mathematics

MDPI

Article

Threshold-Based Post-Quantum Secure Verifiable Multi-Secret Sharing for Distributed Storage Blockchain

Sihem Mesnager [1,2,3,*,†], Ahmet Sınak [2,4,†] and Oğuz Yayla [5,†]

1 Department of Mathematics, University of Paris VIII, 93526 Saint-Denis, France
2 LAGA UMR 7539, CNRS, Sorbonne Paris Cité, University of Paris XIII, 93430 Villetaneuse, France; asinak@erbakan.edu.tr
3 Telecom Paris, 91120 Palaiseau, France
4 Department of Mathematics and Computer Science, Necmettin Erbakan University, 42090 Konya, Turkey
5 Institute of Applied Mathematics, Middle East Technical University, 06800 Ankara, Turkey; oguz@metu.edu.tr
* Correspondence: smesnager@univ-paris8.fr
† These authors contributed equally to this work.

Received: 26 October 2020; Accepted: 11 December 2020; Published: 14 December 2020

Abstract: Blockchain systems store transaction data in the form of a distributed ledger where each node stores a copy of all data, which gives rise to storage issues. It is well-known that the tremendous storage and distribution of the block data are common problems in blockchain systems. In the literature, some types of secret sharing schemes are employed to overcome these problems. The secret sharing method is one of the most significant cryptographic protocols used to ensure the privacy of the data. The main purpose of this paper is to improve the recent distributed storage blockchain systems by proposing an alternative secret sharing method. We first propose a secure threshold verifiable multi-secret sharing scheme that has the verification and private communication steps based on post-quantum lattice-based hard problems. We then apply the proposed threshold scheme to the distributed storage blockchain (DSB) system to share transaction data at each block. In the proposed DSB system, we encrypt the data block with the AES-256 encryption algorithm before distributing it among nodes at each block, and both its secret key and the hash value of the block are privately shared among nodes simultaneously by the proposed scheme. Thereafter, in the DSB system, the encrypted data block is encoded by the Reed–Solomon code, and it is shared among nodes. We finally analyze the storage and recovery communication costs and the robustness of the proposed DSB system. We observe that our approach improves effectively the recovery communication cost and makes it more robust compared to the previous DSB systems. It also improves extremely the storage cost of the traditional blockchain systems. Furthermore, the proposed scheme brings to the DSB system the desirable properties such as verification process and secret communication without private channels in addition to the known properties of the schemes used in the previous DSB systems. As a result of the flexibility on the threshold parameter of the scheme, a diverse range of qualified subsets of nodes in the DSB system can privately recover the secret values.

Keywords: blockchain; distributed storage blockchain; verifiable secret sharing scheme

1. Introduction

Blockchain is an emerging technology that has many interesting real-world application areas such as medical, energy, and financial. However, there are several restrictions on this recent technology. The most significant one is the storage issue in blockchain systems since each node has to store a copy

of all blocks. As time continues, storage is a huge problem because the number of blocks continuously increases in blockchain systems. Secret sharing mechanisms have been used in blockchain systems to share the data block and the secret values among nodes. They assist to strengthen the decentralization and security of data in blockchain as it helps to distribute information in a decentralized way such that the private information is protected from unauthorized access. By secret sharing method, blockchain systems can store information so that every node stores a certain number of shares instead of the entire body of data. Taking into account these benefits, the secret sharing method has vital importance in blockchain systems.

Recently, the concept of distributed storage blockchain has been proposed to distribute the storage costs by the secret sharing method among all nodes in the blockchain network (see for instance [1–5]). In these works, some types of secret sharing methods such as Shamir's secret sharing in [4,5], multi-secret sharing in [1], and local secret sharing in [3] have been employed to distribute the block data among nodes in the blockchain network. In this framework, we incorporate the threshold verifiable multi-secret sharing scheme, AES encryption algorithm for privacy, and Reed–Solomon (RS) code for encoding into the standard distribute storage blockchain to distribute privately the block data among nodes in the blockchain network.

In medical systems, a lot of devices are connected to share remotely the patient data, to make a decision on the health status of the patient, or to make research on the medical data anonymously. This system is so-called the internet of medical things (IoMT). IoMTs need not only to decide on machine learning tools but also to exchange private data with each other. The data exchange can be done either with a central authority or in a decentralized manner. In the later one, blockchain is recently utilized to deploy a practical solution for solving the privacy and security issues, where data updates are stored as blockchain transactions in the system, see [6–11]. It seems that it is vital to find new methods to enhance the privacy of the data stored in the blockchain ledger and to reduce the amount of data stored by each IoMT device.

The main contributions of the paper are listed as follows. First, motivated by the previous secret sharing methods introduced in [12,13], we enhance a threshold-based verifiable multi-secret sharing (VMSS) scheme without private channels, which is one of the well-known secret sharing schemes in cryptography. Second, inspired by the previous works [1–4], we apply the proposed threshold-based VMSS scheme to the distributed storage blockchain (DSB) system to distribute block data among all nodes in a blockchain network. We finally analyze the storage and recovery communication costs and the robustness of the DSB system based on the VMSS scheme. The proposed method reduces the recovery communication cost and improves robustness in the previous DSB systems. It also improves significantly the storage cost of traditional blockchain systems. In addition to the desirable properties of the previous schemes used in the DSB systems, the proposed scheme has the (quantum secure) verification algorithm and secret communication without private channels. We also note that the flexible threshold parameter of the proposed scheme eliminates a drawback of the previous DSB systems on their recovery communication costs and robustness.

The remainder of the paper is structured as follows. In Section 2, we introduce brief history of secret sharing schemes. In Section 3, we give the previous studies and background for distributed storage blockchain systems. In Section 4, we propose a secure threshold-based VMSS scheme that shares securely both a secret key and hash value among nodes in the blockchain system. In Section 5, we incorporate the proposed VMSS scheme into the DSB system. We also explain the distributing and recovering processes of the data block in the proposed DSB system. We finally analyze the storage and recovery communication costs as well as the robustness of the proposed method. We notice that our recovery communication cost and robustness are much better than the previous ones. We conclude the paper in Section 6.

