Ring Signature, Tag, Traitor Tracing                   | All transactions are defaulted to anonymous transactions; Validates the same signer through a tag, Traitor tracing; Uses zero-knowledge proof to verify the tags |
| Ruffing et al. [15]     | Public Blockchain    | Mixing Server, Ring Signature, UDP, Anonymous Addresses | Implements anonymous ring public key; Uses UDP to aggregate addresses and return a mixed transaction |
| Hardjono et al. [18]    | Bitcoin, Public Blockchain | ZKP, Anonymous Addresses, Credential System            | Anonymous user identities; Performance relies on issuer and verifier; Issuer issues the user an anonymous address token to use with Bitcoin; Uses zero-knowledge proof to verify the token |
| Zhang et al. [19]       | Bitcoin, Permissioned Blockchain | Public Key Encryption Keyword Search, URL Linking | Data security; Access control; Privacy preservation and secure search |
| Jiang et al. [20]       | Public Blockchain    | Oblivious Keyword Search with Authorization            | Enables oblivious search over an authorized keyword set in the decentralized storage; Constant size communication cost in the transfer |
| Dubovitskaya et al. [21]| Permissioned Blockchain | MPSI, Gabled Bloom Filter                               | Uses an outsourcing provider who has no knowledge of the data inputs or outputs; Reduced computational complexity |
| Zhu et al. [22]         | Permissioned Blockchain | Blind Signature, Bloom Filter                         | Offers users privacy protection and is easy to implement |
| Aitzhan et al. [23]     | Public Blockchain    | Multi-Signature, Anonymous Stream-Messaging System,    | Anonymous user energy trading messages; Uses multi-signature to cooperate with middleman and energy trading party members to generate transactions |
| Goldfelder et al. [24]  | Public Blockchain    | ZKP, Threshold Signature                               | Prevents arbitrary signing and creation of transactions; Requires more than some threshold number for signature key recovery |
| Axon et al. [25]        | Public Blockchain    | Off-Chain Storage, Key Update                          | Uses off-chain storage with separate blockchain records; 'Public key updating'; Un-linkability of the user’s 'public key (or address) |
| Heilman et al. [26]     | Public Blockchain    | Blind Signature, Smart Contract, Voucher System        | Anonymous user messages; Uses an untrusted third party to issue anonymous vouchers |

(4) Access structures: Access structures can improve privacy in a blockchain by hiding metadata or having transactions that can only be read by certain parties, as determined by access control systems [27]. An anonymous system can hide members’ identities with tickets issued through an issuer [28]. Hardjono et al. proposed anonymous identity in Bitcoin by verifying issued tokens via the issuer and zero-knowledge proofs [18].

(5) Traitor tracing: Traitor tracing can be used to punish parties who act to reduce the anonymity of others, increasing the anonymity of the scheme as parties are less likely to work to one another’s detriment [8]. Bitcoin provides this stochastically [29].
(6) Broadcast encryption: Broadcast encryption provides strong anonymity for receivers, as all users in the group receive the encrypted message and only special users with the issued permission or key can decrypt the message. Additionally, broadcast encryption is fully collusion resistant [30]. Zhang et al. propagated encrypted e-health records to a private blockchain [19]. Only authorized patients and doctors can see them. Jiang et al. also proposed a scheme where the data service provider encrypts and broadcasts data to the blockchain, after which the user can access the target data by keyword search [20].

(7) Secure hardware: Secure hardware limits what the user can do. For example, the hardware could implement mixing before generating transactions or could add randomness in blind signatures [31]. For example, users can use a “Bloom filter” to check whether a message belongs to a set without revealing the message. In Bitcoin, Simple Verification Payment (SVP) validates transactions through a Bloom filter. Intermediaries can remove the property of traceability when depositing and withdrawing money from an exchange. Dubovitskaya et al. proposed a way to obtain data that would strengthen privacy using multiparty private set intersection (MPSI) and garbled Bloom filters [21]. Zhu et al. proposed hiding the relationships on social networks using the server as an intermediary with blind signatures and a Bloom filter [22]. Aitzhan et al. enhanced privacy in an energy trading system through multi-signature and anonymous message streams [23].

Table 2. A scheme classification on options for privacy in blockchain.

| Access Structures | Broadcast Encryption | Key Update | Off Chain Storage | SMPC | Secure Hardware | Anonymity Revocation |
|-------------------|----------------------|------------|-------------------|------|----------------|---------------------|
| User Identification | Linking |
| Mercer, R. [7]     | Offered              | Not offered| One time key      | Not offered | ZKP            | Ring signature      | Not offered  |
|                      |                      |            |                   |      |                | Private issuer      |            |
| Raffing et al. [15] | Offered              | Not offered| One time key      | Not considered | Mixing + Ring signature | Not offered  |
|                      |                      |            |                   |      |                | Private issuer      |            |
| Hardjono et al. [18] | Offered             | Not offered| One time key      | Not considered | ZKP            | Private issuer      | Not offered  |
|                      |                      |            |                   |      |                | Private issuer      |            |
| Zhang et al. [19]   | Offered              | Not offered| URL Linking based- | Not considered | MPSI + Garbled Bloom filter | Private issuer |
|                      |                      |            | Consortium Blockchain |      |                |                    |            |
| Jiang et al. [20]   | Offered              | Not offered| Not offered       | Not offered | OKSA           | Ordered Multi signature | Not offered  |
|                      |                      |            |                   |      |                | Private issuer      |            |
| Dubovitskaya et al. [21] | Offered         | Not offered| Not offered       | Not offered | MSP + Garbled Bloom filter | Not offered  |
|                      |                      |            |                   |      |                | Private issuer      |            |
| Zhu et al. [22]     | Offered              | Not offered| Not offered       | Not considered | MPSI + Garbled Bloom filter | Not offered  |
|                      |                      |            |                   |      |                | Private issuer      |            |
| Aitzhan et al. [23] | Not offered          | Not offered| Not offered       | Not considered | MPSI + Garbled Bloom filter | Not offered  |
|                      |                      |            |                   |      |                | Private issuer      |            |
| Goldfeder et al. [24] | Offered        | Not offered| Not considered | Not offered | MPSI + Garbled Bloom filter | Not offered  |
|                      |                      |            |                   |      |                | Private issuer      |            |
| Axon et al. [25]    | Not offered          | Not offered| One time key      | Not considered | MPSI + Garbled Bloom filter | Not offered  |
|                      |                      |            |                   |      |                | Private issuer      |            |
| Heilman et al. [26] | Offered              | Not offered| Off chain         | Not considered | MPSI + Garbled Bloom filter | Not offered  |
|                      |                      |            |                   |      |                | Private issuer      |            |

(8) Secure multi-party computation (SMPC): SMPC allows parties to work together in such a way that none of them can access any data, and no one is able to leak any secret information [32]. In a blockchain, zero-knowledge proof (ZKP) and zero-knowledge succinct non-interactive arguments of knowledge (Zk-SNARKs) can be used for secure computation. The user uses a secure computation to prove that he has confidential information, without exposing the confidential information. Goldfeder et al. solved the problems of signatures generated without authentication and privacy issues using threshold signatures and zero-knowledge in Bitcoin [24]. Zk-SNARKs provide cryptocurrency users with the ability to hide transaction data using non-interactive zero-knowledge proofs [33]. ZCash is a cryptocurrency that attempts to address the lack of privacy in blockchain transactions using Zk-SNARKs. ZCash uses Zk-SNARKs on a Merkle tree, which contains all previous transactions. It is updated every time a transaction occurs, and if that happens at an average rate of 1000 transactions per second, the Merkle tree will take about 29.2 million years to populate. This is large and inefficient [7]. Jiang et al. proposed a keyword search without leaking the user’s keyword information using oblivious transfer keyword search (OTKS) [20]. Mercer proposed a unique ring signature using zero-knowledge proof-based tags and User Datagram Protocol (UDP). This scheme provides traceability via tags [7].
(9) Off-chain storage: Off-chain storage enhances the privacy of a blockchain by storing sensitive data off-chain and simply accessing it when needed [25]. Axon et al. improved user privacy by storing data off-chain and continually updating keys [25]. Heilman et al. proposed an off-chain system using blind signatures and smart contracts with vouchers for anonymity in Bitcoin [26].

