Universal Redactable Blockchain

Seung Wook Jung*
Konyang University, Nonsan-si, Chungcheongnam-do, Republic of Korea
swjung@konyang.ac.kr

Received: September 30, 2022; Accepted: December 4, 2022; Published: December 31, 2022

Abstract

Due to the immutability of blockchain, many applications are emerging by providing trust to the trustless Internet. However, immutability becomes a weakness in some applications that require modification and deletion, such as personal information and SNS(Social Network Service), and cannot use blockchain technology. Therefore, in this paper, we propose a redactable blockchain scheme that can be used universally regardless of the consensus algorithms and blockchain types. In addition, without using a heavy cryptographic algorithm, and needing to re-mine, the proposed scheme can delete and modify content in the ledger, so it is efficient. Moreover, the proposed scheme can modify and delete at the transaction level. Additionally, the proposed scheme supports encryption for protecting the privacy and there is no need to save the original content after modification or deletion.

Keywords: Erasable Blockchain, Modifiable Blockchain, Privacy

1 Introduction

After publishing Bitcoin [1], blockchain has received much attention as it can provide trust to the Internet without a trusted third party. Since the smartphone was released, people can take a photo or movie clips using the smartphone and upload these to SNS. Therefore, SNS services are prevalent. Blockchain-based SNS services such as Steemit also emerged. In SNS, there is a lot of personal information. However, blockchain-based SNS cannot delete or modify personal information. It is because the blockchain provides integrity by recording transactions on an immutable and verifiable ledger. Therefore, it has problems with 'the right to be forgotten' required by GDPR (General Data Protection Requlation)[2].

In this paper, we intend to support more diverse services based on the blockchain by proposing a universal redactable blockchain scheme that supports fine-grained modification and redaction without using a heavy cryptographic algorithm. The proposed scheme supports any consensus algorithms and any type of blockchain.

1.1 Problem statement and Motivation

In this paper, the redactable blockchain was studied because of the limitations of the following blockchain.

- Legal restrictions: GDPR requires 'right to rectification, 'right to withdraw consent', and 'right to be forgotten'. However, until now, the blockchain cannot be modified or deleted, so personal

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), 13(4):81-93, Dec. 2022
DOI: *10.58346/JOWUA.2022.I4.005*

*Corresponding author: Department of Cybersecurity, Konyang University, 32992 Daehak-ro 121, Nonsan-si, Chungcheongnam-do, Republic of Korea, Tel: +82-41-750-5597 information cannot be recorded on the ledger of the blockchain. Although not only monetary transactions but also various information are recorded in the ledger of the blockchain, it is impossible to record some contents that might be modified or deleted such as personal information.

- Wide range of applications: Services such as Steemit which is a blockchain-based SNS record various information on the blockchain. However, it is impossible to correct or delete sensitive information that has been written wrongly. Therefore, once information such as child porn, personal photos, and videos of a person is recorded in the blockchain, there is an irretrievable side effect. Therefore, to increase the usability of the blockchain, a blockchain that can be modified and deleted is essential for Web3.0 which does not use the off-chain server.
- Modification of contents without hard fork: When a problem such as a DAO (Decentralized Autonomous Organization) hacking incident [3] occurred, the Ethereum blockchain had no choice but to hard fork. because of the immutability of the blockchain. If it were a blockchain that can be modified and deleted, it would have been easily restored without a hard fork.

1.2 Key Contribution

This study aims to enable blockchain to be applied to more diverse application services by proposing a universal redactable blockchain that can be applied to any type of blockchain, and consensus algorithm. The followings are the main contributions of this study.

- This paper proposes a fine-grained redactable blockchain that can be modified and deleted at the transaction level.
- This paper proposes an efficient redactable blockchain without a heavy cryptographic algorithm.
- This paper proposes a redactable blockchain that can be used regardless of private blockchain and public blockchain.
- This paper proposes a redactable blockchain that operates independently of the consensus algorithm.
- This paper proposes a redactable blockchain that provides a basic framework for redaction and can be easily applied without major changes to the existing blockchain.
- This paper proposes a redactable blockchain that satisfies the legal requirements such as the ‘right to rectification’, ‘right to withdraw consent’, and ‘right to be forgotten’.

1.3 Organization of the Paper

This paper is outlined as follows. In Section 2, the paper briefly reviews the related works. This paper presents the proposed scheme in Section 3. Section 4 discusses the security of the proposed scheme. In Section 5, we discuss more advanced topics to prevent changing content without permission. Section 6 concludes this paper.

2 Related Works

The related works are well researched at [4]. This paper borrows and summarizes related works from [4].