2. Related Works

In this section, we mention the previous studies on the types of secret sharing schemes. A secret sharing scheme is one of the most significant cryptographic protocols for sharing data securely. The first secret sharing scheme was introduced in 1979 by Blakley [14] and Shamir [15], independently, which are the threshold-based schemes. Shamir's secret sharing scheme is based on polynomial interpolation over finite fields while Blakley's scheme is based on finite geometry.

A secret sharing scheme consists of a dealer D, a group $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$ of n participants, a secret space S, n share spaces $S_1, S_2, \ldots, S_n$, a share computing procedure, and a secret recovering procedure. The dealer D chooses a secret s from S, and computes a share of s (with the sharing computing procedure) for each participant P_i and then gives the share to P_i for $i \in \{1, \ldots, n\}$. The sharing computing procedure and the secret s are known only by D, while the secret recovering procedure is known by all participants in $\mathcal{P}$. A set of participants who can recover s from their shares is said to be an access set. Indeed, an access set is said to be a minimal access set if any of its proper subsets cannot recover s from their shares. The set of all access sets is said to be an access structure of a scheme.

The usual sharing scheme can only resist passive attacks but not active ones; that is, it is not secure against the dishonest dealer and the malicious participants. Thus, the dealer and the participants are generally assumed to be honest, however, this assumption is not realistic in real-life applications. To eliminate this assumption, the first verifiable secret sharing (VSS) scheme has been introduced in 1985 by Chor et al. [16] by adding the verification algorithm to Shamir's scheme, and later several VSS schemes have been proposed in the literature (see [12,17–20]). In a VSS scheme, not only a dishonest dealer but also a malicious participant can be easily detected utilizing the verification process. It can be then said that a VSS scheme resists against two kinds of active attacks:

- Dishonest dealer can tamper with a share before sending it to participants in the construction protocol.
- Any malicious participant can submit a fake share to the recovery protocol.

In 1995, He and Dawson [21] introduced the first multi-secret sharing (MSS) scheme based on Shamir's scheme that shares multiple secret values simultaneously. In an MSS scheme, only one share is assigned to each participant (indeed, each participant needs to protect only one share) while multiple secrets can be shared. Note that the size of the assigned share is almost the same as that of each secret value.

Harn (1995) [22] introduced the first threshold-based verifiable multi-secret sharing (VMSS) scheme that not only shares multiple secrets simultaneously but also detects the dishonest dealer and participants. Moreover, several threshold-based VMSS schemes have been widely studied in the literature (see [12,13,17,20]). It can be said that the VMSS schemes are secure against both passive and active attacks.

In the usual secret sharing scheme, it is generally assumed that the shares are distributed and collected by the dealer through secure channels. However, the establishment of secure channels between the dealer and the participants has high requirements in the protocol. Thus, for the secure communication between them in a public channel, several techniques were proposed in the literature, one of which is public-key cryptography. For example, Hwang and Chang [23] and Liu et al. [12] made use of the RSA public-key encryption algorithm [24] in their VMSS schemes for secure communication while Zhao et al. [20] used the Diffie–Helman key exchange protocol [25] in their practical VMSS scheme.

3. Preliminaries

In this section, the fundamental concepts of data storage in the blockchain systems are briefly given. We start with the traditional blockchain system, and then we discuss the distributed storage blockchain system.

In the traditional blockchain system, every data block and the hash value pointing to the previous block is stored by each node. It is formalized as follows. Let h_1 and h_2 be two hash functions. Let c be a constant and $H^{(0)} = c$. Let $B^{(t)}$ be the data block to be stored in the t-th block and $W^{(t)} = (H^{(t-1)}, h_2(B^{(t)}))$, where $H^{(t-1)} = h_1(W^{(t-1)})$ for $t = 1, 2, 3, \ldots$. Every node in traditional blockchain system stores all pairs $B^{(t)}$ and $W^{(t)}$ for $t = 1, 2, 3, \ldots$. For example, when a block $B^{(t_0)}$ is created for some t_0 by a node N, N needs to share the pair $B^{(t_0)}$ and $W^{(t_0)}$ with other nodes in the blockchain network. Then, all nodes will have a copy of the block in their storage. This brings a lot of storage costs for each node. Similarly, if a block is lost in a node, then it can be recovered by accessing any node in the blockchain network and copying the data block and the hash value, which is known as recovery communication cost. Hence, the traditional blockchain system has storage and recovery communication costs proportional to the size of $|B^{(t)}| + |W^{(t)}|$ for each node N at each block t. The maximum number of node failure which can be tolerated by the blockchain network is called its robustness. It is easy to observe that the traditional blockchain network with n nodes has the robustness $n - 1$.

The concept of distributed storage blockchain has been recently studied to reduce the storage cost of traditional blockchain systems. First, Dai et al. [2] have adopted network coding to the notion of distributed storage to reduce the storage space for distributed ledger in blockchain systems, and they achieved significant improvement. Second, Raman and Varshney [4,5] have recently proposed the idea of a distributed storage blockchain, which significantly decreases the storage of transactions by using Shamir's sharing scheme. In DSB, all nodes (say, n nodes exist) are divided into L distinct subsets of equal size m, that is, let $\mathcal{A} = \{A_1, \ldots, A_L\}$ be the partition of the set of n nodes, and $n = m \cdot L$. Each subset A_l has the secret key $K_l^{(t)}$ to encrypt a block $B^{(t)}$ as $m_l^{(t)} = E_{K_l^{(t)}}(B^{(t)})$ for $l = 1, \ldots, L$. Then, $m_l^{(t)}$ is divided into m pieces and distributed to each node in A_l. Besides, the secret key $K_l^{(t)}$ (the local secret) and the hash value $W^{(t)}$ (the global secret) are shared to each node in A_l by two independent Shamir's (m, m) sharing schemes. Their data distribution method is formalized in Algorithm 1.

