Hindawi
Wireless Communications and Mobile Computing
Volume 2019, Article ID 6219868, 13 pages
https://doi.org/10.1155/2019/6219868

Research Article

Toward Privacy-Preserving Shared Storage in Untrusted Blockchain P2P Networks

Sandi Rahmadika [1] **and Kyung-Hyune Rhee** [2]

[1] *Interdisciplinary Program of Information Security, Graduate School, Pukyong National University, Republic of Korea*
[2] *Department of IT Convergence and Application Engineering, Pukyong National University, Republic of Korea*

Correspondence should be addressed to Kyung-Hyune Rhee; khrhee@pknu.ac.kr

Received 18 January 2019; Revised 9 March 2019; Accepted 2 April 2019; Published 16 May 2019

Academic Editor: Ilsun You

The shared storage is essential in the decentralized system. A straightforward storage model with guaranteed privacy protection on the peer-to-peer network is a challenge in the blockchain technology. The decentralized storage system should provide the privacy for the parties since it contains numerous data that are sensitive and dangerous if misused by maliciously. In this paper, we present a model for shared storage on a blockchain network which allows the authorized parties to access the data on storage without having to reveal their identity. Ring signatures combined with several protocols are implemented to disguise the signer identity thereby the observer is unlikely to determine the identity of the parties. We apply our proposed scheme in the healthcare domain, namely, decentralized personal health information (PHI). In addition, we present a dilemma to improve performance in a decentralized system.

1. Introduction

Since being introduced to the public through the rise of Bitcoin, blockchain has attracted a lot of attention among researchers, especially the way it deals in a transaction without involving the third parties. The blockchain technology reduces the transaction costs and it improves the efficiency and reliability of the decentralized system in general [1]. Due to its merits, blockchain has been developed in various fields of study such as logistics, e-commerce, trading activity, and healthcare, to name a few. Blockchain in the healthcare area is growing rapidly [2] as a future trend of substantial impact [3]. It aims at improving the quality of service and maintaining the integrity of information. Blockchain must be mature enough in all aspects, especially in security matters before blockchain being applied to a sensitive system (healthcare domain) [4] since it consists of valuable data for patients, providers, and all parties involved.

The shared storage between healthcare providers and the patients is one of the factors that must be considered when determining the scheme of the decentralized healthcare system. The surveys indicate that users often do not fully trust to store their data to third parties [5]. There are decentralized storage providers that provide the alternative services to protect the user's privacy such as Freenet [6] and GNUnet [7]. However, those services still have some drawbacks such as free-rider problem. More precisely, the provider is less motivated to keep improving system reliability due to the fact that there is no significant benefit for the provider to preserve the users' data. Apart from the free-rider problem, the main issue in the decentralized shared storage is related to the privacy of users [8]. An observer may be able to see the contents of online activities or metadata of the user since the data is publicly available.

To deal with the issues, we propose a model of the decentralized shared storage system on the blockchain that provides the privacy of the users without the involvement of third parties. Ring signature algorithm is applied to disguise the original identity of the signer. The parties involved use signatures on behalf of a group; hence, the original identity of the signer is unknown called signer ambiguous. In order

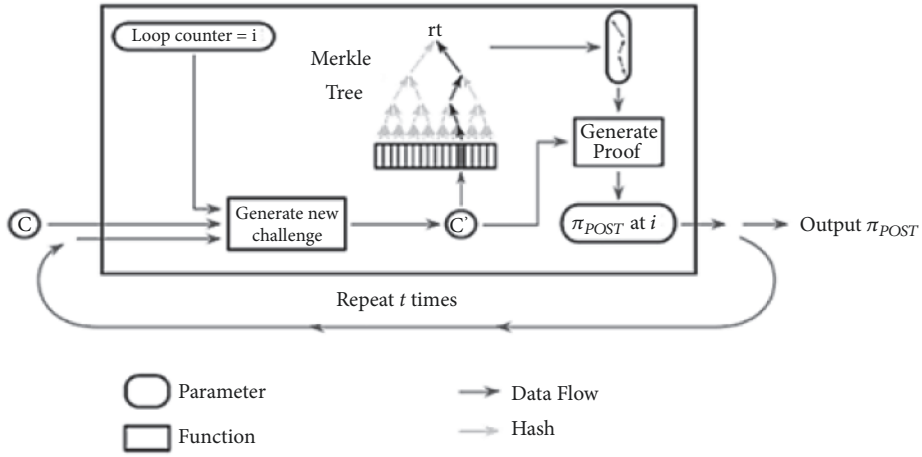

Figure 1: The iterative proof of Filecoin, adapted from [9].

to keep the identity of the parties to remain untraceable, one-time use address (from the stealth address) is adapted so that the observer cannot link the user address based on the transaction that has been carried out in advance.

The predecessor approaches to design the sharing storage system in the blockchain have been started by researchers lately such as Storj [10], storage with financial incentives, and Filecoin (see Figure 1) which generates a proof-of-spacetime (PoST) for the replica [11]. This paper presents the key concepts in the decentralized sharing storage in the healthcare system. The personal health information of the patient is propagated in the peer-to-peer blockchain network and the data are stored in a storage provider. The model of decentralized healthcare system comes from our previous research [12]. The privacy-preserving for the user is beyond the topic at the time. This paper is ongoing research and interrelated with our previous research.

The structure of the paper is organized as follows. Section 2 describes the background and core system component such as the ring signature, CryptoNote, and one-time use address (stealth address). Section 3 presents the system model of decentralized PHI data as well as the concept of ring confidential transaction. Section 4 presents the system analysis including the dilemma of reparameterizing propagation time and block size in order to improve the performance in a transaction. The limitations and future work are written in Section 5. Finally, Section 6 concludes the paper.

2. Background

In this section, we briefly present the essential information of ring signature algorithm, CryptoNote protocol, one-time use transaction address, and stealth address which are basic components for the privacy-preserving model in our system.

2.1. The Essential of Ring Signature. Ring signature is first introduced by Rivest et al. [13] in 2001 through the paper entitled "How to Leak a Secret". The idea of a ring signature originates from the concept signature group proposed by Chaum et al. [14] which in the group signature each member agrees to sign the message. In short, the data is signed on behalf of the group. In the group signature, there is a manager who organizes each activity in a group. As opposed to the group signature, the ring algorithm does not possess a manager in the process and neither have special requirements for creating groups as shown in Figure 2.

In order to form a signature group, the signer requires public keys P_k knowledge from prospective members. The selected public keys are encrypted by using a trapdoor permutation function (RSA, Rabin, and Diffie-Hellman). Due to the nature of the ring signature protocol, there are no specific rules for the number of members in a group. The standard procedure of the ring signature protocol can be defined as follows:

(i) $\boldsymbol{Sign}\,\boldsymbol{\sigma}(msg, P_{sn}, P_{k1}, P_{k2}, P_{k3}, \ldots, P_{kn})$. The signature consists of the public keys $(P_{sn}, P_{k1}, P_{k2}, P_{k3}, \ldots, P_{kn})$ of the members for every message msg concatenated with the secret key P_{sn} of the signer to produce a signature σ.

(ii) $\boldsymbol{Verify}(\boldsymbol{msg},\boldsymbol{\sigma})$. The verification process can be interpreted as accepting a group signature σ which consists of public keys of all the possible signers along with the message msg. The final output is $true$ or $false$.