The first paper to modify the content of a blockchain was [5]. [5] uses the chameleon-hash function which is a heavy cryptographic algorithm and has additional key management concerns. Moreover, in the scheme, whenever a transaction includes content that wants to be removed, the whole block containing the transaction needs to be deleted. The fine-grained modifiable blockchain is proposed in [6] that uses ciphertext-policy attribute-based encryption. That is a heavy cryptographic algorithm. [7] uses truncated hash value to compute the hash value of a block and employs a multi-chain structure. The limitation of [7] is that the sender decides the difficulty level of transaction modification, so a malicious user can set the difficulty level high enough to be unable to modify the transaction. Another limitation of [7] is whenever one wants to modify, the block must conduct re-mining. The proposed scheme in [8] does not use a heavy cryptographic algorithm, but the scheme requires that all transactions in a block originating from the same sender are modified. That is not realistic. [9] proposed a scheme that is based on a consensus-based voting mechanism that does not rely on trust assumptions or heavy cryptographic algorithms. However, the limitation of [9] is that it works on the block level not the transacation level. [10] proposed a redactable proof-of-stake-based blockchain, so it cannot be applied to the other consensus algorithm-based blockchain.

[11] proposed a scheme that allows a local deletion of data. This scheme has a limitation in that every node in the blockchain network conducts a redaction when needed. [12] proposes an approach to redact a blockchain through the use of an additional layer built on top of the blockchain itself. [12] requires that original data is still kept for validation, so the right to be forgotten of GDPR cannot be met.

This paper proposes a scheme that does not use the heavy cryptographic algorithm and supports transaction-level redaction. Also, it does not require keeping original content to support the right to be forgotten. Moreover, this paper's scheme supports universal redaction that is not limited to a specific consensus algorithm and supports public and private blockchains.

3 Proposed Scheme

3.1 Requirement

The proposed scheme satisfies the following requirements.

- To store personal information, personal information must be encrypted.
- Transaction hash value and block hash value should not be changed, so there is no need to mine again.
- By supporting fine-grained modification and deletion, modification and deletion are possible at the transaction level.
- No heavy cryptographic algorithm is used for deletion or modification, so it does not affect processing speed.
- Retactable blockchain must be applied regardless of the type of consensus algorithm and the type of blockchain.
- There is no need to save the original content after modification or deletion.

The first requirement is considering that it can be applied to both public and private blockchains. In other words, in the public blockchain, even if personal information recorded in the ledger is deleted or changed, an adversary can maintain a copy of the existing ledger. Therefore, this paper considers encryption. Because private blockchain is accessible only to permitted entities, encryption may not be considered in private blockchain depending on policy and access control conditions.

To satisfy the second and third requirements, modification and deletion must be possible at the transaction level, and the transaction hash value must not be changed. If not, the block containing a transaction must be changed whenever the modification of the transaction occurs. Therefore, the block must be re-mined through PoW(Proof-of-Work) or PoS(Proof-of-Stack). If a transaction is modified, it is not easy to decide a policy on whether a node that previously mined the transaction should mine again or a node that mines a new block should mine again for the modified block. Also, like Bitcoin, there are blockchain networks in which the amount of cryptocurrency minted through mining is predetermined. If re-mining occurred by the transaction modified, there is a problem with whether to provide cryptocurrency for re-mining. Also, who will participate in re-mining if cryptocurrencies are not provided for re-mining? In addition, if multiple modifications occur during one block time, there is a problem that a lot of computing resources must be consumed to process them in the case of PoW. Therefore, the transaction hash should not be changed.

As a result, the general way to satisfy the first, second, and third requirements is to find a function that satisfies the equation (1).

$$f(a,b,c,d) = f(a',b',c',d') \tag{1}$$

Here, a is the correction value, b is the ciphertext, c is the signature for the plaintext, d is the signature value for b, and c. a' is the new correction value, b' is the new ciphertext when the plaintext is changed, c' is the signature value when the plaintext is changed, and d' is the signature value for b' and c'.

For the fourth requirement, it is to find a function that satisfies equation (1) and does not use heavy cryptographic algorithms such as chameleon hash [13] and pairing-based encryption [14].

For satisfying the fifth requirement, a proposed scheme should be a universal redactable scheme that can be applied regardless of the consensus algorithms such as PoW, PoS, DPoS(Delegated Proof-of-Stack), etc.

Existing methods [9] [12] require that the original content is saved even after redaction occurs. The proposed scheme should overcome this limitation for satisfying the sixth requirement.

3.2 Notations

Table 1 summarizes the notations used in this paper.