Algorithm 1 DSB in [4,5]

Input. Given a partition $\mathcal{A} = \{A_1, \ldots, A_{\frac{n}{m}}\}$

1: **for** $l = 1$ to $\frac{n}{m}$ **do**
2: Generate the secret key $K_l^{(t)}$.
3: Encrypt $B^{(t)}$ with $K_l^{(t)}$ as $m_l^{(t)} = E_{K_l^{(t)}}(B^{(t)})$.
4: Distribute and store $m_l^{(t)}$ among m peers in A_l.
5: Store $K_l^{(t)}$ and $W^{(t)}$ by (m, m) Shamir's sharing.
6: **end for**

As seen in Algorithm 1, in DSB, each node has $\frac{|B^{(t)}|}{m} + 2|W^{(t)}|$ storage cost and $|B^{(t)}| + 2m|W^{(t)}|$ recovery communication cost at each block t since the secret key and the hash value may be recovered by accessing m nodes in another subset. If the size of the hash value is extremely small compared to the size of the data block, which is usually the case in real-life applications, then the storage cost of traditional blockchain is excessively reduced by the DSB system. On the other hand, a single node failure in a subset A_l causes the loss of the key in A_l and so, data in A_l can not be reachable anymore. This says that a blockchain network based on DSB with n nodes has robustness $\frac{n}{m} - 1$. We finally note that a single node failure in every subset in DSB causes the loss of the blockchain data inevitably.

Recently, Kim et al. [3] have proposed a local secret sharing (LSS) scheme to improve the DSB storage by using locally recoverable codes (LRC) [26] and trivial maximum distance separable (MDS) codes [27] (Chapter 11). In particular, they first obtain the LSS scheme from LRC, and then LSS is used suitably in DSB. In DSB with LSS, the hash value $W^{(t)}$ and the secret key $K_l^{(t)}$ are simultaneously shared by LSS through all nodes in the blockchain network. Since a single error in an $(n, k, m-1)$-LRC of length n and dimension k can be recovered by $(m-1)$ correct symbols [26], then a single node failure

in DSB with LSS can be tolerated by the blockchain network due to the proposed $(m, m-1)$-threshold LSS scheme. Right after, the encrypted data block is encoded by a trivial $(m, m-1)$-MDS code, which is a code with a single parity symbol. Hence, a blockchain network based on DSB with LSS has robustness $\frac{2n}{m} - 1$, and each node has $\frac{|B^{(t)}|}{m-1} + |W^{(t)}|$ storage cost and $|B^{(t)}| + (m-1)|W^{(t)}|$ recovery communication cost at each block t.

Very recently, to improve the DSB system, Chen et al. [1] have proposed a low-storage scheme with a multi-secret sharing (MSS) scheme based on polynomial interpolation. The DSB with MSS divides the transaction block into multiple pieces and then stores them in different nodes, but it does not encrypt the transaction block. It stores only data block but not secret key and hash value. In this system, the block $B^{(t)}$ is to be shared between n parties. They first divide the block $B^{(t)}$ into m equal length pieces denoted by $b_1, b_2, \ldots, b_m$ such that their concatenation $b_1||b_2||\ldots||b_m = B^{(t)}$ and $m < n$. The proposed MSS is based on recursion, and to encode the piece b_i for $i = 1, \ldots m$, it generates a sharing polynomial $g_i(x) = b_i + g_{(i-1,1)}x + g_{(i-1,2)}x^2 + \cdots + g_{(i-1,i)}x^i$ of degree i over a finite field $\mathbb{F}_q$, where q is a large odd prime greater than pieces b_i and n. Then, it distributes the shares $g_m(x_1), g_m(x_2), \ldots, g_m(x_n)$ to the corresponding nodes, where $x_1, x_2, \ldots, x_n$ are the public indexes of nodes. This scheme is an $(n, m+1)$-threshold secret sharing since any $m+1$ nodes or more can reconstruct the block $B^{(t)}$, but no group of m or fewer nodes can do so. The reconstruction of this scheme is an inverse process. Any $m+1$ of nodes can first reconstruct a polynomial $g_m(x)$ of degree m with the constant term b_m, and then reconstruct recursively a polynomial $g_i(x)$ of degree i with the constant term b_i for $i = 1, \ldots, m-1$. Thereafter, the block $B^{(t)}$ is obtained by concatenating $b_1||b_2||\ldots||b_m = B^{(t)}$. Hence, in DSB with MSS scheme [1], each node has $\frac{|B^{(t)}|}{m}$ storage cost and $|B^{(t)}|$ recovery cost, and its robustness is $n - m - 1$.

4. A Threshold-Based Verifiable Multi-Secret Sharing (VMSS) Scheme

In this section, we propose a secure threshold-based verifiable multi-secret sharing scheme based on Feldman's VSS scheme introduced in [19] (originally, based on Shamir's threshold scheme introduced in [15]).

4.1. Description of the Proposed Threshold-Based VMSS Scheme

Let n be a positive integer. Let D be the dealer and $\mathcal{P} = \{P_1, \ldots, P_n\}$ be a set of n participants. Let the threshold be denoted by t with $2 \leq t \leq n$. Let $\mathbb{F}_p$ be a finite field for a prime $p > 2n - t + 1$, and $\mathbb{F}_p^* = \mathbb{F}_p \setminus \{0\}$. Let $\mathcal{F}$ be a function on $\mathbb{F}_p$. These parameters are generated cooperatively by D and all participants. We now describe a secure (n, t)-threshold verifiable multi-secret sharing scheme without private channels.

- **Construction phase.** Let two distinct secrets s_0 and s_1 in $\mathbb{F}_p^*$ be given to be shared. The dealer D performs the following steps.
  - D chooses random elements $a_k \in \mathbb{F}_p$ for $k = 1, \ldots, t-1$, and constitutes the $(t-1)$-degree polynomial $f : \mathbb{F}_p \to \mathbb{F}_p$ as $f(x) = s_0 + a_1 x + \cdots + a_{t-1}x^{t-1}$.
  - D commits all coefficients of $f(x)$ by masking them with a function $\mathcal{F}$, namely computes $C_0 = \mathcal{F}(s_0)$, $C_k = \mathcal{F}(a_k)$ for $k = 1, \ldots, t-1$. This commitment guarantees that no one can do cheating in the scheme.
  - D broadcasts the public commitments: $(C_0, \ldots, C_{t-1})$ for verification.
  - D selects randomly distinct elements $x_i \in \mathbb{F}_p^*$ and computes the shares $y_i = f(x_i) \in \mathbb{F}_p$ for a participant P_i for $i = 1, \ldots, n$.
  - D constitutes the polynomial

$$f_1(x) = \sum_{i=0}^{n} y_i \prod_{j=0, j\neq i}^{n} \frac{x - x_j}{x_i - x_j} \tag{1}$$

of degree n by using $(n+1)$ points $(x_0 = 0, y_0 = s_1), (x_1, y_1), \ldots, (x_n, y_n) \in \mathbb{F}_p^2$ by the Lagrange interpolation method.
- D encrypts y_i with a public key encryption $\mathcal{E}$ as $c_i = \mathcal{E}_{K_i}(y_i)$ by using the public key K_i of P_i for $i = 1, \ldots, n$.
- D sends the pair (x_i, c_i) to the participant P_i in the public channel for $i = 1, \ldots, n$.
- D selects randomly distinct elements $x_v \in \mathbb{F}_p^* \backslash \{x_1, \ldots, x_n\}$, and evaluates $f_1(x_v)$ for $v \in \{n+1, \ldots, 2n-t+1\}$.
- D broadcasts the public points $(x_v, f_1(x_v))$ for $v \in \{n+1, \ldots, 2n-t+1\}$.