Generating a ring signature can be used directly by the signer without involving the group manager. The initial step is the signer computes the symmetric key Sym_k as the hash value of the message msg to be signed as $Sym_k = h(msg)$. The more complicated variant generates $Symk$ as $h(msg, P_{k1}, P_{k2}, P_{k3}, \ldots, P_r)$. However, the simpler creation is also secure. An initial random value R_v (or "glue") is chosen by the signer uniformly at random form $\{0,1\}^b$, where 2^b is some power of two which is larger than all modulo $n_i's$. Furthermore, the signer selects the number of signatures x_i from the ring members $1 < i < r, i \neq s$, where r is the ring members and s is the order

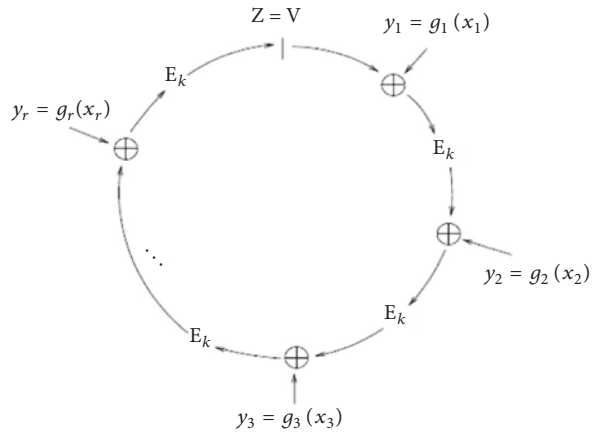

Figure 2: Ring signature algorithm which is defined by any member of a group of parties each having keys.

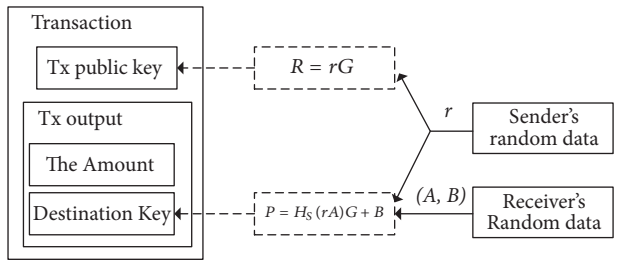

Figure 3: The structure of CryptoNote standard transaction.

of the member (s-th member) who is the actual signer. Hence, the signature gets a new value which is signified by $y_i = g(x_i)$. Finally, the signature of the message msg can be defined as $(P_{k1}, P_{k2}, P_{k3}, \ldots, P_{kn}; R_v; x_1, x_2, \ldots, x_r)$. The verification process is straightforward by describing the message received from the sender via secure channel $(P_{k1}, P_{k2}, P_{k3}, \ldots, P_{kn}; R_v; x_1, x_2, \ldots, x_r)$.

2.2. CryptoNote Protocol. The use of the ring signature algorithm in blockchain transaction was first introduced in 2012 which is part of the CryptoNote protocol [15] and updated in 2013. The CryptoNote constructs the ring signature using the public key of the random addresses and it provides privacy for the patient by leveraging the stealth address protocol [16]. By doing so, the observer believes that the signer has a secret key that corresponds to the cryptocurrencies, but the observer cannot determine the specific identity of the sender. However, the original ring signature protocol would allow double-spending attack in the blockchain transaction. The original ring signature protocol cannot determine the origin of the coin due to the fact that there is no marker if the transaction has been sent to the recipient. As a solution, CryptoNote provides one-time use ring signatures and key image as a marker.

The key image acts as a unique marker for every transaction. The key image gives the information about the transaction with a particular signature σ_n. If the same signature is used more than once, the miners will reject the transaction. In other words, any attempt to double-spend will indicate the use of the same key image. The destination of each CryptoNote output is a public key $(P_{k1}, P_{k2}, P_{k3}, \ldots, P_{kn})$ which is derived from the recipient's address combined with the sender's random value. In this regard, the sender asks the recipient's public key (A, B) via secure channel and the sender generates the one-time public key $P = H_s(rA)G + B$ as can be seen in Figure 3. Based on the key image that recipient belongs to, the recipient checks every passing transaction using his/her secret key (a, b) and calculates $P' = H_s(aR)G + B$, where H_s is a cryptographic hash function $\{0, 1\}*$ and G is a base point. Finally, the recipient can define $aR = arG = rA$ and $P' = P$. In addition, the recipient can recover the corresponding one-time secret key $x = H_s(aR) + b; P = xG$. By signing x transaction, the recipient can spend this output at any time. The security from this protocol is an untraceable transaction for the observer since the incoming message received by a recipient associated with one-time public keys (unlinkable).

2.3. One-Time Use Transaction and Stealth Address. We first present the one-time use transaction in general and we briefly describe the drawbacks of one-time use address in the blockchain transaction. To address the problem, we use the stealth address to protect the recipient information in our system model.

2.3.1. One-Time Use Transaction. Using the same address for each transaction on the blockchain network allows observers to track transactions to the original sender even though the

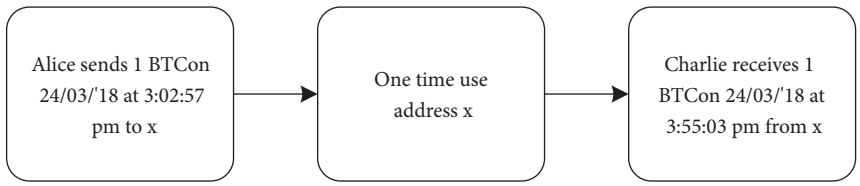

Figure 4: One-time use transaction.

address is in the pseudonymous form. Since the new address does not have a track record in the blockchain metadata, the observer is unlikely to track information from a transaction. However, one-time use transaction address provides privacy for the users by extending the address to the counter-parties which detail information of the transaction still publicly available on the blockchain network. Roughly speaking, the sender enables the recipient to spend the funds that he just received even though the recipient identity remains hidden as can be seen in Figure 4.

For instance, an address X is a one-time address, but the observer has knowledge that Alice sent 1BTC to X and Charlie received from X. It can be used by the observer to infer that X corresponds to Alice and Charlie. By gathering the details on where Charlie's fund originated and where Alice's funds were passed on to, it could avail deanonymize the one-time use transaction. This scheme is also called *transaction graph analysis*. Therefore, in order to develop privacy on the decentralized blockchain system, a new protocol is necessary such as stealth address protocol. In other words, it is a security protocol that has become ubiquitous almost by stealth [17].

2.3.2. Stealth Address. One-time use address for transaction might be inconvenient to manage since the new address must be generated by the recipient for every transaction that will be carried out. Unlike the one-time use payment address, the stealth addresses get rid of this requirement. In short, stealth addresses can be generated as follows:

(i) The recipient generates a parent key pair $Pub(A, B)$ and publishes his/her public key (the published key is called *stealth address*).

(ii) The sender will be able to use the stealth address of the recipient to commit a new one-time use payment address for a particular transaction.

(iii) At this stage, the recipient uses their parent private key $P_s(a, b)$ which is generated in advance to spend the funds received. The generating process of stealth address can be seen in Figure 5.

The addresses are generated using trapdoor permutation such as Diffie-Hellman key exchange protocol [18]. By leveraging the stealth address protocol, the system allows eliminating the possibility for observers to link a one-time address to another. In this sense, the observer cannot determine the recipient's address and is unlikely to link the transaction to the recipient due to the new address being generated based on his/her stealth address for every transaction. Stealth address protocol is used in CryptoNote and Zcash combined with various other techniques such as zk-SNARKs in order to provide the stronger privacy for the user in the decentralized blockchain network. The cryptographic approaches can prevent ad hoc networks against external attackers using node authentication and data encryption [19].

3. Our System Model

In this section, we elaborate the model of decentralized PHI data (the model is from our prior work), the sequences of the standard blockchain transaction, the group of ring signature, and the ring confidential transaction of PHI.