3.3 Proposed Scheme

3.3.1 Overall Architecture

Figure 1 shows overall architecture and processes. At first, a user sends a transaction with the digital signature of the user using the user's private key. The transaction can be a plaintext transaction, a

Table 1: Notations

notation	explanation	notation	explanation
TN	the transaction number	T	the type of transaction
A	transaction sender's address	B	transaction receiver's address
P	the plaintext	S_1, S_2	the signature value
pri	the private key	I	the correction value
I	the correction value	$\oplus$	exclusive OR
$Sign()$	an elliptic curve-based digital signature algorithm		
$H()$	cryptographically secure hash function		

redactable plaintext transaction, or a redactable encrypted transaction. The plaintext transaction may not include any personal information and cannot be redactable. The redactable plaintext transaction may include personal information that could be modified or redacted. The redactable plaintext transaction may be used in the private blockchain, so only the permitted entity can access the personal information. The redactable encrypted transaction may include encrypted personal information and could be modified or redacted. The redactable encrypted transaction can be used in the public blockchain.

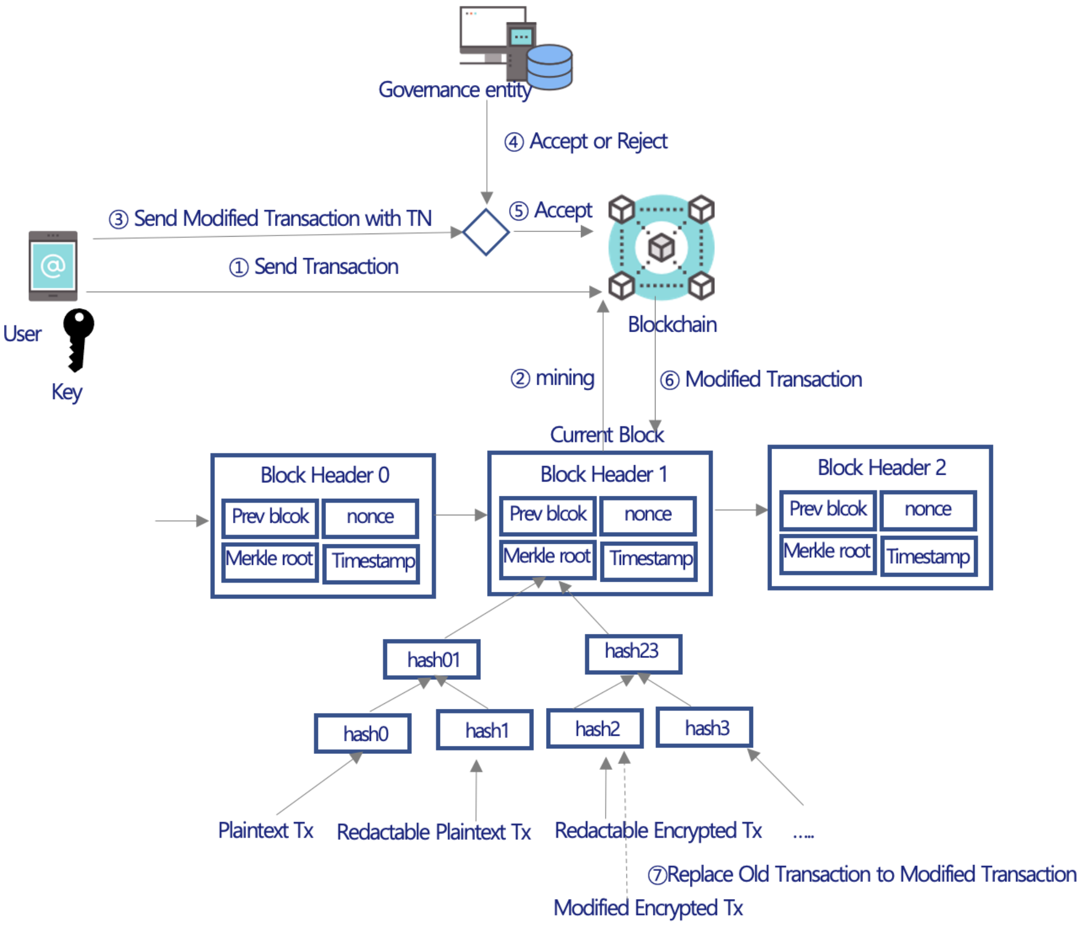

Figure 1: Overall Architecture

After the user sends the transaction to a blockchain full node, the full node mine the current block of which structure shows in figure 1. The block header consists of a previous block header hash value, a nonce, a timestamp, and a Merkle root.