- **Verification phase.** Each participant P_i can perform the following verification operation to verify her own share.
  - P_i privately decrypts c_i with the public key decryption algorithm $\mathcal{D}$ as $y_i = \mathcal{D}_{K_i'}(c_i)$ by using her own private key K_i' for $i = 1, \ldots, n$.
  - P_i checks the validity of her share y_i and its consistency with the public information, namely, P_i verifies whether

$$\mathcal{F}(y_i) = \prod_{k=0}^{t-1} (C_k)^{x_i^k}. \tag{2}$$

  If the verification in (2) holds for every $i = 1, \ldots, n$, then each y_i is valid, and hence D is assumed to be honest.

- **Recovery phase.** Suppose that any t authorized participants $\{P_{j_1}, \ldots, P_{j_t}\} \subseteq \mathcal{P}$, where $\{j_1, \ldots, j_t\} \subseteq \{1, \ldots, n\}$, can recover the shared secrets s_0 and s_1.
  - Each P_{j_v} encrypts her share y_{j_v} with a public key encryption $\mathcal{E}$ as $z_{j_v} = \mathcal{E}_{K_D}(y_{j_v})$ by using the public key K_D of D, and sends the pair (x_{j_v}, z_{j_v}) to D in the public channels for $v \in \{1, \ldots, t\}$.
  - D decrypts z_{j_v} as $y_{j_v} = \mathcal{D}_{K_D'}(z_{j_v})$ by using her own private key K_D' for every $v \in \{1, \ldots, t\}$.
  - D verifies the validity of each y_{j_v} by using the verification equation in (2) for every $v \in \{1, \ldots, t\}$. If each share y_{j_v} is valid, then the points $(x_{j_1}, y_{j_1}), \ldots, (x_{j_t}, y_{j_t})$ are accepted from t authorized participants.
  - The authorized participants $\{P_{j_1}, \ldots, P_{j_t}\}$ can cooperatively reconstruct the secret s_0 by using their private points (x_{j_v}, y_{j_v}) for $v \in \{1, \ldots, t\}$ from the following formula

$$s_0 = \sum_{v=1}^{t} y_{j_v} \prod_{u=1, u \neq v}^{t} \frac{-x_{j_u}}{x_{j_v} - x_{j_u}}. \tag{3}$$

  Similarly, by using their private points (x_{j_v}, y_{j_v}) for $v \in \{1, \ldots, t\}$ and $(n-t+1)$ public points $(x_v, f_1(x_v))$ for $v \in \{n+1, \ldots, 2n-t+1\}$, they can cooperatively recover the secret s_1 from the the following formula

$$\begin{aligned} s_1 = & \sum_{v=1}^{t} y_{j_v} \prod_{u=1, u \neq v}^{t} \frac{-x_{j_u}}{x_{j_v} - x_{j_u}} \prod_{w=n+1}^{2n-t+1} \frac{-x_w}{x_{j_v} - x_w} \\ & + \sum_{v=n+1}^{2n-t+1} f_1(x_v) \prod_{u=1}^{t} \frac{-x_{j_u}}{x_v - x_{j_u}} \prod_{w=n+1, w \neq v}^{2n-t+1} \frac{-x_w}{x_v - x_w}. \end{aligned} \tag{4}$$

The proposed VMSS scheme has the following desirable properties to be applied in many practical systems such as decentralized mechanisms.

- The proposed scheme can simultaneously share two secrets while storing only one share by each participant.

- Due to the verification algorithm (2) of the proposed scheme, both the dishonest dealer and malicious participants can be easily detected. To be more precise, the dealer's cheating can be detected by a participant, and the dealer can detect any malicious participant.
- The dealer can securely communicate with participants through public channels since the shares are encrypted with the public key encryption algorithm. Indeed, the proposed scheme realizes secret sharing without a private channel, which is a very significant property in many practical applications where a private channel is very hard to be established.
- The participants can reuse repeatedly their shares in another reconstruction round because the employed function $f(x)$ is fixed and the shares are encrypted by the public key algorithm.

Remark 1. *Secret values s_0 and s_1 may have different threshold parameters, but we prefer to use the same threshold t for both s_0 and s_1 in the proposed scheme.*

We note that the proposed VMSS scheme has some assumptions on the securities of function $\mathcal{F}$, encryption $\mathcal{E}$, and decryption $\mathcal{D}$. In the literature, $\mathcal{F}$ is generally proposed to be the modular exponentiation function, so the security of the verification process depends on the hardness of the discrete logarithm problem (DLP). Similarly, for the encryption $\mathcal{E}$ and decryption $\mathcal{D}$, the RSA public key algorithm is generally proposed in the literature, and its security depends on the hardness of the integer factorization problem (IFP). In the proposed VMSS scheme, the modular exponentiation function and RSA public key algorithm may be preferred, respectively, for $\mathcal{F}$, $\mathcal{E}$, and $\mathcal{D}$. In this case, the security of the proposed scheme is based on two intractable problems DLP and IFP, which are assumed to be hard problems at present. On the other hand, these intractable problems are not quantum secure, and they can be broken by Shor's algorithm on a quantum computer. Therefore, we also suggest using quantum secure algorithms in the proposed scheme, which is rather important for its usability in the post-quantum world.