(10) Anonymous digital signature: Anonymous digital signature provides untraceability by performing signatures that include the signatures of others. The initial blind signature proposed by David Chaum differs between the signer and the message author [33]. Since then, anonymous digital signatures have provided anonymity by hiding the user’s single signature in multiple signatures of the group. The following describes the various methods of anonymous digital signature. Additionally, as shown in Tables 3 and 4, the existing anonymous digital signatures are not efficient and lack several other features, such as identity verification and signature authentication.

(11) Blind Signature: Blind signatures can effectively protect the confidentiality of user messages for user privacy and anonymity [34]. With a blind signature, the user is separate from the signer. Users use interactive signing protocols and work with signers to generate signatures. The signer generates a blind signature of message m for the user in a blind state (i.e., the signer does not know the content of message m), and the user “unblinds” the message signed with the blind signature to obtain a signed message m. Blind signatures use a TTP signer, and the user must fully trust the signer. Additionally, with an interactive signature, the user must obtain a new blind signature each time a signature is required.

(12) Group Signature: A method that allows one of the group members to sign a message anonymously. We know that a member of the group has signed it but do not know exactly who signed it. The essential element of the group signature is the group manager [35]. The group manager can add new members to the group and has the right to reveal the signers in case of a dispute. The group manager has a lot of authority.

(13) Ring Signature: The signer can be anyone from the group who has a key, so a message signed using a ring signature is guaranteed by a specific member in the group [8]. The ring signature is characterized by the difficulty of knowing which member’s key in the group has been used to sign. The term ring signature comes from the fact that the structure of the signature algorithm is like a ring. Thus, all group members are in the same position, and there is no central member. Ring signatures use the public key of only one member in the group, so there is no need to generate a new signature key when adding or deleting group members. However, the size of the public keys increases in proportion to the number of group members.

(14) Threshold Signature: A signature that divides the signature key into several secret pieces, and when more than a certain number of secret pieces (threshold) is acquired, the original signature key can be recovered [24].

|                  | Group Signature | Ring Signature | Threshold Signature | Blind Signature |
|------------------|-----------------|----------------|--------------------|-----------------|
| TTP              | Group manager   | Not required   | Key distribution   | Signer          |
| Anonymity        | Offered         | Offered        | Offered            | Offered         |
| Revocation Anonymity | Group manager   | Only signer oneself | Group manager     | Only itself     |
| Signing Threshold| Not considered  | Not considered  | k of n             | Not considered  |
| Setup Phase      | Group key generation | Public(Validity) key Generation | Secret pieces sharing | Not required  |
Table 4. A comparison of the efficiencies of anonymous digital signatures.

| Signature Type       | Signature Size (bit) | Key Gen Time | Proof Time | Verify Time | Related Scheme |
|----------------------|----------------------|--------------|------------|-------------|----------------|
| Group Signature      | O (1)                | O (n)        | -          | Less than 1 (s) | -              |
| Ring Signature       | O (n)                | O (n)        | -          | Less than 1 (s) | [7,8,17]       |
| SNARKS               | -                    | 1200 (s)     | 1320 (s)   | 4.68 (s)    | [16,33]        |
| Blind Signature      | O (1)                | O (1)        | -          | Less than 1 (s) | [31,34]       |
| Multi Signature      | O (n)                | O (1)        | -          | Less than 1 (s) | [23]           |
| Threshold Signature  | O (1)                | O (1)        | -          | Less than 1 (s) | [24]           |

3. Preliminary

In this section, we discuss the background and preliminaries required in this paper.

3.1. Anonymous Credential System

A credential system is a system in which a user can obtain credentials from institutions and can announce those credentials. It is called an anonymous credential system when others do not know if the actions are all done by the same user [36]. In Figure 1, the sender (user) registers identity and key with issuer (institution). The sender authenticates to issuer via a key and requests an anonymous credential (pseudo ID). The sender and the receiver perform a secret calculation for mutual authentication. At this time, the sender uses the pseudo ID obtained from issuer. Therefore, the receiver does not know the real identity of the sender. This is an anonymous credential system associated with privacy and anonymity. This guarantees that a credential issued by an institution has accurate information and users can be validated on demand. Anonymous credential systems are sometimes referred to as pseudonym systems because users are issued a credential with a different pseudonym by each institution. A pseudonym allows users to use a different name at each institution. Each institution can associate users with user accounts through a pseudonym, but institutions cannot know the true identities of their users. However, an anonymous credential can assure a relationship between the user and the institution, and it can prove the relationship to other institutions [37]. In relation to our work, a private blockchain uses an anonymous credential system that manages users’ identity information and issues pseudonym IDs to the users. Users can generate messages anonymously via the pseudonym ID. The private blockchain can provide the user’s identity information upon demand.

![Diagram of the anonymous credential system.](image)

3.2. Bilinear Pairing

Bilinear pairing assumes that it is difficult to solve the discrete logarithm problem when multiplicative groups $G_1$ and $G_2$ of a larger multiplicative group $n$ have the same constant $p$. $g_1$ is the constructor of $G_1$ and $g_2$ is the constructor of $G_2$. $\psi$ is a homogeneous calculation from $G_1$, $G_2$ and $\psi(g_2) = g_1$. If $e : G_1 \times G_2 \to G_T$ in $L = (n, G_1, G_2, G_T, e, g_1, g_2)$ satisfies the following conditions, $e : G_1 \times G_2 \to G_T$ is termed a bilinear pairing [38].
Bilinear: \( \forall u \in G_1, v \in G_2 \) and \( a, b \in \mathbb{Z}_n \), \( e(u^a, v^b) = e(u, v)^{ab} \).

Nondegenerate: For \( u \in G_1, v \in G_2, e(u, v) \neq \emptyset \), where \( \emptyset \) is the zero factors of \( G \).

Computability: There exists an efficient algorithm allowing the computation \( e(u, v) \) for \( \forall u \in G_1, v \in G_2 \).

Complexity assumption: Let \( P \) be the generator for the groups \( G_1 \) and \( G_2 \) is defined above. The \( q \)-BDHI problem is defined as follows [39]: Given the \( q+1 \)-tuple \( (P, x_P, x^2P, \ldots, x^q) \in G_1^{q+1} \) as input, compute \((P, P)^{1/x} \in G_2^{+}\). An algorithm \( A \) has an advantage \( \epsilon \) in solving \( q \)-BDHI \( G_1 \) if

\[
\Pr[A(P, x_P, x^2P, \ldots, x^q) = e(P, P)^{1/x}] \geq \epsilon
\]

where the probability is over the random choice of \( x \in \mathbb{Z}_p^* \) and random bits of \( A \).