The Merkle tree [15] is essentially a binary tree that stores the hash values. As figure 1 shows, from the leaf nodes to the root node, it will constantly combine two adjacent hash values into a string and calculate the hash of the string until getting the hash of the root node. In figure 1 shows three types of leaf nodes that are the plaintext transaction, the redactable plaintext, and the redactable encrypted transaction.

After the full node mines the current block, if the user wants to modify or redact the transaction, the user sends the modified transaction. In this example, there is a governance entity that can accept or reject the modified transaction. If the governance entity accepts the modification of the transaction, the governance entity relays the modified transaction to the full node. In Figure 1, the user sends the modified transaction of the redactable encrypted transaction. The full node replaces the old redactable encrypted transaction with the new modified encrypted transaction.

In a cryptographically secure hash function, changing just one bit changes more than half of the result. Therefore, if the transaction is replaced by another transaction, the hash value of the transaction (hash2 in figure 1) is changed. Also, hash23 and Merkle root are changed. However, our proposed scheme does not need any of the hash values in the Merkle tree. The detail of the proposed scheme is explained in the following subsections.

3.3.2 Plaintext Transaction Structure

The plaintext transaction structure is as follows.

$$Tx = TN|T|A|B|P|Meta|S_1 \tag{2}$$

TN stands for a transaction number and T stands for the type of the transaction. The type of transaction can be an immutable plaintext transaction, a redactable plaintext transaction, and a redactable encrypted transaction, so the type field is required. A indicates sender's address, B indicates receiver's address, and P stands for the plaintext. B is an optional field. P may be a combination of several elements rather than a single element. For example, $P = Tx_{Input}|Tx_{output}$, where Tx_{input} and Tx_{output} represent the transaction input and transaction output of Bitcoin. $Meta$ is an optional field to contain the information of P. S_1 denotes $Sign_{pri}(T|A|B|P|Meta)$. $Sign$ represents an elliptic curve-based signature algorithm like ECDSA(Elliptic Curve Digital Signature Algorithm)[16], pri represents A's private key and the signature provides integrity by signing the part that should not be changed in the transaction.

When the blockchain node receives the transaction, the node verifies the signature. If the result of the verification is a success, the transaction is included in the block when mining the block. When mining the block, the node creates $H(T|A|B|P|Meta|S_1)$ and puts it in the Merkle tree, where H represents a cryptographically secure hash function.

3.3.3 Redactable Plaintext Transaction Structure

The following shows the redactable plaintext transaction structure.

$$Tx = TN|T|I|Sign_{pri}(I)|A|B|P|Meta|S_1 \tag{3}$$

The structure is the same as that of a plaintext transaction, but I and the signatures for I are added. Where I represents the correction value and P represents the plaintext. The value of T is the type of P such as redactable plaintext or immutable plaintext. A is the sender's address, B is the receiver's address, and $Meta$ is used to indicate the information of the plaintext such as personal information or bank account. Here, $Meta$ and B are optional values.

When a blockchain node receives the transaction, the node verifies $Sign_{pri}(I)$ and
$S_1 = Sign_{pri}(T|A|B|P|Meta)$. If the result of the verification is a success, then the node puts the transaction into a block. When creating the block, the node creates $H(T|A|B|I \oplus P \oplus S_1|Meta)$ and inserts it into the Merkle tree.

Let's look at the process of deleting and changing plaintext. Let P' stand for changed P. When P is changed, S_1 is also changed. Let the changed signature value be S_1'. What we should find is I' that satisfies the equation (4).

$$I' \oplus P' \oplus S_1' = I \oplus P \oplus S_1 \tag{4}$$

In the end, $I' = I \oplus P \oplus S_1 \oplus P' \oplus S_1'$. I, P, and S_1 are written in the blockchain ledger, so these are publicly known. The sender A knows P' and S'. Therefore, the sender A can find I'. When the value of I' is found, a new transaction is created as shown in the equation (5). The sender A creates and propagates the transaction to a full node that has the blockchain ledger and mines blocks. The full node that received the transaction verifies $Sign_{pri}(I')$ and S_1' with A's public key. If it is correct, the full node finds the original transaction with TN and verifies the existing S_1 to check if the transaction is created by the same user. If the result of the verification of S_1 is a success, the original transaction is changed to the new transaction. Finally, the full node propagates the new transaction to other full nodes.

$$Tx = TN|T|I'|Sign_{pri}(I')|A|B|P'|Meta|S_1' \tag{5}$$

a and a' in the equation (1) are I and I' in the equation (4) respectively. This is the redactable plaintext transaction structure, so b and b' in the equation (1) are the plaintexts P and P' in the equation (4) respectively. Also, c and c' in the equation (1) are the signatures S_1 and S_1' for the plaintexts P and P' in the equation (4) respectively. There is no ciphertext in the equation (4), so d and d' are omitted. Therefore, the equation (4) satisfies the equation (1).