4.2. Post-Quantum Secure Methods

Lattice-based cryptosystems are known to be quantum secure as there has no feasible (traditional and quantum) attacks against them. Besides, lattices are so easy to implement in software and hardware environments. Therefore, several secret sharing schemes based on lattices were proposed in [17,28–31]. For instance, the knapsack function $\mathcal{F}_b : R^d \to R$,

$$\mathcal{F}_b(X = (X_1, X_2, ..., X_d)) = \langle X \cdot b \rangle = \sum_{i=0}^{d} X_i b_i \tag{5}$$

is proposed for the verification function in [31], where $R = \mathbb{Z}_p[x]/g(x)$ for some irreducible polynomial $g(x) \in \mathbb{Z}_p[x]$ of degree N, prime p, and random $b = (b_1, b_2, \ldots, b_d) \in R^d$. It is known that finding the inverse of $\mathcal{F}_b$ for any $b \in R^d$ is as hard as solving the approximate shortest polynomial problem [32].

In this case, $f(x) = a_0 + a_1 x + \cdots + a_{t-1} x^{t-1}$ is a polynomial over R^d such that $a_k \in R^d$ for $k = 0, \ldots, t-1$, and $f : \mathbb{Z}_p \to R^d$. Here, for simplicity, we denote $s_0 = a_0$. Then the verification phase (2) is performed by the participants and the dealer from the public commitments $\mathcal{F}_b(a_k)$ for $k = 0, \ldots, t-1$ as follows

$$\mathcal{F}_b(y_i) = \mathcal{F}_b(\sum_{k=0}^{t-1} a_k x_i^k) = \langle \sum_{k=0}^{t-1} a_k x_i^k \cdot b \rangle = \sum_{k=0}^{t-1} \langle a_k \cdot b \rangle x_i^k = \sum_{k=0}^{t-1} \mathcal{F}_b(a_k) x_i^k \tag{6}$$

for $i = 1, 2, \ldots, n$. Similarly, one can use lattice-based NTRU public key cryptosystem [33] for functions $\mathcal{E}$ and $\mathcal{D}$. Thus, the proposed VMSS becomes a lattice-based post-quantum scheme.

4.3. Security Analysis of the Proposed VMSS Scheme

In this subsection, we analyze the correctness and the security of the proposed scheme in terms of verifiability and privacy.

Theorem 1. *The proposed (n,t)-threshold VMSS scheme satisfies the following three security requirements.*

1. ***Correctness:*** *Any t or more honest participants can correctly recover the secrets s_0 and s_1 if D is honest.*
2. ***Verifiability:*** *D cannot distribute a fake share to any participant, and any participant cannot submit a false share to the recovery algorithm.*
3. ***Privacy:*** *Any group of less than t participants cannot reach the shared secrets s_0 and s_1.*

Proof.

1. The correctness of the proposed scheme follows from the recovery formula given in (4).
2. The dishonest D and any malicious participant cannot pass through the verification process given in (2) since the employed function $\mathcal{F}$ is secure against the known attacks. To be more precise, when the verification function $\mathcal{F}$ is based on the usual modular exponentiation function, its security depends on the DLP that is assumed to be a hard problem. Moreover, when $\mathcal{F}$ is the lattice-based knapsack function given in (5), the security of the verification process depends on the lattice-based hard problem that is assumed to be quantum secure.
3. An attacker cannot derive any private information from the public information in the proposed scheme. The possible scenarios are explicitly explained below.

    - An attacker cannot obtain any useful information about the committed secret values a_k from the public commitments $C_0 = \mathcal{F}(s_0)$ and $C_k = \mathcal{F}(a_k)$ since $\mathcal{F}$ is a secure function based on the DLP or lattice-based hard problem. Hence, an attacker cannot derive any information about the polynomial $f(x)$ from the public information.
    - An attacker cannot obtain any useful information about the private share y_i from the encrypted shares $c_i = \mathcal{E}_{K_i}(y_i)$ and $z_{j_v} = \mathcal{E}_{K_D}(y_{j_v})$ since the RSA algorithm and lattice-based NTRU system are secure cryptosystems. Thus, public information does not leak any information about the private shares of the participants.
    - Even if an attacker corrupts up to $(t-1)$ authorized participants in the proposed scheme, s/he still cannot get any useful information about the private share of any other honest participant, and so cannot reconstruct the secrets s_0 and s_1.

□

Theorem 2. *In the proposed (n,t)-threshold VMSS scheme, the verification is succeeded if D and participants follow correctly the protocol.*

Proof. Suppose that $\mathcal{F}$ is the modular exponentiation function. For simplicity, we assume $s_0 = a_0$ in $f(x)$. If D follows accurately the protocol, then we get the following

$$\mathcal{F}(y_i) = \mathcal{F}(f(x_i)) = \mathcal{F}(a_0 + a_1 x_i + \cdots + a_{t-1} x_i^{t-1}) = \mathcal{F}(a_0)\mathcal{F}(a_1)^{x_i} \cdots \mathcal{F}(a_{t-1})^{x_i^{t-1}} = \prod_{k=0}^{t-1} (C_k)^{x_i^k}$$

for every $i = 1, \ldots, n$. If the participants follow accurately the protocol, then we get similarly the following holds

$$\mathcal{F}(y_{j_v}) = \prod_{k=0}^{t-1} (C_k)^{x_{j_v}^k}$$

for every $v \in \{1, \ldots, t\}$. Suppose that $\mathcal{F}_b$ in (5) is used in the verification algorithm. Then, the participants can verify their shares as given in [31] by checking

$$\mathcal{F}(y_i) = \mathcal{F}(f(x_i)) = \sum_{k=0}^{t-1} \mathcal{F}_b(a_k) x_i^k,$$

where $\mathcal{F}_b(a_k)$ is the public commitment of the secret value a_k for $k = 0, \ldots, t-1$. The proof is then completed. □

As a result of Theorems 1 and 2, we conclude that the proposed VMSS is a (post-quantum) secure scheme against attackers and malicious users (that is, it resists both active and passive attacks).

5. DSB Based on the Proposed VMSS Scheme

In this section, we first incorporate the proposed threshold-based VMSS scheme into the original DSB system to distribute privately transaction data. We then describe the distribution and recovery processes of data at each block.