3.1. Decentralized PHI Data. A decentralized personal health information model originally came from our previous research [12]. In our predecessor work, blockchain technology is used to manage the personal health information (PHI) data of the patient which obtain from several healthcare providers as can be seen in Figure 6. Based on the decentralized PHI model, we conducted testing in order to find out information about data communication in peer-to-peer networks (see Figure 7). The results obtained show the high level of success in sending data among the parties in the blockchain network that reaches 100% and the average of propagation message is 1:18ms for 100 bytes of Internet Control Message Protocol Echo. By leveraging the model of our previous research, the patients and the healthcare providers allow to collect effectively the PHI data onto a single view with integrity guarantee. Data integrity is essential for the patient in the blockchain network since it is a fundamental component of information security which verifies that the data has remained unaltered in transit from creation and reception [20]. By design, blockchain is tamper-proof, immutability (the data stored are unchanging over time or unable to be changed) so it is suitable for managing sensitive data such as personal health information, digital medical record, and other similar data.

In the decentralized healthcare system model, the patient and the providers are on the same blockchain network. The healthcare providers preserve a diagnosis from the patient and then store it into sharing storage right after the data is confirmed by the miner. The patient afterwards enables to find the data stored by the provider in the storage by searching one by one based on the patient's public key $Pub(A, B)$ attached into the transaction. The patient whose public key is attached to the transaction is the only party who knows the data stored in the storage because he/she has knowledge of the secret key $P_s(a, b)$ to access the PHI data.

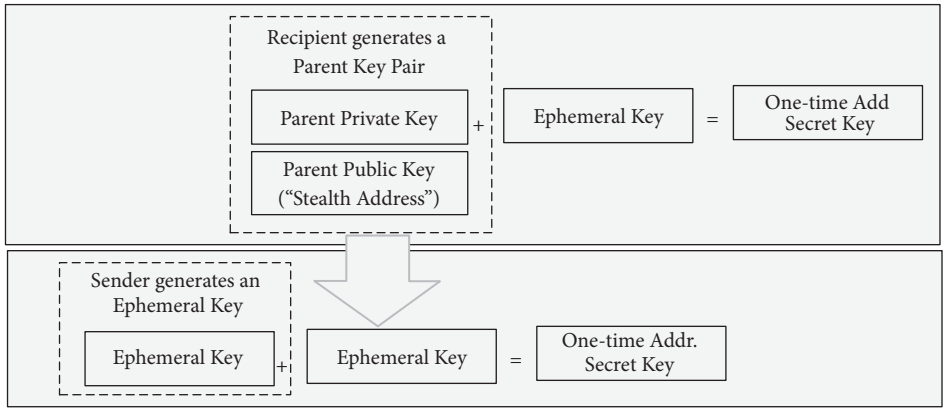

Figure 5: Generating process of stealth address.

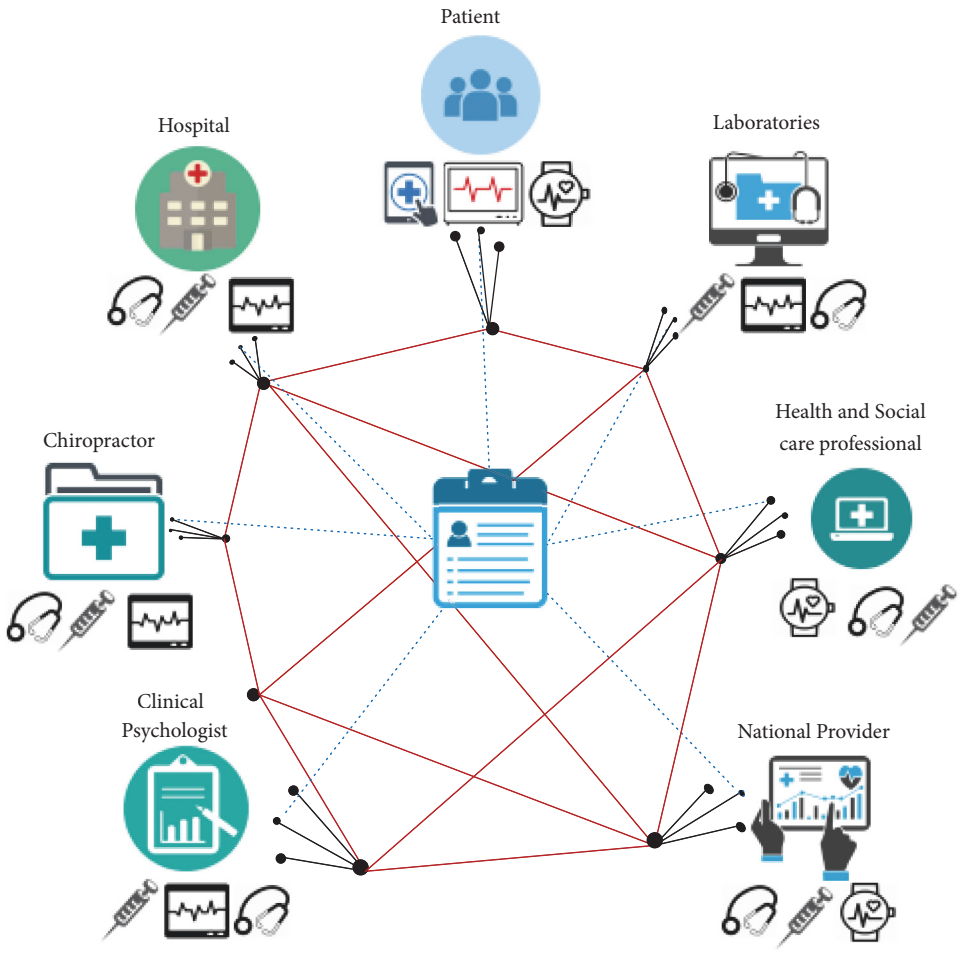

Figure 6: The model of decentralized personal health information data.

As with the security model in general, every party in the system has the unique parent keys. The public key is used to commit transactions, which later will be used to generate a stealth address for every transaction. A pair of keys is obtained from trapdoor permutation functions such as RSA algorithm and Rabin [21] which are generated beforehand. The parent keys of the parties can be defined as follows:

(i) The patient (Pub_α, Pr_α), the pair of patient's public key to commit the transaction can be defined as (A_α, B_α) and the pair of secret keys (a_α, b_α).