This is the redactable plaintext transaction, so the first requirement does not consider. Although Tx is changed, $H(T|A|B|I' \oplus P' \oplus S_1'|Meta)$ in the Merkle tree is the same as $H(T|A|B|I \oplus P \oplus S_1|Meta)$. It is because the equation (4) is satisfied. Therefore, there is no need to change the Merkle tree and thus there is no need to change the block containing the transaction. Therefore, the second requirement is satisfied. Moreover, it changes at the transaction level(the third requirement), so there is no need for re-mining. It uses a simple XOR(Exclusive OR) operation for changing the plaintext very efficiently without using a heavy cryptographic algorithm(the fourth requirement). Also, it does not require re-mining, so it can be applied to any consensus algorithm(the fifth requirement). However, the redactable plaintext transaction structure could be used for the private blockchain. Finally, there is no need to keep the original contents (the sixth requirement).

A redactable plaintext transaction structure can be used when only permitted nodes can access the ledger, such as a private blockchain.

3.3.4 Redactable Encrypted Transaction Structure

The following redactable encrypted transaction can be used to store sensitive information such as personal information in a public blockchain that anyone can access, unlike a redactable plaintext transaction.

The redactable encrypted transaction structure is shown in equation (6).

$$Tx = TN|T|I|Sign_{pri}(I)|A|B|E|Meta|S_1|S_2 \tag{6}$$

Where T is the type of transaction. In the case of ciphertext, the value is e. I is the correction value, and A is the address of the owner of the ciphertext. B is the receiver's address and is an optional value. $Meta$ can indicate the type of plaintexts, such as social security number and address, and is an optional value. Here, E is the ciphertext obtained by encrypting the plaintext P. The algorithm used for encryption can be a secure symmetric encryption algorithm such as AES256(Advanced Encryption Standard) [17]. A has to keep the symmetric encryption key securely. S_1 is $Sign_{pri}(T|A|B|P|Meta)$ and S_2 is $Sign_{pri}(T|A|B|E|Meta|S_1)$

When a blockchain node receives the transaction, the node verifies $Sign_{pri}(I)$ and S_2. Since S_1 is included in S_2, the integrity of S_1 is valid when S_2 is verified. If the verifications are successful, a block containing the transaction is created.

When creating the block, the node creates $H(T|A|B|I \oplus E \oplus S_1 \oplus S_2|Meta)$ and puts it in the Merkle tree to create the block.

Assume that C requests personal information from A. C sends a request message to A. A receives the request message and judges whether C is a legitimate user. If C is a legitimate user, A finds its own transaction using $Meta$ information. If the transaction is found, E is extracted from the transaction and decrypted to obtain plaintext P. Note the owner of E keeps the symmetric encryption key. A sends P and TN to C. C tries to find the transaction from the blockchain ledger with TN. If C attains the transaction, C tries to verify the signature $S_1 = Sign_{pri}(T|A|B|P|Meta)$ with the public key of A. If the verification succeeds, C accepts P. Of course, the communication channel between C and A is secure for confidentiality and integrity. The communication channel could be TLS(Transport Layer Security).

Let's look at deletion and modification. Let P' be the changed P. When P is changed, E, S_1 and S_2 are changed to E', $S'_1 = Sign_{pri}(T|A|B|P'|Meta)$ and $S'_2 = Sign_{pri}(T|A|B|E'|Meta|S'_1)$ respectively. What we want to find is a new correction value I' that satisfies the following equation (7).

$$I' \oplus E' \oplus S'_1 \oplus S'_2 = I \oplus E \oplus S_1 \oplus S_2 \tag{7}$$

That is, $I' = I \oplus E \oplus S_1 \oplus S_2 \oplus E' \oplus S'_1 \oplus S'_2$. I, E, S_1, and S_2 are written in the blockchain ledger, so these are known publicly. The sender A wants to change P to P', then the sender A can compute E', S'_1, and S'_2 with P'. Therefore, E', S'_1, and S'_2 are known to A. With I, E, S_1, S_2, E', S'_1, and S'_2, the sender A can find I' using the equation (7) and can calculate $Sign_{pri}(I')$. The A creates and propagates a new transaction as shown in equation (8). A full node that received the transaction verifies $Sign_{pri}(I')$ and S'_2 with A's public key. If it is correct, the node finds the original transaction with the TN value and verifies the existing S_2 to check if it is the transactions created by the same user. If it is correct, the original transaction changes to the new transaction and the full node propagates the new transaction to other nodes in the blockchain.

$$Tx = TN|T|I'|Sign_{pri}(I')|A|E'|Meta|S_1'|S_2' \tag{8}$$

At this time, $H(T|A|I \oplus E \oplus S_1 \oplus S_2|Meta)$, that is the hash value included in the Merkle tree, is the same as $H(T|A|I' \oplus E' \oplus S_1' \oplus S_2'|Meta)$. This is because the equation (7) is satisfied.