**Definition 1.** \( q \)-BDHI assumption. The \( q \)-BDHI assumption in \( G_1 \) and \( G_2 \) holds if all polynomial-time algorithms have negligible time to solve the \( q \)-BDHI problem.

### 3.3. Blockchain

Blockchain is a distributed ledger technology; all nodes both record and store the same ledger. A block usually consists of the hash value of the previous block, the signature of the contributor, a payload and timestamp. The hash value of the previous block makes the blockchain immutable (Figure 2). Therefore, unlike existing databases, the ledger can only add transactions; data cannot be modified or deleted. Therefore, it is exceedingly difficult to counterfeit or tamper with data. First, data entry is irreversible given the hash calculation within the block. Second, the contents of the previous block form a chain with the contents of the next block, and no node can change. Additionally, as all nodes store records identically, a network can be created and maintained in the absence of central organization. This avoids the possible single-point-of-failure problem of centralized structures and minimizes broker activities and commissions. Blockchains are divided into public blockchains that anyone can join, and permissioned blockchains that accept only authorized members. A permissioned blockchain contains a member who identifies authorized users or groups [40].

![Figure 2. Structure of blockchain.](image-url)
pseudonym IDs, which helps to improve the users’ privacy. Each private blockchain issues pseudonym IDs, which are linked to users’ real identity information. The private blockchain store the users’ data, including the pseudonym ID. A private blockchain negotiates to form a consortium blockchain, which keeps records of the anonymous data via an aggregated signature.

3.4. Elliptic Curve Cryptography

Compared to conventional cryptographic techniques, such as the Digital Signature Standard (DSS), Rivest–Shamir–Adleman (RSA), and Diffie–Helman (DH) key exchange, elliptic curve cryptography (ECC) is considered more efficient for security [41]. ECC is a public key cryptosystem based on elliptic curve theory. An elliptic curve is a curve defined by an equation of the form $y^2 = x^3 + ax + b$, yielding points on an ellipse, which contains a special category of real numbers, used to develop public keys. Early public-key cryptography was based on the fact that it takes a very long time to divide a very large integer into two or more prime numbers [41].

ECC exploits the fact that it also takes an exceptionally long time to determine the discrete logarithm of a random elliptic curve at any given point. Discrete logarithms require that $x$ satisfies the equation $a^x = b$. If $a$ and $x$ are known, it is very easy to find $b$, but if only $a$ and $b$ are known is the difficulty of finding $x$ known as the discrete logarithm problem, which is exploited to render it very difficult to calculate the private key from the public key. In detail, the discrete logarithm problem can be stated as follows: it is difficult to find the logarithm of a surplus system that considers only the remainder of a modular operation. If the previously given statement is satisfied, given $Q$ as a point and $t \in F_p^*$ as an integer, then repeated addition is used to define scalar multiplication, i.e., $tQ = Q + Q + Q + \ldots + Q$ ($t$ times). The domain parameters are members of finite field $F_p$, i.e., $(p, a, b, P, n, h) \in F_p$.

$E$ is an abelian group and the identity element of this group is the point that lies at infinity [42].

3.5. Aggregate Signature and Root Signature

An aggregate signature scheme was proposed by Yuan et al. in 2017 [43]. They proposed a new scheme that was designed to hide the number of transactions in a blockchain, which contains multiple inputs and outputs. It uses an aggregate signature to combine transactions and produce a single group signature.

A root signature is a type of aggregate signature. The user creates a one-time public/private key pair $(PK, SK)$ using ECC. $P$ is the public key set and $PK_i \subseteq P$ refers to a one-time public key (Figures 3 and 4). In our work, each one-time public key $PK_i \in P$ is associated with a one-time private key $SK_i$. A user $u \in U$ creates a root signature for public key verification generated as an aggregate signature using the one-time public keys $PK_i$ for each transaction as input. Others can access, but are not allowed to modify the user’s one-time public keys or the root signature without the private key. The root signature is the same size as the one-time public key. The single signature value of the aggregate one-time public key that comes from this is called the root public key.

![Figure 3. One-time keys and root signature.](image-url)
4. Proposed Scheme

In this section, we explain the proposed scheme. This section consists of six subsections: our contribution, security requirements, overview, entities in the system, workflow, and protocol.

4.1. Our Contribution

We provide two contributions. One is the privacy issue and the other is user identification and linking capabilities.

1) Privacy issue: There can be privacy problems when sharing a ledger among groups. In the consortium blockchain, a member authenticated in multiple private blockchain with the same identifier can cause privacy concerns, such as inference of members. In this paper, we aimed to guarantee privacy through anonymity via pseudonym IDs, which are issued by an anonymous credential system and by using aggregated signatures.

2) User identification and linking capabilities: With complete anonymity it is not possible to track malicious members in a permissioned blockchain. Therefore, traceability and linkability can be provided through a signature that includes a tag or a group manager who knows users’ identities. However, with a group manager, iterative user authentication, distribution of group keys, and signing are inefficient. Therefore, we propose an anonymous protocol based on an anonymous credential system [36]. In addition, we use a root signature-based one-time key and public key certificate system. This assures security through binding one-time public keys to a root public key. A user may own multiple pseudonym IDs according to the anonymous credential system, and other groups identify users with the pseudonym IDs [37]. The actual identity of the user is concealed by using the pseudonym IDs, and user identification and linking capabilities are provided through the private blockchain server (PBS) when users act maliciously. In addition, the PBS’s signature ensures the authentication of its members and messages to other groups in the consortium blockchain [44]. Finally, we generate an efficient, compressed signature using aggregate signatures.

4.2. Security Requirements

In this subsection, we describe the requirements for creating an anonymous protocol with user identification and linking capabilities for user privacy in a permissioned blockchain. At the same time, we analyze the advantages and disadvantages of existing methods for satisfying these requirements.

1) Authorization: The permissioned blockchain only allows users who have been identified to join the network. Only authenticated users have authorization to access the services. Only users
authenticated by the permissioned blockchain can generate signatures. Signatures are verified only by group members. Signatures that fail verification are invalid.

2) Confidentiality: Confidentiality means that a message is not visible to other users. Sensitive information that requires confidentiality can be encrypted as in [19,20]. Encryption and decryption have a lot of overhead.

3) Prevention of inferring user’s identity: If a user uses the same key repetitively, it can allow for inferring with the user’s identity. Keys can be used as one-time-keys or changed periodically [17,25]. However, depending on the life cycle of the keys, it may be necessary to solve problems such as verification, revocation, and renewal of the keys.

4) Anonymity: In a group signature, a condition can be added that only the manager identifies the signer to improve anonymity [18]. Additionally, in permissioned blockchains, private blockchains and consortium blockchains use different methods to reduce manager overhead and improve privacy. Privacy needs to separate user verification and signature authentication to minimize information exposure.

User identification and linking capabilities: Sometimes a permissioned blockchain needs to trace user messages [31]. A tag is attached so that the signature of the same signer can be verified through a TTP. Reference [7] proposed a ring signature tag using Zk-SNARKs on the ring signature without a TTP. This method separates all the users’ identity information and only checks whether a message is generated from the same user. The requester verifies the user’s real identity information via TTP and only verifies whether the same user signed.