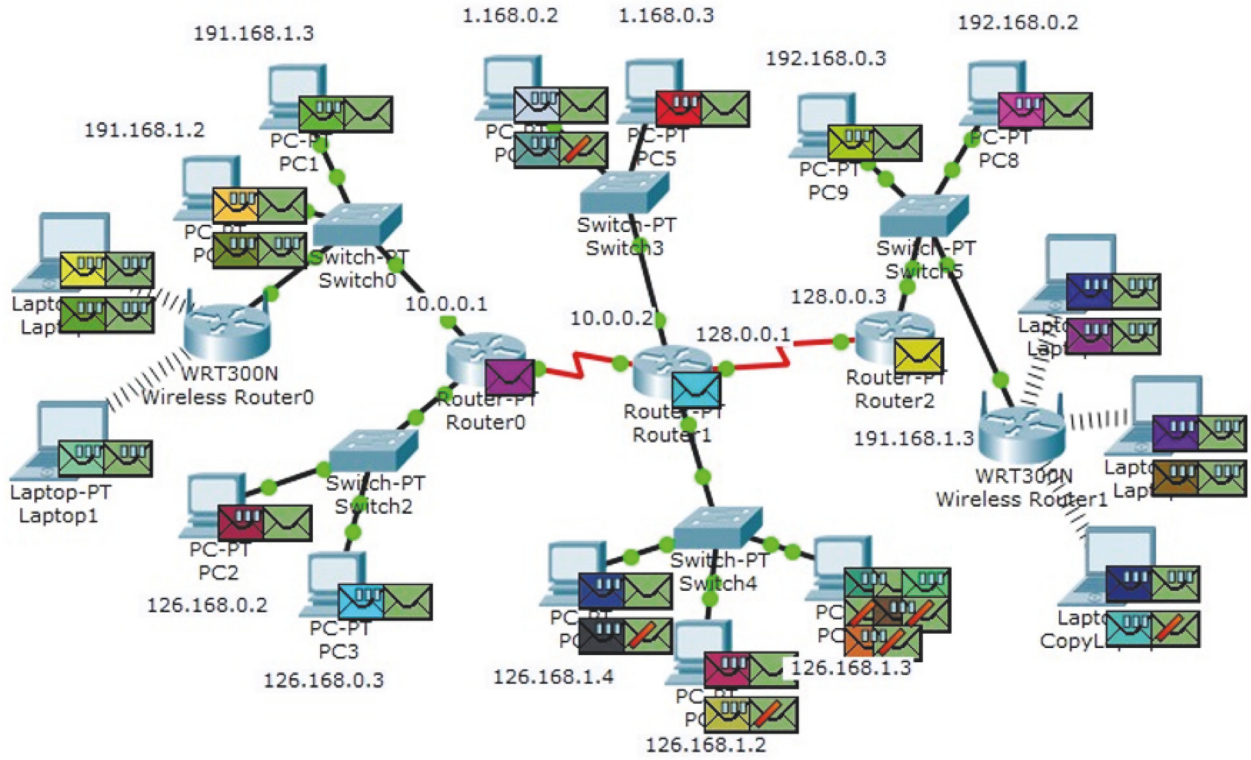

Figure 7: Propagating the personal health information data to the entire nodes.

(ii) Hospital (Pub_β, Pr_β), the pair of hospital's public key (A_β, B_β) and the pair of secret keys(a_β, b_β).

(iii) Chiropractor (Pub_γ, Pr_γ), Chiropractor's public key (A_γ, B_γ) and the pair of secret keys(a_γ, b_γ).

(iv) Clinical psychologist (Pub_δ, Pr_δ), clinical psychologist's public key (A_δ, B_δ) and the pair of secret keys (a_δ, b_δ).

(v) National provider $(Pub_\varepsilon, Pr_\varepsilon)$, the public key of national provider $(A_\varepsilon, B_\varepsilon)$ and the pair of secret keys $(a_\varepsilon, b_\varepsilon)$.

(vi) Health and social care provider (Pub_ζ, Pr_ζ), health and social provider's public key (A_ζ, B_ζ) and the pair of secret keys (a_ζ, b_ζ).

(vii) Laboratories(Pub_η, Pr_η), the pair of public keys of laboratories can be defined (A_η, B_η) and the pair of secret keys (a_η, b_η).

$$P = H_s(rA_\alpha)G + B_\alpha \tag{1}$$

$$P' = H_s(a_\alpha R)G + B_\alpha, then \tag{2}$$

$$a_\alpha R = a_\alpha rG = rA_\alpha; \quad P' = P \tag{3}$$

The sequence of a standard transaction in decentralized PHI data starts from the provider who wants to store the patient's data in the storage where the patient has published his address to the provider beforehand. The provider unpacks the address and gets the patient's public key (A_α, B_α). The provider picks a random $r \in [1, l-1]$: where l is a prime order of the base point G. The provider then generates a one-time public key based on the pair public key (1) of the patient (the process is signified by Figure 3).

The patient and the healthcare providers possess a pair of public keys (A_n, B_n) for different purposes. The first public key A_n is used to generate a one-time public key, whilst the public key B_n is attached to the transaction as the tracking value used by the patient to find the data addressed to him/her. The provider uses P as a destination key for the output and attaches the new value $R = rG$ into the transaction. The PHI data with attachments to P and R values are stored into shared storage after being validated by the miner. The patient later checks every transaction using his private key (a_α, b_α) and calculates the new value P' (2). Finally, the patient compares the value P received with the value P' decrypted (3).

3.2. The Group of Ring Signature. In the decentralized PHI system, there is a group ring signature consisting of patient and several healthcare providers. In order to generate a group, the system does not demand special requirements and also unlikely require a manager to manage the group. All that is needed to create a ring signature group is the public key of each party $(Pub_\alpha, Pub_\beta, Pub_\gamma, Pub_\delta, Pub_\varepsilon, Pub_\zeta$, and $Pub_\eta)$.

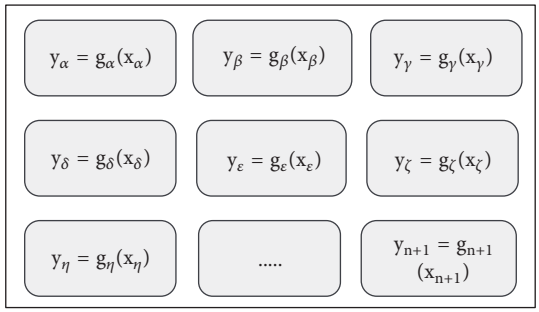

Figure 8: The group of ring signature which is derived from the public keys of the member.

Once a ring signature group has been generated, all members of the group are allowed to use the signature and combine it with the private key of the sender. It grants users fine-grained control over the level of anonymity associated with a certain signature [22].

The signer enables to choose the number of signatures that they want to use in the transaction to provide an ambiguous signer. Later, the public key is used for encryption $y_n = g_n(x_n)$ as shown in Figure 8. Intuitively, g_n is defined by the extended trapdoor permutation function such as RSA and Rabin algorithm. In practice, for Rabin's functions $g_n(x_n)$ extends to $f_i(x_i) = {x_i}^2 \bmod n_i$ over $\{0,1\}^b$: where 2^b is the power of two which is larger than all modulo ${n_i}'s$. The bucket consists of b-bit numbers $w = q_i n_i + r_i$: where $r_i \in \mathbf{Z}^*_{n_i}$ and $(q_i + 1)n_i \leq 2^b$. For any b-bit numbers w is nonnegative integers q_i and r_i. The bucket values are mapped by the extended Rabin mapping g_n. So, the value of $g_i(w)$ is signified by (4).

$$g_i(w) = \begin{cases} q_i n_i + f_i r_i & \text{if } (q_i + 1)\, n_i \leq 2^b \\ w & \text{else} \end{cases} \quad (4)$$

Intuitively, $g_i(w)$ is a permutation function over $\{0,1\}^b$ which is also a one-way trapdoor function because only a person knows the inverted value of f_i for a given input. Therefore, we can define the public keys for the prospective members as follows:

(i) The patient $\longleftarrow y_\alpha = g_\alpha(x_\alpha)$.

(ii) Hospital $\longleftarrow y_\beta = g_\beta(x_\beta)$: Chiropractor $\longleftarrow y_\gamma = g_\gamma(x_\gamma)$; whilst, the clinical psychologist $\longleftarrow y_\delta = g_\delta(x_\delta)$.

(iii) National provider $\longleftarrow y_\varepsilon = g_\varepsilon(x_\varepsilon)$: health and social care provider $\longleftarrow y_\zeta = g_\zeta(x_\zeta)$; Laboratories $\longleftarrow y_\eta = g_\eta(x_\eta)$.

(iv) For additional members of a group, it can be generated at any time as long as the public key of the new member is known $y_{\{n+1\}} = g_{\{n+1\}}(x_{\{n+1\}})$;

$$R_{sg} = y_\alpha \oplus y_\beta \oplus y_\gamma \oplus y_\delta \oplus y_\varepsilon \oplus y_\zeta \oplus y_\eta \oplus \ldots \oplus y_{\{n+1\}} \quad (5)$$

As can be seen in (5), a group of ring signature is constructed based on encryption of the member's public key. By design, the sender signs the message for the individual transaction using the signature of the group without a single group manager involved. Because the sender uses the signature of the group, the observer cannot judge the identity of the real sender for the corresponding transaction. The sender enables to choose the number of signatures that he/she wants to use in the transaction. The signature of a group can be used at any time in a transaction without having permission from the owner of the key. For instance, in a particular transaction the provider y_β uses the following keys to sign a message $h(msg, y_\gamma, y_\delta, y_\alpha$ and his key $y_\beta)$.