This is the redactable encrypted transaction structure. a and a' in the equation (1) are I and I' in the equation (7) respectively. b and b' in the equation (1) are the ciphertexts E and E' in the equation (7) respectively. Also, c and c' in the equation (1) are the signatures S_1 and S_1' for the plaintexts P and P' in the equation (7) respectively. d and d' are the signature S_2 and S_2' for E and E' in the equation (7) respectively. Therefore, the equation (7) satisfies the equation (1).

This is a redactable encrypted transaction, so the personal information is encrypted in the transaction. Therefore, the first requirement is satisfied. Although Tx is changed, $H(T|A|B|I' \oplus E' \oplus S_1' \oplus S_2'|Meta)$ in the Merkle tree is the same as $H(T|A|B|I \oplus E \oplus S_1 \oplus S_2|Meta)$. It is because the equation (7) is satisfied. Therefore, there is no need to change the Merkle tree and thus there is no need to change the block. Therefore, the second requirement is satisfied. Moreover, it is changing at the transaction level(the third requirement), so there is no need for re-mining. It uses a simple XOR operation for changing the plaintext very efficiently without using a heavy cryptographic algorithm(the fourth requirement). Also, it does not require re-mining, so it can be applied to any consensus algorithm. Therefore, the proposed scheme is independent of the consensus algorithm and can be used for private blockchain or public blockchain(the fifth requirement). Finally, there is no need to keep the original contents after chaining the transaction (the sixth requirement).

If deletion is desired, P is filled with 0 bits.

3.4 Rationale

Here, we focus on the redactable encrypted transaction structure. However, the same rationale can be directly applied to redactable plaintext transactions.

This paper uses the correction value I. If the plaintext is changed or deleted by filling it with 0, the ciphertext and S_1 are also changed. Therefore, S_2 which verifies the ciphertext, and S_1 are also changed. Therefore, $E \oplus S_1 \oplus S_2$ is changed. By changing the correction value I by the changed amount, $I \oplus E \oplus S_1, \oplus S_2$ and the newly created $I‘ \oplus E' \oplus S_1' \oplus S_2'$ can be made equal. Assume that the difference between $E' \oplus S_1' \oplus S_2'$ and $E \oplus S_1 \oplus S_2$ is D'. $E \oplus S_1 \oplus S_2 \oplus D' = E' \oplus S_1' \oplus S_2'$. Let us that I' is $I \oplus D'$, then $I' \oplus E' \oplus S_1' \oplus S_2' = I \oplus D' \oplus E \oplus S_1 \oplus S_2 \oplus D' = I \oplus E \oplus S_1, \oplus S_2$. Therefore, the input value of the hash becomes the same.

This paper uses the signature S_1 of the plaintext because the ciphertext is stored in the blockchain and can be verified, but the plaintext cannot be known. Therefore, it cannot be confirmed whether the plaintext is correct or an attacker intercepted it and changed it. Therefore, the signature S_1 of the plaintext is provided to check the integrity of the plaintext. In addition, if the digital signature value is created whenever the sender sends the plaintext, it costs computation. Therefore, S_1 is stored in the blockchain ledger. Moreover, a big problem is that a malicious user sends another plaintext P'' instead of the plaintext stored in the blockchain, and sends the digital signature value of P''. Then the receiver verifies P'' instead of the plaintext stored in the blockchain. This significantly undermines the features that ensure the integrity of the blockchain. Therefore, the plaintext's digital signature value S_1 is stored in the blockchain ledger to ensure integrity.

This paper uses the digital signature value of the transaction, S_2. Of course, a digital signature value is included in the transaction in order to prove that the transaction was created by a legitimate user.

The signature value $sign_{pri}(I)$ of the correction value I was used. The correction value must be changed by only a legitimate user.

4 Security Analysis and Comparison

4.1 Security Analysis

From a security point of view, it is assumed that the adversary never knows the symmetric key used for encryption. Then the legitimate user will be safe. However, if the encryption key is exposed, of course, the plaintext can be recovered.