4.3. Overview

The proposed system is a permissioned blockchain. It is a consortium blockchain that consists of several private blockchains, which in turn consist of a PBS and users. The system is shown in Figure 5. (1) A user requests issuance of pseudonym ID using the user’s OffID (offline ID) and a one-time key pair to join the group. At this point, the root signature binds the user’s one-time public keys to the root public key. (2) The private blockchain issues public-key certificates for the user’s one-time public keys and root signature and manages the user identity list with an OffID and linking pseudonym ID for the user. (3) The PBS encrypts the pseudonym ID and transmits it to the user. (4) The user creates an anonymous transaction by signing a message using the pseudonym ID and a one-time private key. (5) The private blockchain generates a transaction for the consortium blockchain using the PBS’s aggregate signature.

![Figure 5. Entities and overview of proposed scheme.](image)
4.4. Entities in the System

In this section, we define the entities for our anonymous protocol for permissioned blockchain.

Private blockchain server: The private blockchain server takes on the roles of the public key certificate issuer, group signer, and user identifier. We abbreviate this to PBS.

Users: Users act as group members, signature verifiers, and linkers.

Public key certificate issuer: The public key certificate issuer issues a public key certificate to the users using the PBS’s aggregate signature upon user request.

Group signer: The group signer creates a group signature for the message using the PBS’s private key and aggregate signature.

Verifier: The verifier verifies whether a transaction, which is signed using the PBS’s aggregate signature, was created using a user’s private key.

User identifier: For the user’s identity requests, the user identifier discloses the identity to the requester using OffIDList and a public key certificate.

Linker: Linker computes whether two transactions were created by the same signer without revealing the user identity.

4.5. Workflow

4.5.1. System Setup Phase

In this phase, the PBS sets the appropriate security strength according to the security policy and generates key pairs. The PBS generates a user OffIDList, and the public key and private key for user identification.

4.5.2. Join Phase

In this phase, the user sends a one-time public key, root signature, and OffID to the PBS to obtain a public key certificate and pseudonym ID. Then, the PBS updates the user identities list List = (List[1], ..., List[n]). Identity information about the ith registered user is included. Additionally, it creates a pseudonym ID list. Finally, the PBS returns the pseudonym IDs encrypted with the user’s public key and the user’s public key certificate with the PBS’s signature.

4.5.3. Aggregate Signature Phase

In this phase, the user creates a signature, which is then committed as a transaction on the consortium blockchain. The aggregate signature consists of two phases: signature creation in a private blockchain and signature creation in the consortium blockchain. In the signature creation in a private blockchain phase, the user creates a signature using a pseudonym ID. Subsequently, in the signature creation in the consortium blockchain stage, PBS compresses a message signed with pseudonym IDs of multiple users and broadcasts it to the consortium blockchain.

(3-1) Signature creation in a private blockchain: The user creates a hash value with a message and a one-time private key and sends it to the PBS together with their pseudonym ID and the public key certificate. The PBS validates the user through the user’s public key certificate and creates a transaction with the user’s hash value, which is signed using the PBS’s one-time private key and sent to the private blockchain. Finally, if more than half of the private blockchain’s members validate the PBS’s signature completely, then the transaction will be added and committed to the private blockchain.

(3-2) Signature creation in the consortium blockchain: The PBS’s signature is propagated to the consortium blockchain as a single group signature through an aggregate signature.

4.5.4. Validation Signature Phase

In this phase, verifiers in the consortium blockchain who want to verify the message can verify that the message was generated by a user of a legitimate private blockchain by verifying the PBS’s signature.
4.5.5. User Identification and Linking Capability Phase

In this phase, upon request, the PBS provides member identity verification. It provides user identification through private blockchain records, public key certificates, and OffIDList. A requestor can have local connectivity only by calculating whether a transaction is created from the same member through a Bloom filter of a private blockchain.

4.5.6. Renewal and Revocation of Public Key Certificate Phase

In this phase, the PBS performs renewal and revocation of public key certificates upon withdrawal and/or rejoining of members. The user transmits both the one-time key pair and root signature created previously to the PBS to request the renewal or revocation of the public key certificate. After that, the user can request a new certificate through a newly created one-time key pair and the root signature.

4.6. Protocol

This section describes the protocol of formulas according to the workflow in Section 4.5. The protocol consists of seven phases: a system set up, join, aggregate signature, signature validation, user identification and linking, renewal, and revocation public key certificate (Figure 6).

![Figure 6. Scenarios of the proposed scheme.](image)

4.6.1. System Parameters

Table 5 shows the notation and properties of the system parameters of the proposed scheme.


| Symbol     | Quantity                                                                 |
|------------|--------------------------------------------------------------------------|
| PBSi       | Private blockchain server subset PBSi ⊆ ConsortiumBlockchain(CB)         |
| UIi        | Private blockchain user subset UIi ⊆ Ui                                 |
| KOpen, Wopen, Vopen, Yopen, Zopen, Lopen, Ropen, Sopen, Sopen, Sopen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lopen, Ropen, Sopen, Sopen, Lop
4.6.3. Join Phase

**Step 1.** The user sends certificate information \((ID_U, \sigma, l_U, PK_{UI})\) to the PBS and requests a public key certificate.

**Step 2.** The PBS verifies \(R\) using the one-time public key \(PK_{UI}\) of the public key certificate information sent by user.

\[
P'_i = H( R \times ID_S ) \times G + ID_U, (\sigma, g_2) = \prod_{i=i}^{K} e(P_i, PK_U) \quad ID_S \times R
\]

\[
= D_S \times r \times G = ID_{SB} \times r \quad E(\sigma, g_2) = e(P'_i, g_2^{SK_{UI}}) = (P'_i, PK_{UI}) = OK
\]

**Step 3.** The PBS proceeds with the following if the verification of the user’s public key list is valid, and stops if it is not.

**Step 4.** The PBS selects random numbers \(x, y \in Zp^*\) and then calculates \(A\) after summing \(LIST\) at \(n+i\)-th list.

\[
A = [1/(\theta + x)] (Q_1 - [y]Q_2 - Z) (= [1/(\theta + x)] (Q_1 - [y]Q_2 - [z]W))
\]

\[
LIST[n + 1] = (ID, [y]Q, A, x, y, upk[i] = Z, TID)
\]

**Step 5.** The PBS signs the set of user identity list parameters \((i, A, x, y)\) and propagates it to the private blockchain.

\[
H(r \times (i, A, x, y)) \times G + ID_{PB}
\]

**Step 6.** The PBS generates certificate signature information for multiple users using the aggregate signature \(\sigma_B\).

\[
P'_i = H( r \times ID_S ) \times G + ID_U, \sigma_U = \sum_{i=1}^{n} p_i^{SK_{UI}}
\]

**Step 7.** The private blockchain verifies the signature of the PBS.

\[
ID_S \times R = ID_S \times r \times G = ID_S \times r
\]

\[
E(\sigma, g_2) = e(P'_i, g_2^{SK_{UI}}) = (P'_i, PK_{SI}) = OK
\]

**Step 8.** After verification, the private blockchain collects the aggregated signature from the PBS and manages its public key certificate as a leaf node of a hash tree.

**Step 9.** The PBS returns a public key certificate \((ID_U, \sigma_U, l_U, PK_{UI}, ID_S, \sigma_S)\) containing \(ID_S\) and verification \(\sigma_S\) data to the user.