3.3. Ring Confidential Transaction of PHI. Ring confidential transaction (RingCT) is used in Monero cryptocurrency [16] in order to improve the privacy of the users. Intuitively, the ring confidential transaction aims to hide the value of the actual amount in a transaction by combining the value of the current transaction with the value of the predecessor transactions. By doing so, the observer cannot tell the exact value of a transaction. The value of the transaction can be the number of coins sent or other data depending on the type of system that applies RingCT [23]. Unlike in the Monero transaction, the value transaction in the Bitcoin is publicly available in the plaintext. The observer might be able to analyse the transaction values for certain purposes in the particular period of time. Therefore, public data will be dangerous if misused maliciously.

$$RCT = txid\,[(8) \parallel (2) \parallel (5) \parallel (11) \parallel (12) \parallel (15)] \quad (6)$$

Suppose all outputs in Figure 9 exist, whilst the transaction that provider enables to spend is highlighted in green. Whenever the provider creates a transaction, he uses a RingCT to disguise which input is actually being spent. In this regard, the provider combines the current transaction with the previous transactions (highlighted in gray). The confidential ring signature for the transaction is shown in (6). The provider allows using the RingCT directly without permission and node manager involvement. As well as the group signature, the sender is free to use the value of previous transactions in order to disguise the value of the current transaction. By leveraging the model, it is possible to create privacy-preserving for users in the decentralized peer-to-peer shared storage in healthcare area with the following main objectives:

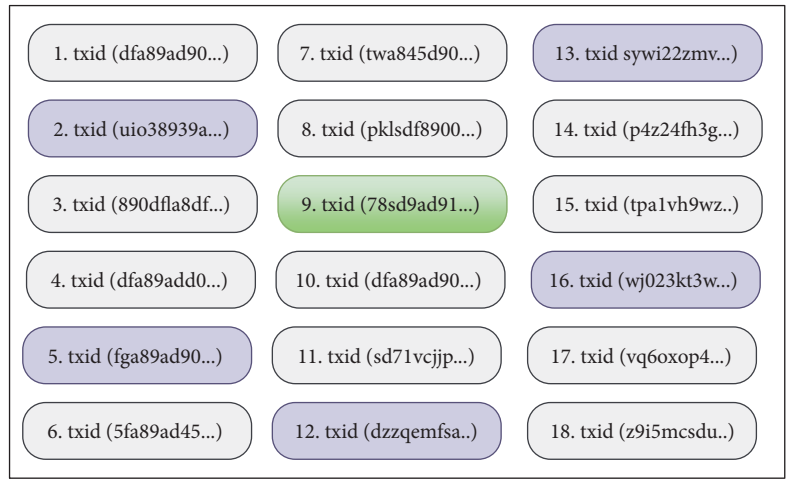

Figure 9: Ring confidential transaction based on the prior transactions.

(i) **Untraceability** with the goal to protect the sender's information in the form of the address *(public key)*. The observer cannot trace where the coin was received and where the coin originated from.

(ii) **Unlinkability** to preserve the recipient's identity. This can be achieved by using a stealth address where the recipient makes two public keys that are used to create a one-time address and the other as the view key.

(iii) **Confidential Values** to disguise the value of a transaction. The value of the current transaction is combined with the values of the transactions that have been carried out previously so it becomes an obscured transaction.

4. Systems Analysis and the Dilemma

The main protocols for building privacy-preserving for blockchain shared storage model have been discussed in previous section. In this section, we present the relation between each protocol. The combination of the protocols provides a system that enables protecting the privacy of users in the decentralized shared storage. The procedures are displayed gingerly, starting from generating the members of ring signature through to mechanism of storing data in the blockchain shared storage. At the end of this section, we elaborate on some dilemmas.

We demonstrate a case study in which one of the providers (hospital) wants to store diagnostic data of the patient into the decentralized shared storage. The provider has known the address *(public key)* of the patient beforehand. In this sense, the hospital acts as the sender whilst the patient acts as the recipient of the PHI message. The provider attaches the address of the patient in the form of a *view key* for each transaction so that only patients likely enable to track data stored in shared storage using his/her knowledge. The miners have a duty to validate the data from the sender before being saved into shared storage, yet they are responsible for adding blocks to the blockchain network. The miner gets a reward after successfully adding the new block as with the blockchain system in general. Algorithm 1 is a description of the whole system process starting with making public keys through to data stored in shared storage. The process sequence for privacy-preserving shared storage in untrusted blockchain P2P networks as follows:

(i) The party possesses a pair of parent keys (Pub_{key}, Pr_{key}) that have been generated beforehand based on trapdoor permutation function such as RSA, Rabin, or Diffie-Hellman algorithm. The patient is the recipient($Pub_{\alpha}, Pr_{\alpha}$), whilst the hospital is the sender (Pub_{β}, Pr_{β}) in this case study.

(ii) The hospital is the sender of the patient's diagnosis data (in this case). The hospital constructs a ring signature group based on public key from providers and patient that has been known previously. The hospital also added its public key to the group.

$$R_{sg\beta} = g_{\alpha}(x_{\alpha}) \oplus g_{\boldsymbol{\alpha}}(x_{\beta}) \oplus g_{\boldsymbol{\gamma}}(x_{\boldsymbol{\gamma}}) \oplus g_{\delta}(x_{\delta}) \oplus g_{\varepsilon}(x_{\varepsilon}) \oplus g_{\zeta}(x_{\zeta}) \oplus g_{\eta}(x_{\eta}) \oplus \ldots \oplus g_{n+1}(x_{n+1}) \tag{7}$$

(iii) When the group of ring signature $R_{sg\beta}$ is generated, the sender enables to use the signature of each group member without having to get permission from the owner of public keys. In this case study, the hospital chooses to use all signatures from group members as shown in (7). The hospital can add new members at any time as long as the public key is known.

(iv) The patient as the recipient later makes the stealth address based on a pair of parent keys. The patient creates a pair of public keys $Pub_{\alpha}(A_{\alpha}, B_{\alpha})$ and sends it to the hospital via secure channel. The hospital generates a new one-time address based on stealth