There is one subtle issue. The legitimate user who realizes that the symmetric encryption key has been exposed erases or changes the plaintext using a redactable blockchain. At this time, there is a problem of whether an attacker can recover the original plaintext. The attacker can recover the plaintext if they have the original encrypted transaction. However, if the attacker only has the changed transaction, the attacker will not be able to recover them.

In addition, what is the minimum information required to recover the original plaintext? If the attacker has the original ciphertext E, the attacker can decrypt it in any case. In the absence of the original ciphertext E, what information is needed to create the original ciphertext? If the attacker has I, S_1, and S_2, the attacker can recover E. It is because the changed I', S_1', S_2', and E' are stored in the blockchain. If the attacker knows I, the attacker can compute D' through $I \oplus I'$. Because the attacker knows D', the attacker can compute $D' \oplus E' \oplus S_1' \oplus S_2'$ and can get $E \oplus S_1 \oplus S_2$. Therefore, if I, S_1, and S_2 are known, E can be computed through XOR operation. Therefore, the attacker can obtain the original plaintext by decrypting E with the obtained symmetric encryption key.

In conclusion, if the attacker has the previous I, S_1, and S_2 and knows the encryption secret key, even if a legitimate user changes the plaintext, the attacker exploits a security vulnerability that can restore the original plaintext. Therefore, the encryption secret key must be kept securely. In our scheme, only the legitimate user can change the plaintext, because only the legitimate user has a signing key. Therefore, the attacker cannot change the transaction. Moreover, if the attacker does not have any of I, S_1, or S_2, our scheme has a security advantage in that the attacker cannot get the original plaintext.

4.2 Comparison

The table 2 shows the comparison of our scheme and existing schemes. [5] uses the cameleon-hash function which is a heavy cryptographic algorithm and only supports block-level redaction. [6] also uses a heavy cryptographic algorithm that is ciphertext-policy attribute-based encryption, but supports transaction-level modification. [7] has to re-mine, when the modification is required. Also, [7] is based-on PoW consensus algorithm and based-on public blockchain. [8] is based on proof-of-space consensus algorithm and this scheme requires that all transactions in a block originating from the same sender are modified. [9] supports only block-level modification. [10] works on the PoS consensus algorithm and does not support transaction-level modification. Also, if a block is changed, the re-mine is required. [11] works on the UTXO-based blockchain such as BitCoin. if the modification of the transaction, [11] requires re-mining. [12] requires saving the original content after modification. Our scheme does not require re-mining and a specific consensus algorithm and a specific type of blockchain. Also, the proposed scheme supports transaction-level modification and does not use a heavy cryptographic algorithm. Moreover, the proposed scheme does not require saving the original content after modification.

Table 2: Comparison

	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	our scheme
no need re-mining	O	O	X	O	O	X	X	O	O
supporting the transaction-level modification	X	O	O	X	X	X	O	O	O
no need heavy cryptographic algorithm	X	X	O	O	O	O	O	O	O
no need specific consensus algorithm	O	O	X	X	O	X	O	O	O
no need a specific type of blockchain	O	O	X	O	O	O	X	O	O
no need to save the original content	O	O	O	O	O	O	O	X	O

5 Discussion

The private key can be split using a well-known technique such as Shamir Secret Sharing [18]. For a 2-of-3 threshold scheme [19], one key piece is stored by the user, and the other key pieces are kept by a trusted third party. Suppose only when the DAO (Decentralized Autonomous Organization) [20] approves the change, a transaction can be changed by joining the signature using a key piece. Through this, the information recorded in the blockchain will be able to be changed only when it is justified.

6 Conclusion

In this paper, we proposed a universal redactable blockchain scheme that can be used regardless of the consensus algorithms without using a heavy cryptographic algorithm. This scheme can be applied to both private and public blockchains. The proposed scheme can be modified at the transaction level without changing the block and the chain of blocks by creating a hash value identical to the hash value recorded in the Merkle tree only with a simple XOR operation using the correction value I. Therefore, the computation is very simple. Moreover, the proposed scheme can be changed to fine-grain. In addition, we looked at it in terms of security, and it is a scheme that can be used safely if the user meets the conditions for safely storing his/her symmetric encryption key. Although further research is needed in the future, the Multi-Sign technique or threshold cryptography can be applied to prevent unauthorized changes to the contents stored in the blockchain.

Acknowledgments

This paper was supported by the Konyang University Research Fund in the second half of 2021

References

[1] S. Nakamoto. A peer-to-peer electronic cash system, 2008. `https://bitcoin.org/bitcoin.pdf` [Online; Accessed on December 15, 2022].

[2] P. Voigt and A.V.d. Bussche. The eu general data protection regulation (gdpr), 2017. `https://gdpr-info.eu/` [Online; Accessed on December 15, 2022].