**Step 10.** The user verifies the public key certificate including the ID and verification data \(\sigma_S\).

\[
ID_S \times R = ID_{SA} \times r \times G = ID_S \times r
\]

\[
E(\sigma, g_2) = e(P'_i, g_2^{SK_{SI}}) = (P'_i, PK_{SI}) = OK
\]

4.6.4. Aggregate Signature Phase

(1) **Signature creation in private blockchain**

**Step 1.** User sends the public key certificate \((ID_U, \sigma_A, l_U, PK_{UI}, ID_S, \sigma_S)\), and one-time key pair \(SK_{AI}\) and \(SK_{UI}\), to the PBS for authentication and simultaneously creates the pair \(x = H(m, SK_{UI})\) to request a signature.
Step 2. The PBS verifies the authenticity of the user by the one-time private key $SK_{Ui}$, public key $PK_{Ui}$, and root public key value with the public key certificate via the hash tree of the private blockchain. The public key certificate with the PBS’s signature is validated PBS’s aggregate signature.

\[
E(\sigma, s_2) = e \left( p_i^e, s_2^{SK_i} \right) = \left( p'_i, PK_{Si} \right) = OK \tag{15}
\]

\[
PK_{Ui} + s_2^{SK_{Ui}} = OK \tag{16}
\]

\[
P_i = H \left( r \times ID_3 \right) \times G + ID_{Ui}, \sigma_U = \sum_{i}^{n} p_i^{SK_{Ui}} \tag{17}
\]

(2) Signature creation in consortium blockchain

Step 1. The consortium blockchain network generates a global hash tree using the relationship $x = H \left( m, SK_{Ui} \right)$, and then a global timestamp value $S_i$, by linking the global coordinated time to the hash tree root value.

Step 2. The consortium blockchain network broadcasts the user’s anonymous ID to the PBSs after generating token $(S_i, SK_{Si})$ and returns the token to the user.

4.6.5. Signature Validation Phase

Step 1. The user sends their one-time private key, the token $(S_i, SK_{Si})$ for the message, and $E(M)$ encrypted using the verifier’s public key to the verifier.

Step 2. The verifier calculates the signature token of the consortium blockchain using the Merkle tree.

Step 3. The verifier verifies the signature of the private blockchain.

\[
ID_{SA} \times R = ID_3 \times r \times G = ID_{SA} \times r \tag{18}
\]

\[
E(\sigma, s_2) = e \left( p_i^e, s_2^{SK_{Si}} \right) = \left( p'_i, PK_{Si} \right) = OK \tag{19}
\]

4.6.6. User Identification and Linking Capability Phase

Step 1. The PBS performs the following process using pseudonym ID values $LV = \left( r, D_1, D_2, D_3, c, sa, sx, s_{r}, sy \right)$ and its public key $g_{nok} = \left( \eta, \xi \right)$.

Step 2. The PBS calculates $[y]Q = D3 - [\xi]D1$ and searches for the signer’s real identity corresponding to entry LIST $[I] = (ID, [y]Q, A, x, y, upk[i] = Z)$. If user $i$ is found, the PBS selects $r \in Z_p$ and calculates $K_{open}, W_{open}, V_{open}, c_{open}$, and $s_{open}$.

\[
K_{open} = [\eta]D1, W_{open} = [r]U, V_{open} = [r]D1 \tag{20}
\]

\[
c_{open} = Hp(\sigma || g || K_{open} || W_{open} || V_{open}) \tag{21}
\]

\[
s_{open} = r + c_{open} \eta \left( \text{mod} \ p \right) \tag{22}
\]

Step 3. The PBS calculates $T = (i, \tau = (K_{open}, c_{open}, s_{open}), upk[i] = Z, Y_{1i} = [yi]Q_2, Y_{2i} = [yi]B_1, i = [xi]Q, \text{Calculate} X_{2i} = [xi]B_1$ and discloses the identity of $Ui$ to the requester.

Step 4. If a requester or an evaluator, such as a judge, wants to verify the relationship between the pseudonym ID verification value $\sigma$ and the user $i$, the public key certificate issuance process is performed to verify the user’s public key verification value, public key, public key certificate issuance information, and private blockchain. The user’s identity is verified through the public key certificate and information list.

Step 5. To determine whether two transactions were done by the same user, the linker creates a Bloom filter set using the two transactions with the transaction signature of the consortium blockchain.
Theorem 2. Let \((G_1, G_2)\) be a \((t', \epsilon')\)-bilinear group pair for co-Diffie–Hellman, with each group of order \(p\), with respective generators \(g_1\) and \(g_2\), with an isomorphism computable from \(G_2\) to \(G_1\), and with a bilinear map \(e : G_1 \times G_2 \rightarrow G_T\). Then the bilinear aggregate signature scheme on \((G_1, G_2)\) is \((t, q_H, q_S, N, \epsilon)\)-secure against existential forgery in the aggregate chosen-key model for all \(t\) and \(e\) satisfying \(e \geq e(q_S + N) \cdot \epsilon\) and \(t \leq t' - G_1(q_H + 2q_S + N + 4) - (N - 1)\), where \(e\) is the base of natural logarithms, and exponentiation and inversion on \(G_2\) take time \(eG_1\).

(2) Anonymity: In our proposed scheme, a private blockchain server (PBS) issues and manages pseudonym IDs of the users to ensure the users’ unlinkability. Additionally, a PBS creates an aggregate signature for the consortium blockchain to ensure the user’s untraceability. The user uses the created one-time key pairs \((SK_{UI}, PK_{UI})\) and pseudonym IDs are issued via the one-time key pairs. Additionally, signature forgery of legitimate users is very difficult for the following reasons: Forgery of the user’s key pair \((SK_{UI}, PK_{UI})\) is very difficult based on the difficulty of the elliptic curve discrete logarithms problem (ECDLP) [45]. Creating a root signature is exceedingly difficult, depending on parallelism and the DH [46].

Theorem 1. Given a primitive element \(P\) and another element \(T\) on an elliptic curve, the ECDLP problem is to find the integer \(d\), where \(1 < d < qE\) such that \(P + \ldots + P = dP = T\), \(|P| = d\) times 3.

4.6.7. Renewal and Revocation Public Key Certificate Phase

Step 1. The user sends all their one-time key pairs together with their public key certificate to the PBS.

Step 2. The PBS validates the one-time key pairs and discards the existing public key certificate and identity list.

5. Analysis

We compare the security of the proposed scheme with other schemes according to Section 4.1 security requirements (Table 6).

5.1. Analysis of Security

(1) Authorization: We propose an anonymous credential system for permissioned blockchain. Our scheme reduces overhead and performs authentication efficiently using an anonymous credential system. \(\sigma, t_0\) are used to calculate \(SK_{UI}\) so only the user can create it, and it can be verified with \(ID_{UI}, PK_{UI}\). Additionally, \(\sum_i p\text{in}^k\text{s}_i\) is generated by the PBS only through \(SK_{SI}\) is \(E\). \(E(\sigma, g_2) = e(P_i, g_{2^k_i}) = (P_i, PK_{SI})\)

Permisioned blockchain generates signatures only through user-created key pairs \((SK_{UI}, PK_{UI})\) and public key certificates \((ID_{UI}, \sigma_{UI}, t_0, PK_{UI}, ID_S, \sigma_S)\).