In the DSB system, each data block is stored in certain subsets of the set of all nodes by distributing it among nodes in each subset. Assume that the set $\mathcal{P} = \{P_1, \ldots, P_n\}$ of n nodes in a blockchain network is divided into $\frac{n}{m}$ distinct subsets in $\mathcal{A} = \{A_1, \ldots, A_{\frac{n}{m}}\}$ and each subset has m participants. Assume that each subset A_l has the proposed (m, t_l)-threshold VMSS scheme to share simultaneously the global secret s_0 and local secret s_l for $l = 1, 2, \ldots, \frac{n}{m}$. In the (m, t_l)-threshold VMSS scheme of the subset A_l, $\{P_{l,1}, \ldots, P_{l,m}\}$ is a set of m participants, D_l is the dealer and t_l is its own independent threshold for $l = 1, 2, \ldots, \frac{n}{m}$. We now incorporate the proposed VMSS scheme into the framework of the DSB system in Algorithm 2.

Remark 2. *In Algorithm 2, we assume $t_0 \leq t_l$ for all $l = 1, \ldots, \frac{n}{m}$. Depending upon the applications of the blockchain systems, we may assume $t_l \leq t_0$ for some $l = 1, \ldots, \frac{n}{m}$. In this case, to recover the global secret s_0, we need at least $(t_0 - t_l)$ more participants, who may be selected among the rest of the participants of the corresponding set or from the other subsets. For example, it may be assumed that each subset should collaborate to reconstruct the global secret s_0.*

5.1. Storing Data Block

We here describe how to distribute and store transaction data at each block. To distribute transaction data at each block, we first encrypt it by AES-256 for its confidentiality, then share privately the hash value of the block by the proposed scheme for its integrity, and finally encode by Reed–Solomon code.

Each subset A_l for $l = 1, \ldots, \frac{n}{m}$ follows the following processes to distribute and store data at each block. Each subset A_l has the same data block $B^{(t)}$ and the same hash value $W^{(t)}$ of the t-th block. Assume that $W^{(t)} \in \mathbb{F}_p$ and $B^{(t)} \in \mathbb{F}_q$, where p is a prime whose size about 256 bit-length and q is an extremely large prime. Each A_l first generates the secret key $K_l^{(t)} \in \mathbb{F}_p$ and then encrypts the data block $B^{(t)}$ with the AES-256 symmetric key encryption algorithm using $K_l^{(t)}$ as $m_l^{(t)} = E_{K_l^{(t)}}(B^{(t)})$. Here, the secret key $K_l^{(t)}$ (the local secret) and the hash value $W^{(t)}$ (the global secret) are simultaneously shared among m nodes in A_l by the proposed (m, t_l)-threshold VMSS scheme, introduced in Algorithm 2. Thereafter, the encrypted data $m_l^{(t)}$ is encoded into $c_l^{(t)}$ by Reed–Solomon code $RS(m, t_l)$ before distributing it among m nodes in A_l. We note that RS-code is an example of non-trivial MDS codes [27] (Chapter 11) and this coding process decreases the recovery communication cost and enhances the robustness. Finally, the encoded data $c_l^{(t)}$ are distributed to each node in A_l so that any t_l authorized nodes in A_l can reconstruct it in the recovery phase. The DSB with the proposed VMSS scheme is summarized in Algorithm 3.

Algorithm 2 The proposed (m, t_l)-threshold VMSS scheme for DSB

Input. Given a partition $\mathcal{A} = \{A_1, \ldots, A_{\frac{n}{m}}\}$ and threshold parameters t_0, t_l with $2 \leq t_0 \leq t_l \leq m$

1: Set a global secret $s_0 \in \mathbb{F}_p^*$, commit $C_0 = \mathcal{F}(s_0)$ and publish C_0.
2: Generate random elements $a_k \in \mathbb{F}_p$, commit $C_k = \mathcal{F}(a_k)$ and publish C_k for $k = 1, \ldots, t_0 - 1$.
3: Construct a global $(t_0 - 1)$-degree polynomial $f(x) = s_0 + a_1 x + \cdots + a_{t_0-1} x^{t_0-1}$.
4: **for** $l = 1$ to $\frac{n}{m}$ **do**
5: **Construction in** A_l**:** D_l performs the following steps.
6: **for** $i = 1$ to m **do**
7: Select distinct elements $x_{l,i} \in \mathbb{F}_p^*$ and evaluate the share $y_{l,i} = f(x_{l,i})$.
8: Encrypt $y_{l,i}$ as $c_{l,i} = \mathcal{E}_{K_{l,i}}(y_{l,i})$ by using the public key $K_{l,i}$ of $P_{l,i}$.
9: Send the pair $(x_{l,i}, c_{l,i})$ to $P_{l,i}$ in the public channel.
10: **end for**
11: Set a local secret $s_l \in \mathbb{F}_p^*$.
12: Construct a local m-degree polynomial $f_l(x)$ defined as in (1) for the secret s_l.
13: **for** $v = m + 1$ to $2m - t_l + 1$ **do**
14: Select distinct elements $x_{l,v} \in \mathbb{F}_p^*$, and evaluate $f_l(x_{l,v})$.
15: Broadcast the point $(x_{l,v}, f_l(x_{l,v}))$.
16: **end for**
17: **Verification in** A_l**:** $P_{l,i}$ performs the following steps.
18: **for** $i = 1$ to m **do**
19: Decrypt $c_{l,i}$ as $y_{l,i} = \mathcal{D}_{K'_{l,i}}(c_{l,i})$ by using own private key $K'_{l,i}$.
20: Verify $\mathcal{F}(y_{l,i}) = \prod_{k=0}^{t_0-1} (C_k)^{x_{l,i}^k}$.
21: **end for**
22: **Recovery in** A_l **with** (m, t_l)**-threshold VMSS scheme**
23: **for** $v = 1$ to t_l **do**
24: P_{l,j_v} encrypts y_{l,j_v} as $z_{l,j_v} = \mathcal{E}_{K_D}(y_{l,j_v})$ by using the public key K_{D_l} of D_l.
25: P_{l,j_v} sends the pair (x_{l,j_v}, z_{l,j_v}) to D_l in the public channel.
26: D_l decrypts z_{l,j_v} as $y_{l,j_v} = \mathcal{D}_{K'_D}(z_{l,j_v})$ by using own private key K'_{D_l}.
27: D_l verifies the validity of each y_{l,j_v} as in Step 20.
28: **if** y_{l,j_v} is valid **then** accept the points (x_{l,j_v}, y_{l,j_v})
29: **else** reject
30: **end if**
31: **end for**
32: By using the verified private points (x_{l,j_v}, y_{l,j_v}) for $v \in \{1, \ldots, t_l\}$, the global secret s_0 can be cooperatively recovered from the formula in (3). Here, at least t_0 points can do it since $t_0 \leq t_l$.
33: By using the verified private points (x_{l,j_v}, y_{l,j_v}) for $v \in \{1, \ldots, t_l\}$ and the public points $(x_{l,v}, f_l(x_{l,v}))$ for $v \in \{m + 1, \ldots, 2m - t_l + 1\}$, the local secret s_l can be cooperatively recovered from the formula in (4).
34: **return** s_0 and s_l.
35: **end for**