[3] V. Dhillon, D. Metcalf, and M. Hooper. The dao hacked. In *Blockchain Enabled Applications*, pages 67–78. Apress Berkeley, 2017.

[4] D. Sartori. Redactable blockchain: how to change the immutable and the consequences of doing so. Master's thesis, University of Twente, 2020.

[5] G. Ateniese, B. Magri, D. Venturi, and E. Andrade. Redactable blockchain–or–rewriting history in bitcoin and friends. In *Proc. of the 2nd IEEE European symposium on security and privacy (EuroS&P'17), Paris, France*, pages 111–126. IEEE, April 2017.

[6] D. Derler, K. Samelin, D. Slamanig, and C. Striecks. Fine-grained and controlled rewriting in blockchains: Chameleon-hashing gone attribute-based. In *Proc of the 2019 Network and Distributed Systems Security Symposium (NDSS'19), San Diego, CA, USA*, pages 1–15. NDSS, February 2019.

[7] N. Lee, J. Yang, M.M.H. Onik, and C. Kim. Modifiable public blockchains using truncated hashing and sidechains. *IEEE Access*, 7:173571–173582, 2019.

[8] X. Cai, Y. Ren, and X. Zhang. Privacy-protected deletable blockchain. *IEEE Access*, 8:6060–6070, 2019.

[9] D. Deuber, B. Magri, and S.A.K. Thyagarajan. Redactable blockchain in the permissionless setting. In *Proc. of the 40th IEEE Symposium on Security and Privacy (SP'19), San Francisco, USA*, pages 124–138. IEEE, May 2019.

[10] J. Xu, X. Li, L. Yin, B. Guo, H. Feng, and Z. Zhang. Redactable proof-of-stake blockchain with fast confirmation. *Cryptology ePrint Archive*, 2019(1110), 2019.

[11] M. Florian, S. Henningsen, S. Beaucamp, and B. Scheuermann. Erasing data from blockchain nodes. In *Proc. of 4th IEEE European Symposium on Security and Privacy Workshops (EuroS&PW'19), Stockholm, Sweden*, pages 367–376. IEEE, June 2019.

[12] S.A.K. Thyagarajan, A. Bhat, B. Magri, D. Tschudi, and A. Kate. Reparo: Publicly verifiable layer to repair blockchains. In *Proc. of 25th International Conference on Financial Cryptography and Data Security (FC'21), Virtual*, volume 12675 of *Lecture Notes in Computer Science*, pages 37–56. Springer, March 2021.

[13] H. Krawczyk and T. Rabin. Chameleon hashing and signatures. *Cryptology ePrint Archive*, 1998, March 1998.

[14] P.S.L.M. Barreto, H.Y. Kim, B. Lynn, and M. Scott. Efficient algorithms for pairing-based cryptosystems. In *Proc. of 22nd Annual international cryptology conference (CRYPTO'02), Santa Barbara, USA*, volume 2442 of *Lecture Notes in Computer Science*, pages 354–369. Springer, August 2002.

[15] M. Szydlo. Merkle tree traversal in log space and time. In *International Conference on the Theory and Applications of Cryptographic Techniques(EUROCRYPT'04), Interlaken, Switzerland*, volume 3027 of *Lecture Notes in Computer Science*, pages 541–554. Springer, May 2004.

[16] D. Johnson, A. Menezes, and S. Vanstone. The elliptic curve digital signature algorithm (ecdsa). *International journal of information security*, 1:36–63, 2001.

[17] J. Daemen and V. Rijmen. Aes proposal: Rijndael, 1999. `http://www.cs.cmu.edu/afs/cs/project/pscico-guyb/realworld/www/docs/rijndael.pdf` [Online; Accessed on December 15, 2022].

[18] A. Shamir. How to share a secret. *Communications of the ACM*, 22:612–613, 1979.

[19] Y.G. Desmedt. Threshold cryptography. *European Transactions on Telecommunications*, 5:449–458, 1994.

[20] C. Jentzsch. Decentralized autonomous organization to automate governance, November 2016. https://lawofthelevel.lexblogplatformthree.com/wp-content/uploads/sites/187/2017/07/WhitePaper-1.pdf [Online; Accessed on December 15, 2022].

Author Biography

Seung Wook Jung received B.S. and M.S. degrees in Electronic Engineering from Soongsil University in 1998 and 2000, and Ph.D. degrees in the University of Siegen in 2005. Currently, he is an assistant professor at Konyang University. His research interests include Blockchain, Applied Cryptography, Privacy, Network Security, and Cloud Computing Security.