Additionally, signature forgery of legitimate users is very difficult for the following reasons: Forgery of the user’s key pair \((SK_{UI}, PK_{UI})\) is very difficult based on the difficulty of the elliptic curve discrete logarithms problem (ECDLP) [45]. Creating a root signature is exceedingly difficult, depending on parallelism and the DH [46].
Table 6. A comparison of features of schemes for privacy in blockchain.

| Scheme                      | Authorization | Confidentiality | Prevention of Inferring Users’s Identity | Anonymity | Traceability (Conditional) | Anonymity Revocation |
|-----------------------------|---------------|-----------------|------------------------------------------|-----------|---------------------------|---------------------|
|                             |               |                 | Non-Linkability                          | Non-traceability |                          | User Identification | Linking          |
| Mercer, R. [7]              | Public issuer | Offered         | Weak                                     | Not offered | Ring signature            | Private issuer      |
| Ruffing et al. [15]         | Private issuer| Not Considered  | Weak                                     | Mixing server | Mixing server            | Not offered         |
| Hardjono et al. [18]        | Private issuer| Not Considered  | Strong                                   | ZKP        | ZKP                       | Private issuer      |
| Zhang et al. [19]           | Private issuer| Offered         | Weak                                     | Pseudonym ID | Not offered               | Private issuer      |
| Jiang et al. [20]           | Public issuer | Offered         | Weak                                     | OTKS + OMG | OTKS + OMG                | Private issuer      |
| Dubovitskaya et al. [21]    | Private issuer| Offered         | Weak                                     | MSPI + Gabled Bloom filter | MSPI + Gabled Bloom filter | Private issuer      |
| Zhu et al. [22]             | Private issuer| Offered         | Weak                                     | Gabled Bloom filter | Gabled Bloom filter | Private issuer      |
| Aitzhan et al. [23]         | Public issuer | Not Considered  | Weak                                     | Anonymous messaging stream | Anonymous messaging stream | Not offered         |
| Goldfeder et al. [24]       | Public issuer | Offered         | Strong                                   | Off Chain  | Off Chain                 | Not offered         |
| Axon et al. [25]            | Public issuer | Offered         | Strong                                   | Off chain + Blind signature | Off chain + Blind signature | Private issuer      |
| Heilman et al. [26]         | Private issuer| Offered         | Weak                                     | Aggregate signature | Anonymous credential system + PKI | Private blockchain + Bloom filter |
| Proposed Scheme             | Private issuer| Considered      | Strong                                   | Pseudonym ID | Anonymous credential system + PKI | Private issuer + PKI |


Prevention of inferring user’s identity: We need to improve key updates to prevent others from inferring a user’s identity through a user’s repeated use of the same key. However, we prefer to avoid a complex secret computation operation. We propose an ECC-based one-time key and public key certificate system. An aggregate signature-based root signature is a binding one-time public key and private key. It provides efficient verification of one-time keys and defends against security threats such as changing keys by malicious actors. Additionally, our scheme uses one-time keys to prevent user inference and cryptographic algorithms from solving problems such as one-time-key verification, revocation, and renewal.

(3) Confidentiality: Our scheme does not need encryption as a default. However, if the users need to perform encryption, an encryption key will be generated and used by our protocol.

(4) User identification and linking capabilities: We propose unmasking the user identity through a PBS user identity list and PKI. Additionally, the private blockchain network node acts as a linker. It can provide link information to the verifier upon request. When necessary, user identification and linking capabilities can be provided by revoking anonymity. The PBS has a list of identity information provided by the user \((1D_U, \sigma, PK_U)\) when requesting a public key certificate. User identification is provided according to List \([i] = \text{List}_1, \ldots, \text{List}_t\).

Our scheme further minimizes the exposure of users’ information by separating the roles of user identification and linking. The linker has \(Kopen = [\eta]D1, Wopen = [r]U, Vopen = [r]D1, copen = H_p(\sigma || g || Kopen || Wopen || Vopen), sopen = r + copen \eta \mod p^2 t = (i, T = (Kopen, copen, sopen), upk[i] = Z, Y1, i = [yi]Q2, Y2, i = [yi]B, X1, i = [xi]Q, X2, i = [xi]B).\) The requestor can determine if transactions are linked through the above parameters.

5.2. Analysis of Efficiency

Existing blockchain transactions, \(n\) in number, are proportionally increased in size by 71 bytes through the ECDSA signature. However, the proposed scheme generates signatures for efficient single verification through double-linear mapping and ECC-based signature and public key generation. The length of the trapdoor increases with the user identification and linking capabilities of the group signature compared to the existing group signature, but providing a single compressed signature provides the same 160-bit signature regardless of \(n\). In addition, the group authentication through anonymous credit has reduced the computational count of members of general group signatures.

Tables 7 and 8 show that zero-knowledge verification has relatively fast key size and verification but requires separate key generation and verification, which is slow. In particular, according to [47,48], Table 8 compared each execution time. Based on this rationale, we can calculate the time for each skill’s basic performance. We considered the calculation time of each technique according to the criteria of Table 8. The protocol’s basic implementation steps of other proposed technologies were checked and time was calculated. Figure 7 compares the speeds of the proposed schemes with the schemes that provide privacy in the existing blockchain. The median value is abbreviated as a tilde. The highest point and the lowest point are connected to the peak point. Verification time is 212 ms, and key generation time is increased to 1000 ms compared to the schemes [20,21], but it is very efficient because no verification time is required.
Table 7. A comparison of the efficiencies of schemes for privacy in blockchain.

| Scheme                         | Tx Size (bytes) | Key Gen Time (ms) | Proof Time (ms) | Verify Time (ms) | Number of Computations |
|--------------------------------|-----------------|-------------------|-----------------|------------------|------------------------|
| Mercer, R. [7]                 | 417             | Less than 1 s     | Proof time * O (n) | 211.8           | 1ECDSA + AE * O (n) 2VF +1DE |
| Ruffing et al. [15]            | 25 * O (n) + 321| Less than 1 s     | -               | 211.8            | M+1AE+1ECDSA + 2VF+1DE  |
| Hardjono et al. [18]           | 346             | 1,230,000         | 1,470,000       | 5.1              | 1ZKP + 1ECDSA +1AC+ 1VF |
| Zhang et al. [19]              | 322             | 1,230,000         | 1,470,000       | 4.68             | 1ECDSA + AE * O (n) 1ZKP+2VF |
| Jiang et al. [20]              | 400 + KW * O (n) | Less than 1 s     | -               | 219              | 1ECDSA + AE * O (n)+ 2VF |
| Dubovitskaya et al. [21]       | 400 + KW * O (1) | Less than 1 s     | -               | 212              | 1ECDSA + AE * O (n)+ 1VF |
| Aitzhan et al. [23]            | 320             | Less than 1 s     | -               | Less than 1 s    | 2ECDSA +2 VF           |
| Goldfeder et al. [24]          | 322             | 1,230,000         | 4,410,000       | 4.68             | 1ECDSA + 3ZKP+1 KR     |
| Axon et al. [25]               | 346             | Key gen time * O (n) | -            | 211.8            | 1ECDSA + AE * O (n)+ 2VF |
| Heilman et al. [26]            | 350             | Less than 1 s     | -               | Less than 1 s    | 1ECDSA + AE * O (n)+2VF |
| Proposed Scheme                | 346             | Less than 1 s     | -               | 212              | 1ECDSA + AE * O (n)+ 2VF |

ECDSA: Elliptic Curve Digital Signature Algorithm. ZKP: Zero Knowledge Proof. AC: Access control. VF: Signature Verification. AE: Asymmetric Encryption. DE: Decryption.