1: **Procedure** *Shared_Storage*:
2: **Trapdoor Function: (parent keys)** ∗*trapdoor permutation function e.g. RSA, Rabin*
3: $Patient \longleftarrow hash(Pub_\alpha, Pr_\alpha)$
4: $Hospital \longleftarrow hash(Pub_\beta, Pr_\beta)$ ∗*∀party has parent keys*
5: $Chiropractor \longleftarrow hash(Pub_\gamma, Pr_\gamma)$
6: $Clc.Psychologist \longleftarrow hash(Pub_\delta, Pr_\delta)$
7: $NationalProvider \longleftarrow hash(Pub_\varepsilon, Pr_\varepsilon)$
8: $Social\ care \longleftarrow hash(Pub_\zeta, Pr_\zeta)$
9: $Laboratories \longleftarrow hash(Pub_\eta, Pr_\eta)$
10: **The sender creates a group of ring signature:**
11: **Procedure** *Use Public Key* $\forall$ $parties \in parentkeys$ ∗*use public key of the parties as an input*
12: **Create RS:**
13: $R_{sgn} \longleftarrow g_\alpha(x_\alpha) \oplus g_\beta(x_\beta) \oplus g_\gamma(x_\gamma) \oplus \ldots \oplus g_{n+1}(x_{n+1})$
14: **AddNewMembers:** ∗*in case: add new members*
15: $\boldsymbol{Update}\ R_{sgn} \longleftarrow Get\_NewPubkey \oplus PubKey_{(n+1)}$
16: **end procedure**
17: **Procedure** *Create StealthAddr*(*byRecipient*): ∗*the recipient creates stealth address*
18: $ParentKey$: $PubKey_n \longrightarrow PubKey_n(A_n, B_n)$
19: $Send(A_n) \longrightarrow to\_sender\_(via\_SecureChannel)$
20: $(A_n, B_n)received \longrightarrow Create\_New\_OTP$ ∗*sender creates new OTP for recipient*
21: **end procedure**
22: **Procedure** *Confidential Ring Signature*: ∗*confidential ring signature as an option*
23: *if* $\forall txs$ *BC use CRS* ***then return*** True
24: $NewValue \longleftarrow CurrentValue \oplus Prev.Value$
25: *Include to the new tx* $\longleftarrow$ ***NewValue***
26: ***elsereturn*** False
27: **end procedure**
28: **Procedure** *Sign the PHI Data*(*Msg*): ∗*in this case, the PHI data is from the hospital*
29: ***Get*** $Msg \in Diagnosis\_Data(Hospital)$
30: ***Sign*** $\boldsymbol{\sigma} \longleftarrow choose\_signer \in R_{sgn}$
31: ***RingCT*** $\longleftarrow choose\_prev\_txs \in CRS$
32: ***Msg*** $\longleftarrow attached'KeyImage'$ ∗*KeyImage: to prevent storing the same data*
33: ***CurrMsg*** $\longleftarrow Sign\,\sigma \parallel RingCT \parallel KeyImage$
34: **end procedure**
35: **Procedure** *Send to Blockchain Network*:
36: ***Get*** $CurrMsg \oplus OTP(from\ StealthAddr)$
37: ***Send*** *to blockchain network*
38: ***Wait*** *miner's confirmation*
39: ***if*** *CurrMsg is valid* ***then return*** True ∗*transaction is success*
40: ***Add*** *newBlock to BC network*
41: $AddCurrMsg \longrightarrow Shared\_Storage$
42: ***elsereturn*** False
43: **End**

Algorithm 1: Shared Storage of PHI Data.

address received. Only patients can access data stored in shared storage by using his knowledge $Pr_\alpha(a_\alpha, b_\alpha)$.

$$OTP = H_s\left(rA_\alpha\right)G + B_\alpha \tag{8}$$

(v) The sender unpacks the address received which contains the public address of the recipient $Pub_\alpha(A_\alpha, B_\alpha)$. The sender picks the random value $r \in [1, l-1]$ to generate one-time public key and attaches B_α as a view address in the shared storage.

(vi) For certain types of data, the sender can use confidential ring signatures *(RingCT)* to disguise information of the data by adding value to transactions that have occurred before such as $RCT_\beta = txid(3) \parallel txid(7) \parallel txid(5) \parallel txid(9) \parallel txid(n)$.

(vii) The hospital signs the diagnosis data *msg* using the member key from the ring signature as follows: $\boldsymbol{Sign}\,\boldsymbol{\sigma}(msg, y_\alpha, y_\beta, y_\gamma, y_\delta, y_\varepsilon, y_\zeta, y_\eta, \ldots, y_n)$ which consists of the public keys of the members $Pub_\alpha, Pub_\beta, Pub_\gamma, Pub_\delta, Pub_\varepsilon, Pub_\zeta$, and Pub_η.

(viii) Key image is attached by the sender to prevent double spending. It can be interpreted as a scheme to prevent the same data stored twice in blockchain storage. The hospital sends the following series of data to the blockchain network to be confirmed by miners: $Sign\,\sigma \parallel msg \parallel OTP \parallel RingCT \parallel keyimage$.

(ix) The diagnosis data from the hospital along with attachments of the files are then sent to the blockchain network to be verified by the miners before the data are stored in a decentralized shared storage. Whenever the data has been stored successfully, the patient traces one by one the transaction on the blockchain network until the patient finds his/her view keys B_n which correspond to the patient.

(x) The patient with his/her knowledge decrypts and compares the value of the data received with the decrypted value $a_\alpha R = a_\alpha rG = rA_\alpha; P' = P$.

The Algorithm 1 presents an overall model of the system which starts with generating key pairs for the parties, creating a ring signature group and stealth addresses, signing PHI data until data are confirmed and stored in the decentralized shared storage. The observer is unlikely to track the sender's transaction since the sender uses a signature on behalf of a group in which the sender's signature is obscured. Furthermore, the observer cannot associate one transaction with another, so it is indistinguishable whether the transaction was sent by the same sender. In other words, the identity of the sender remains a secret. Likewise, the identity of the recipient cannot be tracked by the observer nor malicious providers because the recipient sends the stealth address to the sender; thereafter, the sender creates a one-time address to protect the privacy of the recipient. The value of a PHI data is kept from being seen because it is combined with the value of previous transactions through the *RingCT*. The identity of the user along with the information of the transactions on the blockchain network is paramount; therefore, the decentralized shared storage systems need to manage the confidentiality as well as ensure the integrity of the PHI data.

Apart from the model that has been described, there is another important factor which plays the role to a decentralized system called transaction propagation. A numerous number of transactions are distributed to the entire nodes in the blockchain peer-to-peer network, resulting in propagation delay. The structure of a P2P network allows the peer to disseminate information to the other nodes that are connected to the sender [24]. There are several studies conducted by researchers to measure how effective the block distribution in the peer-to-peer networks such as experiments in which the goal is to find out the number of successful connections and experiments to determine the effect of the number of nodes against the block.

One among parameters that affect transaction propagation is block size. Block size can be understood as the maximum limit of a block to be filled up with various transactions on it. It also can be thought of as a bundle of transactions, with each block needing to get verification before it can be accepted by the network. Each block has its own size depending on the type of transaction called the block size. The maximum block size in Bitcoin stands at 1MB. Miners enable to choose the number of transactions to be processed in a block. If Bitcoin miners commit a transaction that exceeds the maximum limit, then the block will be rejected by the network. The motivation of block size is to prevent the attack such as denial-of-service attacks. The size of a block also affects the length of confirmation time. When a node receives a new transaction, the recipient confirms the validity of the block before accepting it. The duration of the confirmation process depends on the size of the block. By design, the larger the size of a block is, the longer it takes to confirm. Therefore, the size of a block plays an important role in the blockchain in general because it directly affects the delay time.

Propagation delay is inseparable from the size of a block. There is a correlation between block size and the propagation delay until the node receives the block. The larger the size of a block, the more transactions that can be done, yet it affects the propagation time and sacrifice the security. The influence of a block size to the propagation time can be seen in Figure 10 [25]. The block size in the transaction is varied up to 350 KB in order to find out how long it takes for the node to receive the block. To reach 25% of total transaction is visualized with a red line, 50% for the green line, and 75% for the blue line. The results are in line with the theory which states the larger the size of a block, the greater the propagation time.

We take measurements for orphaned blocks by following the withholding attack (selfish mining) strategy that was first discovered in 2014. In this study we do not elaborate on the details of withholding attack strategy, we recommend that readers refer to the references [26–28]. The intuition of this attack is to keep the finding block secret until the attacker's network becomes the longest chain in the blockchain network. For a particular condition, this attack does not possess any benefit because the block is stored in the attacker's network which is only known by the attacker (nonprofit). The attackers get the reward if only the block found is propagated to the public and get confirmed by other miners.

The simulation is carried out in order to know the performance of dishonest miners which follow the selfish mining protocol. The aims of this strategy are to discover the new block till becoming the longest chain and gaining the revenue after publishing the block to the public network [29]. In our setting, we arrange the selfish miners to compete with honest miners in 14 days to solve the proof-of-work and discover the new block. The maximum of mining power of selfish miner is 0.4 and it is running randomly from 0.0 through 0.4 within 14 days in the simulation as shown in Figure 11. There are 12 nodes of dishonest miners with different mining power from 0.0 to 0.4. We set the maximum number of mining power at 0.4 of total mining power in the network. Based on the simulation result, we conclude that whenever the dishonest miner has 0.322 mining power, it is enough to get the unfair revenue and allows gaining the revenue larger than it should be. In general, there are 51 new blocks successfully added as well as 4 orphaned blocks recorded. The average of block generation time is 7.72 minutes.