Algorithm 3 DSB based on the proposed (m, t_l)-threshold VMSS

Input. Given a partition $\mathcal{A} = \{A_1, \ldots, A_{\frac{n}{m}}\}$

1: Set the hash value $W^{(t)} = s_0 \in \mathbb{F}_p$ as the global secret.
2: **for** $l = 1$ to $\frac{n}{m}$ **do**
3: Generate the secret key $K_l^{(t)} = s_l \in \mathbb{F}_p$ for the AES-256.
4: Encrypt $B^{(t)}$ with the AES-256 algorithm as $m_l^{(t)} = E_{K_l^{(t)}}(B^{(t)})$.
5: Share and store $K_l^{(t)}$ and $W^{(t)}$ among m nodes in A_l by (m, t_l)-threshold VMSS given in Algorithm 2.
6: Encode $m_l^{(t)}$ into $c_l^{(t)}$ by $RS(m, t_l)$.
7: Distribute and store $c_l^{(t)}$ among m nodes in A_l.
8: **end for**

5.2. Recovering Data Block

The recovering method is an inverse process that is performed in a backward and first-out manner. We below describe how to recover the shared data block at each block. For each subset A_l, where $l = 1, \ldots, \frac{n}{m}$, the following steps are performed to recover the shared data at each block. Each A_l first reconstructs the encoded data $c_l^{(t)}$ from the authorized t_l nodes, then $c_l^{(t)}$ is decoded by Reed–Solomon code $RS(m, t_l)$, and hence A_l gets the encrypted data $m_l^{(t)}$. Thereafter, the authorized t_l nodes from A_l can reach the hash value $s_0 = W^{(t)}$ and its secret key $s_l = K_l^{(t)}$ from the equation in (4). Next, A_l decrypts the encrypted data $m_l^{(t)}$ as $B_l^{(t)} = D_{K_l^{(t)}}(m^{(t)})$ with the AES-256 encryption algorithm using its secret key $K_l^{(t)}$, and hence gets the data block $B^{(t)}$ of the t-th block. Finally, A_l checks its integrity with the corresponding hash value $W^{(t)}$. Algorithm 4 formalizes how to recover the shared data block $B^{(t)}$ of the t-th block.

Algorithm 4 Recovery algorithm for $B^{(t)}$ in the DSB based on VMSS

Input. Given a partition $\mathcal{A} = \{A_1, \ldots, A_{\frac{n}{m}}\}$

1: **for** $l = 1$ to $\frac{n}{m}$ **do**
2: Concatenate $c_l^{(t)}$ from the authorized t_l nodes in A_l.
3: Decode $m_l^{(t)}$ from $c_l^{(t)}$ by $RS(m, t_l)$.
4: Reconstruct both $K_l^{(t)}$ and $W^{(t)}$ by the proposed (m, t_l)-threshold VMSS given in Alg. 2.
5: Decrypt $m_l^{(t)}$ with the AES-256 algorithm as $B^{(t)} = E_{K_l^{(t)}}(m_l^{(t)})$.
6: Check the integrity of $B^{(t)}$ with the hash value $W^{(t)}$.
7: **end for**

5.3. Costs Analysis and Robustness for DSB Based on VMSS

In this subsection, we present the storage and recovery communication costs, and also the robustness of the proposed DSB system. We note that they depend on the threshold parameter t_l of each subset A_l for $l = 1, 2, \ldots, \frac{n}{m}$.

We assume that the data block $B^{(t)}$ of the t-th block is stored in $\mathbb{F}_q$, and the secret key $K_l^{(t)}$ and the hash value $W^{(t)}$ are stored in $\mathbb{F}_p$. In real-world applications, q is an extremely large prime when comparing a prime p of size about 256 bit-length.

Storage cost. We compute the storage cost for each node at the t-th block. Algorithm 3 distributes and stores the data block $B^{(t)}$, hash value $W^{(t)}$ and secret key $K_l^{(t)}$ in each subset A_l. In Step 5, storing both $K_l^{(t)}$ and $W^{(t)}$ by the proposed (m, t_l)-threshold VMSS scheme has $\log_2 p$ storage cost for each node. In Step 7, the cost of storing $m_l^{(t)}$ encoding by $RS(m, t_l)$ to $c_l^{(t)}$ among m nodes is equal to $\frac{\log_2 q}{t_l}$

for each node. Hence, the storage cost S_{VMSS} of the proposed DSB based on VMSS for each node in A_l is equal to

$$S_{\text{VMSS}} = \frac{\log_2 q}{t_l} + \log_2 p$$

bit operations.

Recovery communication cost. We first recall that a single node failure can be easily tolerated by receiving the stored data from any node in the traditional blockchain. Moreover, a single node failure in each subset can be recovered by accessing all nodes and $(m-1)$ nodes in the same subset for the original DSB and the DSB based on LSS, respectively. In the proposed DSB based on (m, t_l)-threshold VMSS scheme, a single node failure in the subset A_l can be recovered by accessing only t_l nodes in A_l. Thus, the recovery communication cost C_{VMSS} of the proposed DSB based on VMSS is equal to

$$C_{\text{VMSS}} = t_l \cdot S_{\text{VMSS}} = \log_2 q + t_l \log_2 p$$

bit operations. Since $2 \leq t_l \leq m$, the recovery communication cost of the proposed system is much better than that of both the original DSB in [4] and the DSB based on LSS in [3].