Table 8. A comparison of the efficiencies of cryptography methods.

| Scheme                          | Key Size (bytes) | Signature Size (bytes) | Key Gen Time (ms) | Sig Gen Time (ms) | Proof Time (ms) | Verify Time (ms) |
|---------------------------------|------------------|------------------------|-------------------|------------------|-----------------|-----------------|
| RSA 1024 with SHA 256           | 1024             | 128                    | 74                | 5.2              | -               | 0.4             |
| ECC secp 192k1 with SHA 256     | 192              | 55                     | 117               | 6.6              | -               | 211.8           |
| ZK-SNARKs(128 bit security)     | -                | 346                    | 123,000           | -                | 147,000         | 5.1             |
| Zerocash                        | -                | -                      | 303,000           | -                | -               | 60,000          |

Figure 7. Comparison of efficiencies with various times for the existing scheme and our scheme.

6. Conclusions

Unlike a permissionless environment, e.g., Bitcoin, where the public network allows network participation without third-party approval, a permissioned blockchain is composed of a set of known, identified nodes that might not completely trust each other [2]. Permissioned models control who participates in the validation protocol; these nodes usually have constructed identities and form a consortium [3]. This permits the transactions created in a private blockchain to be shared as a consortium blockchain. However, if identity information is included or inferred in the transaction created, the sharing of transactions among groups in the consortium blockchain has a user privacy problem. To address this, the permissioned blockchain should be controlled not only with MSP but also use proper cryptography for privacy. Some existing schemes provide privacy by ensuring user unlinkability and untraceability. Unfortunately, a permissioned blockchain sometimes needs to track malicious users. Furthermore, it may be necessary to verify the user’s identity with a legitimate request.
This makes it necessary to impose partial anonymity to protect the user’s privacy while at the same time revealing the user’s identity and providing efficient linking.

In this paper, we proposed an anonymous protocol with improved user privacy in a permissioned blockchain to fulfill the requirements for anonymity, linking, and user identification capabilities. We used an anonymous protocol based on an anonymous credential system and a root signature-based one-time key and public key certificates system. It assures security through binding a one-time key pair. A user’s pseudonym ID may own multiple pseudonym IDs according to the anonymous credential system. Other groups may recognize users via their pseudonym IDs. We ensured privacy through anonymity via a pseudonym ID issued by an anonymous credential system and an aggregator signature. The identity of the user provides anonymity by revealing the user identification with the pseudonym ID and sometimes provides linking and user identification capabilities through the private blockchain. This enabled our proposed scheme to track malicious members in the permissioned blockchain. We generated an efficient compression signature using an aggregator signature. It can ensure the authentication of its members and messages to other groups in the consortium blockchain and provides an efficient signature size. We compared existing schemes with our scheme for privacy in blockchains according to the security requirements. We found that our proposed scheme has an advantage. In the future, we will improve our scheme to enable its use in the real world.

Author Contributions: Conceptualization, I.L. and G.R.; methodology, D.S. and G.R.; software, G.R.; validation, M.Z.A.B. and G.R.; formal analysis, D.S. and M.Z.A.B.; investigation, I.L.; resources, I.L.; data curation, G.R.; writing—original draft preparation, G.R.; writing—review and editing, M.Z.A.B. and G.R.; visualization, D.S.; supervision, I.L.; project administration, I.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2019R1A2C1085718) and was supported by the Soonchunhyang University Research Fund.