To the extent, we obtained the information related to the parameters that affect the propagation time in the blockchain peer-to-peer network based on our findings and the prior works of literature. There are numerous articles proposing the improvement of propagation delays by changing the network topology, minimizing the verification time of a transaction,

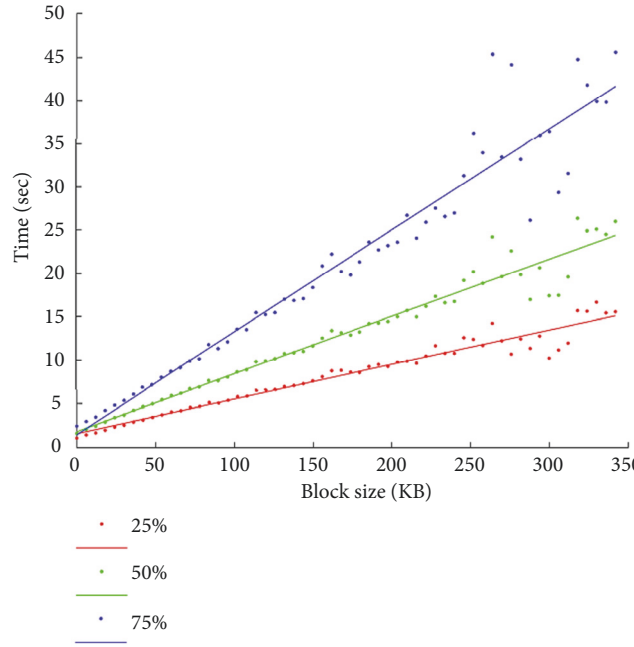

Figure 10: Relationship between block size and the time.

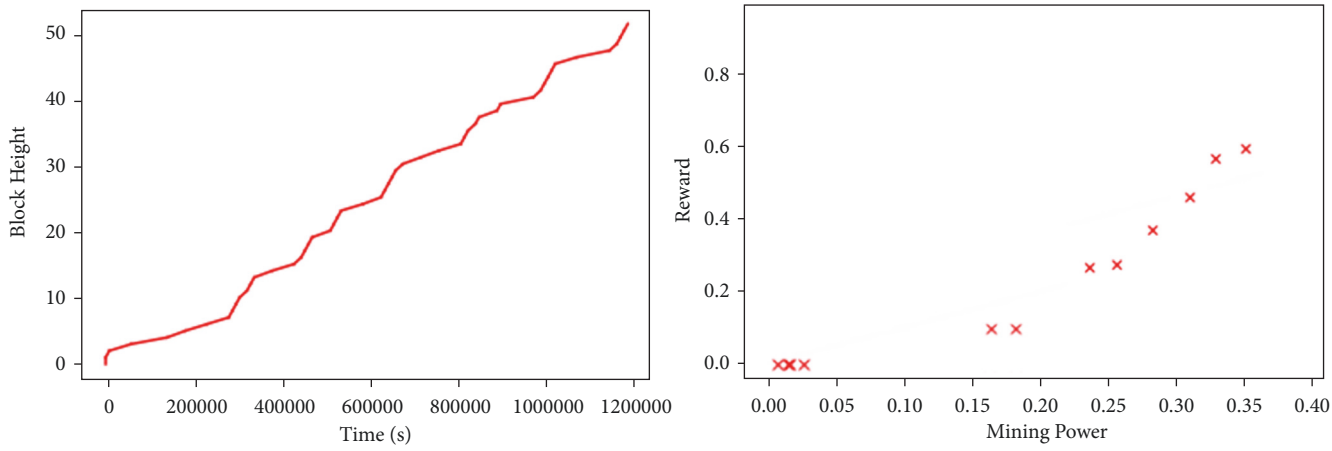

Figure 11: Block height against time in the P2P network (left); the performance of selfish mining attack (right).

and reconstructing the message exchange protocol in the blockchain network. Generally speaking, the presence of having the propagation of delays can cause a lot of damage in the blockchain network [30, 31].

There are many considerations to increase the effectiveness of the blockchain. Therefore, we select block propagation and block size parameters to be discussed as follows:

(i) *Speeding up the block generation*. In theory, it would be remarkably beneficial if the block generation time is resetting as fast as possible for each transaction so that each user will get faster payments. The propagation time includes the length of time for the distribution of transactions in the peer-to-peer network and the time for verification of a block. But the problem is that if the block generation time is speeding up, there will be a lot of orphaned blocks. It can be understood because each node will receive many new blocks that are spread through peer-to-peer networks. The *tie-breaking* protocol causes each node to only accept one valid block for the same type of transaction, so that it will reject transactions from other nodes that cause the orphaned block to appear. Due to many new orphaned blocks that will emerge, it will motivate rational miners to adopt the selfish mining strategy that rivals the main network and causes the forked chain [32].

(ii) *Decrease block generation*. Slowing down the block generation time for each transaction will reduce the speed of transactions on the peer-to-peer network. Positively, it gives some merits such as providing a

better security system. Roughly this happens due to a decreasing number of orphaned blocks that will eliminate the selfish mining attack.

(iii) *Increasing the block size*. The bigger the capacity of a block, the more the transactions that can be filled up into the block. It will slow the propagation of every transaction to all nodes in the peer-to-peer network. The slow propagation time will cause new problems, namely, double-spending attack. Attacker could use the same coin for two or more different transactions. Slower propagation of blocks on the peer-to-peer network resulted in the fact that the block cannot be fully accepted by the nodes. As a result, when the transaction is received, the block will confirm that the block is valid even though the transaction has been used and confirmed by other nodes in the same network.

(iv) *Reducing the capacity of block size*. Because of only a few transactions that occur within a block, it will speed up the propagation time for every transaction. It causes many orphaned blocks to emerge and allow for the occurrence of selfish mining attacks that harm the system. Yet, this decision gives the advantages such as fast payments and fast transaction. However, it should ensure the immutability of the block and transaction [33].

5. Limited and Future Work

We assume that shared storage is interconnected to a blockchain network where miners have access into it. In terms of the type of shared storage, we do not define it in detail; instead we assume the shared storage has all the capacity needed to support the proposed system. One of the drawbacks of our system is related to the time needed by the recipient to find data in shared storage, because the recipient must seek for data one by one based on the key view, so that the patient observes each transaction in the blockchain. In the long run, this might become an obstacle if the blockchain network is expanding. There will be many transactions occurring so that it will be difficult for the recipient to monitor every transaction which belongs to him/her.

For future work, the model and capacity of shared storage need to be observed further. The access control in the shared storage is also essential to ensure that system keeps safe. Incentive mechanisms also need to be considered for the storage providers. It aims to motivate providers to contribute to protecting the privacy of the users as suggested by [34–36]. Furthermore, it is important to carefully consider the type of block to be used including parameters directly involved in the system such as block size and type of consensus, to name a few. Additionally, the consensus selection mechanism is paramount in the blockchain. For instance, a new method by expanding the Byzantine consensus via hardware-assisted secret sharing can solve the scalability problem in the blockchain [37].

6. Conclusions

The model of privacy in the blockchain peer-to-peer shared storage has been fully presented. The idea of ring signature combined with several protocols provides the solutions for privacy issues on the blockchain transaction. The identity of the sender and the recipient remains hidden, unlinkable, and untraceable from the observer. The key image is attached to prevent the same data from being stored multiple times, and it can be used as well to prevent double spending and data duplication. Based on our findings and information from several literature reviews, it can be said that increasing the performance of the decentralized blockchain requires a very deep analysis since it is directly related to the security. There are advantages and drawbacks for each decision taken. For future work, a scheme that provides incentives needs to be developed as a motivation to maintain the decentralized system.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2019-0-00403) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation) and partially supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2018R1D1A1B07048944).