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

References
1. Swan, M. Blockcanin:Blueprint for a New Economy; O’Reilly Media, Inc.: Sebastopol, CA, USA, 2015.
2. Ali, S.; Wang, G.; Bhuiyan, M.Z.A.; Jiang, H. Secure Data Provenance in Cloud-Centric Internet of Things via Blockchain Smart Contracts. In Proceedings of the 2018 IEEE SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI, Guangzhou, China, 8–12 October 2018; pp. 991–998.
3. Amiri, M.J.; Agrawal, D.; Abbadi, A.E. CAPER: A cross-application permissioned blockchain. Proc. VLDB Endowment 2015, 12, 1385–1389. [CrossRef]
4. Cachin, C. Architecture of the hyperledger blockchain fabric. In Proceedings of the Distributed Cryptocurrencies and Consensus Ledgers “DCCL 2016”, Chicago, IL, USA, 25 July 2016; p. 4.
5. Bhuiyan, M.Z.A.; Zaman, A.; Wang, T.; Wang, G.; Tao, H.; Hassan, M.M. Blockchain and big data to transform the healthcare. In Proceedings of the International Conference on Data Processing and Applications, Guangzhou, China, 12–14 May 2018; pp. 62–68.
6. Omar, A.A.; Bhuiyan, M.Z.A.; Basu, A.; Kiyomoto, S.; Rahman, M.S. Privacy-friendly platform for healthcare data in cloud based on blockchain environment. Future Gener. Comput. Syst. 2019, 95, 511–521. [CrossRef]
7. Mercer, R. Privacy on the blockchain: Unique ring signatures. arXiv 2016, arXiv:1612.01188.
8. Sun, S.F.; Au, M.H.; Liu, J.K.; Yuen, T.H. RingCT 2.0: A compact accumulator-based (linkable ring signature) protocol for blockchain cryptocurrency Monero. In European Symposium on Research in Computer Security; Springer: Berlin/Heidelberg, Germany, 2017; pp. 456–474.
9. Yin, B.; Mei, L.; Jiang, Z.; Wang, K. Joint cloud collaboration mechanism between vehicle clouds based on blockchain. In Proceedings of the 2019 IEEE International Conference on Service-Oriented System Engineering (SOSE), Oxford, UK, 13–16 April 2019; pp. 227–2275.
10. Yang, H.; Wu, Y.; Zhang, J.; Zheng, H.; Ji, Y.; Lee, Y. BlockONet: blockchain-based trusted cloud radio over optical fiber network for 5G fronthaul. In Proceedings of the 2018 Optical Fiber Communications Conference and Exposition (OFC), San Diego, CA, USA, 11–15 March 2018.
11. Gentry, C. A Fully Homomorphic Encryption Scheme. Ph.D. Thesis, Stanford University, Stanford, CA, USA, 2009.
12. Noether, S.; Mackenzie, A. Ring confidential transactions. Ledger 2016, 1, 1–18. [CrossRef]
13. [ANN] Joinmarket—Coinjoin that People will Actually Use. 2015. Available online: https://bitcointalk.org/index.php?topic=919116.0 (accessed on 9 January 2015).
14. JoinMarket’s Privacy Is Degraded (for a While). 2016. Available online: https://gist.github.com/chris-belcher/00255ecfe1bc4984fcf7c65e25aa8b4b (accessed on 27 May 2020).
15. Ruffing, T.; Moreno-Sanchez, P.; Kate, A. P2P Mixing and Unlinkable Bitcoin Transactions. In Proceedings of the Network and Distributed System Security Symposium, San Diego, CA, USA, 26 February–1 March 2017.
16. Hopwood, D.; Bowe, S.; Hornby, T.; Wilcox, N. Zcash Protocol Specification; Zerocoin Electric Coin Company: 2016. Available online: https://github.com/zcash/zhits/blob/master/protocol/protocol.pdf (accessed on 27 May 2020).
17. Noether, S. Ring Signature Confidential Transactions for Monero; IACR Cryptology ePrint Archive: Lyon, France, 2015; p. 1098. Available online: https://eprint.iacr.org/2015/1098 (accessed on 11 November 2015).
18. Hardjono, T.; Pentland, A. Verifiable anonymous identities and access control in permissioned blockchains. arXiv 2019, arXiv:1903.04584.
19. Zhang, A.; Lin, X. Towards secure and privacy-preserving data sharing in e-health systems via consortium blockchain. J. Med. Syst. 2018, 8, 140. [CrossRef]
20. Jiang, P.; Guo, F.; Liang, K.; Lai, J.; Wen, Q. Searchain: Blockchain-based private keyword search in decentralized storage. Future Gener. Comput. Syst. 2017, 107, 781–792. [CrossRef]
21. Dubovitskaya, A.; Xu, Z.; Ryu, S.; Schumacher, M.; Wang, F. Secure and trustable electronic medical records sharing using blockchain. In Proceedings of the AMIA annual symposium, Washington, DC, USA, 4–8 November 2017; Volume 2017, p. 650.
22. Zhu, X.; Su, Y.; Gao, M.; Huang, Y. Privacy-preserving friendship establishment based on blind signature and bloom filter in mobile social networks. In Proceedings of the 2015 IEEE/CIC International Conference on Communications in China (ICCC), Shenzhen, China, 2–4 November 2015; pp. 1–6.
23. Aitzhan, N.Z.; Svetinovic, D. Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams. IEEE Trans. Dependable Secur. Comput. 2016, 15, 840–852. [CrossRef]
24. Gennaro, R.; Goldfeder, S.; Narayanan, A. Threshold-optimal DSA/ECDSA signatures and an application to Bitcoin wallet security. In Proceedings of the International Conference on Applied Cryptography and Network Security, London, UK, 19–22 June 2016; pp. 156–174.
25. Axon, L.M.; Goldsmith, M. PB-PKI: A privacy-aware blockchain-based PKI. In Proceedings of the 14th International Conference on Security and Cryptography 2016, London, UK, 19–22 June 2016.
26. Heilman, E.; Baldimtsi, F.; Goldberg, S. Blindly signed contracts: Anonymous on-blockchain and off-blockchain bitcoin transactions. In Proceedings of the International Conference on Financial Cryptography and Data Security, Bridgetown, Barbados, 22–26 February 2016; pp. 43–60.
27. Ouaddah, A.; Elalam, A.A.; Ouahman, A.A. Towards a novel privacy-preserving access control model based on blockchain technology in IoT. In Europe and MENA Cooperation Advances in Information and Communication Technologies; Springer: Berlin/Heidelberg, Germany, 2017; pp. 523–533.
28. Barber, T.P.; Payne, L.D. Method and System for Creation and Verification of Anonymous Digital Credentials. U.S. Patent 20180181745A1, November 2015. Available online: https://patents.google.com/patent/US9191370B2/en (accessed on 27 May 2020).
29. Kiayias, A.; Tang, Q. Traitor deterring schemes: Using bitcoin as collateral for digital content. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, 12–16 October 2015; pp. 231–242.
30. Boneh, D.; Gentry, C.; Waters, B. Collusion resistant broadcast encryption with short ciphertexts and private keys. In Proceedings of the Annual International Cryptology Conference, Santa Barbara, CA, USA, 14–18 August 2005; pp. 258–275.
31. Chaum, D. Blind signatures for untraceable payments. In Advances in Cryptology; Springer: Boston, MA, USA, 1983.
32. Zhu, Y.; Song, X.; Yang, S.; Qin, Y.; Zhou, Q. Secure Smart Contract System Built on SMPC Over Blockchain. In Proceedings of the 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Halifax, NS, Canada, 30 July–3 August 2018; pp. 1539–1544.
33. Lipmaa, H. Prover-efficient commit-and-prove zero-knowledge SNARKs. In Proceedings of the International Conference on Cryptology in Africa, Fes, Morocco, 13–15 April 2016; pp. 185–206.
34. Chaum, D. Security Without Identification: Transaction Systems to Make Big Brother Obsolete. Commun. ACM 1985, 28, 10. [CrossRef]
35. Kiayias, A.; Tsionis, Y.; Yung, M. Traceable signatures. In Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, 2–6 May 2004; pp. 571–589.
36. Wu, C.N.; Fan, C.I.; Huang, J.J.; Tseng, F.Y.; Kikuchi, H. Probably Secure Efficient Anonymous Credential Scheme. Int. J. Softw. Innov. (IJSI) 2018, 6, 18–35. [CrossRef]
37. Singh, A.; Fhom, H.C.S. Restricted usage of anonymous credentials in vehicular ad hoc networks for misbehavior detection. Int. J. Inf. Secur. 2016, 16, 195–211. [CrossRef]
38. Han, J.; Li, Y.; Chen, W. A Lightweight and privacy-preserving public cloud auditing scheme without bilinear pairings in smart cities. Comput. Stand. Interfaces 2019, 62, 84–97. [CrossRef]
39. Choi, S.G.; Park, K.; Yung, M. Short traceable signatures based on bilinear pairings. In International Workshop on Security; Springer: Berlin/Heidelberg, Germany, 2006; pp. 88–103.
40. Vukolic, M. Rethinking permissioned blockchains. In Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts, Abu Dhabi, UAE, 3–7 April 2017.
41. Chaudhry, S.A.; Farash, M.S.; Naqvi, H.; Sher, M. A secure and efficient authenticated encryption for electronic payment systems using elliptic curve cryptography. Electron. Commer. Res. 2016, 16, 113–139. [CrossRef]
42. Mahmood, K.; Chaudhry, S.A.; Naqvi, H.; Kumari, S.; Li, X.; Sangaiah, A.K. An elliptic curve cryptography based lightweight authentication scheme for smart grid communication. Future Gener. Comput. Syst. 2018, 81, 557–565. [CrossRef]
43. Yuan, C.; Xu, M.X.; Si, X.M. Research on a new signature scheme on blockchain. Secur. Commun. Netw. 2017, 2017, 4746586. [CrossRef]
44. Ra, G.J.; Seo, D.; Bhuiyan, M.Z.A.; Lee, I.Y. An anonymous protocol for member privacy in a consortium blockchain. In Proceedings of the International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage, Atlanta, GA, USA, 14–17 July 2019; pp. 456–464.
45. Zhang, J.; Cui, J.; Zhong, H.; Chen, Z.; Liu, L. PA-CRT: Chinese Remainder Theorem Based Conditional Privacy-preserving Authentication Scheme in Vehicular Ad-hoc Networks. IEEE Trans. Dependable Secur. Comput. 2019, 16, 271–286. [CrossRef]
46. Boneh, D.; Gentry, C.; Lynn, B.; Shacham, H. Aggregate and verifiably encrypted signatures from bilinear maps. In Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, 4–8 May 2003; pp. 416–432.
47. Jinasaena, T.; Meegama, R.; Marasinghe, R. Access Control of Medical Images using Elliptic Curve Cryptography through Effective Multi-Key Management in a Mobile Multicasting Environment. Comput. Sci. Eng. 2017, 7, 1–7.
48. Boneh, D.; Gentry, C.; Lynn, B.; Shacham, H. ZeroCash: Decentralized anonymous payments from bitcoin. In Proceedings of the 2014 IEEE Symposium on Security and Privacy, San Jose, CA, USA, 18–21 May 2014; pp. 459–474.