References

[1] G. Karame and E. Androulaki, *Bitcoin and Blockchain Security*, Artech House Information Security and Privacy Series, 2016.

[2] R. J. Krawiec, D. Housman, M. White et al., "Blockchain: Opportunities for health care," in *Proceedings of the NIST Workshop Blockchain Healthcare*, pp. 1–16, 2016.

[3] D. E. O'Leary, "Configuring blockchain architectures for transaction information in blockchain consortiums: The case of accounting and supply chain systems," *Intelligent Systems in Accounting, Finance and Management*, vol. 24, no. 4, pp. 138–147, 2017.

[4] D. Yang, J. Gavigan, and Z. W. Hearn, "Survey of Confidentiality and Privacy Preserving Technologies for Blockchains," R3, pp. 1–32, 2016.

[5] H. Kopp, D. Mödinger, F. Hauck, F. Kargl, and C. Bösch, "Design of a privacy-preserving decentralized file storage with financial incentives," in *Proceedings of the 2nd IEEE European Symposium on Security and Privacy Workshops, EuroS and PW 2017*, pp. 14–22, April 2017.

[6] T. W. Clarke, I. Sandberg, O. Wiley, and B. Hong, "A distributed anonymous information storage and retrieval system," *Journal of Chemical Information and Modeling*, vol. 53, no. 9, pp. 1689–1699, 2001.

[7] E. Heilman, L. AlShenibr, F. Baldimtsi, A. Scafuro, and S. Goldberg, "TumbleBit: an untrusted bitcoin-compatible anonymous payment hub," in *Proceedings of the 2017 Network and Distributed System Security Symposium*, 2017.

[8] S. Wu, Y. Chen, Q. Wang, M. Li, C. Wang, and X. Luo, "CReam: a smart contract enabled collusion-resistant e-auction," *IEEE Transactions on Information Forensics*, 2018.

[9] J. Benet and N. Greco, "Filecoin: A Decentralized Storage Network," *Protoc. Labs*, 2018.

[10] S. Wilkinson, T. Boshevski, J. Brandoff, and V. Buterin, "Storj a peer-to-peer cloud storage network," Technical report, storj.io, 2014.

[11] B. Fisch, "Poreps: Proofs of space on useful data," Report 2018/678, Cryptology ePrint Archive, 2018.

[12] S. Rahmadika and K. Rhee, "Blockchain technology for providing an architecture model of decentralized personal health information," *International Journal of Engineering Business Management*, vol. 10, pp. 1–12, 2018.

[13] R. L. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret," in *International Conference on the Theory and Application of Cryptology and Information Security*, vol. 2248 of *Lecture Notes in Comput. Sci.*, pp. 552–565, Springer.

[14] D. Chaum, E. van Heyst, and C. Science, "Group Signatures," in *Adv. Cryptology—EUROCRYPT'91*, vol. iii, pp. 257–265, 1991.

[15] N. Van Saberhagen, "CryptoNote v 2.0," *Self-published*, pp. 1–20, 2013.

[16] S. Noether, A. Mackenzie, and T. M. Research Lab, "Ring confidential transactions," *Ledger*, vol. 1, pp. 1–8, 2016.

[17] E. Gallery and C. J. Mitchell, "Trusted computing: Security and applications," *Cryptologia*, vol. 33, no. 3, pp. 217–245, 2009.

[18] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[19] A. A. Korba, M. Nafaa, and S. Ghanemi, "An efficient intrusion detection and prevention framework for ad hoc networks," *Information and Computer Security*, vol. 24, no. 4, pp. 298–325, 2016.

[20] S. Rahmadika, P. H. Rusmin, H. Hindersah, and K. H. Rhee, "Providing data integrity for container dwelling time in the seaport," in *Proceedings of the 6th International Annual Engineering Seminar, InAES 2016*, pp. 132–137, Indonesia, August 2016.

[21] K. Schmidt-Samoa, "A new rabin-type trapdoor permutation equivalent to factoring," *Electronic Notes in Theoretical Computer Science*, vol. 157, no. 3, pp. 79–94, 2006.

[22] A. Bender, J. Katz, and R. Morselli, "Ring signatures: stronger definitions, and constructions without random oracles," *Journal of Cryptology. The Journal of the International Association for Cryptologic Research*, vol. 22, no. 1, pp. 114–138, 2009.

[23] K. Lee and A. Miller, "Authenticated data structures for privacy-preserving monero light clients," in *Proceedings of the 3rd IEEE European Symposium on Security and Privacy Workshops, EURO S and PW 2018*, April 2018.

[24] R. Schollmeier, "A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications," in *Proceedings of the 1st International Conference on Peer-to-Peer Computing, P2P 2001*, pp. 101-102, August 2001.

[25] Y. Sompolinsky and A. Zohar, "Accelerating bitcoins transaction processing. fast money grows on trees, not chains," *Eprint.Iacr.Org*, 2014.

[26] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 8437, pp. 436–454, 2014.

[27] A. Sapirshtein, Y. Sompolinsky, and A. Zohar, "Optimal selfish mining strategies in bitcoin," in *Financial Cryptography and Data Security*, vol. 9603 of *Lecture Notes in Computer Science*, pp. 515–532, Springer, Berlin, Germany, 2017.

[28] E. Heilman, "One weird trick to stop selfish miners: fresh bitcoins, a solution for the honest miner (poster abstract)," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 8438, pp. 161-162, 2014.

[29] S. Rahmadika, B. J. Kweka, H. Kim, and K. Rhee, "A scoping review in defend against selfish mining attack in bitcoin," *IT Converg. Pract*, vol. 6, no. 3, pp. 18–26, 2018.

[30] A. Gervais, H. Ritzdorf, G. O. Karame, and S. Capkun, "Tampering with the delivery of blocks and transactions in bitcoin," in *Proceedings of the the 22nd ACM SIGSAC Conference*, pp. 692–705, Denver, Colo, USA, October 2015.

[31] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Čapkun, "On the security and performance of Proof of Work blockchains," in *Proceedings of the 23rd ACM Conference on Computer and Communications Security, CCS 2016*, pp. 3–16, October 2016.

[32] D. T. T. Anh, M. Zhang, B. C. Ooi, and G. Chen, "Untangling blockchain: a data processing view of blockchain systems," *IEEE Transactions on Knowledge and Data Engineering*, 2018.

[33] D. Mendez Mena, I. Papapanagiotou, and B. Yang, "Internet of things: survey on security," *Information Security Journal*, vol. 27, no. 3, pp. 162–182, 2018.

[34] O. Ersoy, Z. Ren, Z. Erkin, and R. L. Lagendijk, "Transaction propagation on permissionless blockchains: incentive and routing mechanisms," in *Proceedings of the 2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, pp. 20–30, June 2018.

[35] Y. He, H. Li, X. Cheng, Y. Liu, C. Yang, and L. Sun, "A blockchain based truthful incentive mechanism for distributed P2P applications," *IEEE Access*, vol. 6, pp. 27324–27335, 2018.

[36] J. Wang, M. Li, Y. He, H. Li, K. Xiao, and C. Wang, "A blockchain based privacy-preserving incentive mechanism in crowdsensing applications," *IEEE Access*, vol. 6, pp. 17545–17556, 2018.

[37] J. Liu, W. Li, G. O. Karame, and N. Asokan, "Scalable Byzantine consensus via hardware-assisted secret sharing," *Institute of Electrical and Electronics Engineers. Transactions on Computers*, vol. 68, no. 1, pp. 139–151, 2019.