Robustness to node failures. We deal with the robustness of the proposed DSB system. The robustness is defined as the maximum number of node failures which can be tolerated by the blockchain network. A single node failure in each subset leads to an effective failure of all m nodes in the original DSB while each subset can tolerate a single node failure in the DSB based on LSS. The proposed DSB based on VMSS can tolerate node failures up to $(m - t_l)$ due to the employed (m, t_l)-threshold VMSS scheme in A_l. This says that the proposed DSB recovers data block up to $(m - t_l + 1)\frac{n}{m} - 1$ node failures if $(m - t_l)$ nodes from one subset and $(m - t_l + 1)$ nodes from the others are failed, which implies that the robustness of the proposed DSB system is much better than that of both the original DSB in [4] and the DSB based on LSS in [3].

Remark 3. *In the proposed DSB system, if $(m - t_l + 1)$ nodes from each subset A_l are failed (indeed, there are totally $(m - t_l + 1)\frac{n}{m}$ node failures), then data block cannot be recovered.*

We summarize in Table 1 the comparison of the previous DSB systems and the proposed DSB system in terms of storage and recovery communication costs, as well as robustness.

Table 1. Comparison of storage cost (SC), recovery communication cost (RCC), and robustness.

Costs	Blockchain	DSB in [4]	DSB with LSS in [3]	DSB with Proposed VMSS
SC	$\log_2 q + \log_2 q$	$\frac{\log_2 q}{m} + 2\log_2 p$	$\frac{\log_2 q}{m-1} + \log_2 p$	$\frac{\log_2 q}{t_l} + \log_2 p$
RCC	$\log_2 q + \log_2 p$	$\log_2 q + 2m \log_2 p + \rho$	$\log_2 q + (m-1)\log_2 p$	$\log_2 q + t_l \log_2 p$
Robustness	$n - 1$	$\frac{n}{m} - 1$	$\frac{2n}{m} - 1$	$(m - t_l + 1)\frac{n}{m} - 1$

Remark 4. *On the internet of medical things (IoMT), it is proposed that blockchain may be used for the immutable storage of data in a decentralized way, see for instance [10,11]. This causes tremendous storage costs in the blockchain nodes. By using the proposed (m, t_l)-threshold VMSS scheme (Algorithm 3) for this case, one can reduce the storage cost by a factor t_l. In addition, the privacy of data does not depend on solely one node, but at least t_l nodes, that is, the data are leaked if at least t_l nodes are fraudulent. Furthermore, any malicious node sharing fake data in the threshold-based VMSS system can be identified by using either (2) or (6). As the medical records have a high level of privacy for their owner, it seems that the proposed methods in this paper would be a good candidate for practical applications.*

6. Concluding Remarks and Future Works

In this paper, we first proposed a secure threshold-based verifiable multi-secret sharing scheme without private channels, in which two secrets (secret key and the hash value of the block) are simultaneously shared in a single sharing process among nodes in a blockchain network. We then incorporate the proposed scheme into the distributed storage blockchain system to distribute and store privately the data block. We finally analyzed the storage and recovery communication costs and the robustness of the proposed DSB system. We observed that the proposed threshold scheme

reduces effectively the recovery communication cost and makes it more robust compared to the previous distributed storage blockchain systems. It also improves extremely the storage cost of the traditional blockchain systems. We note that the proposed scheme brings to the DSB system secure communication through the public channels and verification process based on the post-quantum lattice-based hard problems.

It is worth noting that the proposed threshold-based VMSS scheme can be applied in many practical systems such as decentralized mechanisms (authenticating an electronic voting protocol, an electronic funds transfer, etc.) and all types of distributed storage systems. Our prospects for this work can be listed as follows.

- The extensions of the proposed VMSS scheme to more general frameworks would be good future work.
- Another research problem is to find the best suitable post-quantum encryption and verification algorithms for the VMSS scheme that can improve the standard DSB systems. For example, to design a new VMSS scheme based on a code-based post-quantum verification algorithm and its integration into the DSB system would be interesting future works.
- The standard DSB systems consider the network coding, MDS codes, and LRCs to share storage among nodes in a blockchain network. It would be a nice future work finding a better code family to be used in the DSB systems so that it gives better storage and recovery communication costs.
- Deploying the proposed VMSS scheme into the real-world blockchain-based systems such as IoMT can be considered as a new research direction.

Author Contributions: Conceptualization, methodology, investigation, writing−original draft preparation, writing−review and editing: S.M., A.S. and O.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: We would like to thank the Associate Editor and the anonymous reviewers for their suggestions, which improved not only the readability but also the quality of the paper.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

AES	Advanced Encryption Standard
DLP	Discrete Logarithm Problem
DSB	Distributed Storage Blockchain
IFP	Integer Factorization Problem
LRC	Locally Recoverable Codes
LSS	Local Secret Sharing
MDPI	Multidisciplinary Digital Publishing Institute
MDS	Maximum Distance Separable
MSS	Multi-Secret Sharing
RS	Reed–Solomon
VMSS	Verifiable Multi-Secret Sharing
VSS	Verifiable Secret Sharing

References

1. Chen, H.; Wu, H.L.; Chang, C.C.; Chen, L.S. Light Repository Blockchain System with Multisecret Sharing for Industrial Big Data. *Secur. Commun. Netw.* **2019**, *2019*, 9060756. [CrossRef]
2. Dai, M.; Zhang, S.; Wang, H.; Jin, S. A low storage room requirement framework for distributed ledger in blockchain. *IEEE Access* **2018**, *6*, 22970–22975. [CrossRef]
3. Kim, Y.; Raman, R.K.; Kim, Y.S.; Varshney, L.R.; Shanbhag, N.R. Efficient local secret sharing for distributed blockchain systems. *IEEE Commun. Lett.* **2018**, *23*, 282–285. [CrossRef]

4. Raman, R.K.; Varshney, L.R. Distributed storage meets secret sharing on the blockchain. In Proceedings of the Information Theory and Applications Workshop (ITA), San Diego, CA, USA , 11–16 February 2018; pp. 1–6.
5. Raman, R.K.; Varshney, L.R. Dynamic distributed storage for blockchains. In Proceedings of the IEEE International Symposium on Information Theory (ISIT), Vail, CO, USA, 17–22 June 2018; pp. 2619–2623.
6. Jiang, S.; Cao, J.; McCann, J.A.; Yang, Y.; Liu, Y.; Wang, X.; Deng, Y. Privacy-Preserving and Efficient Multi-Keyword Search over Encrypted Data on Blockchain. In Proceedings of the 2019 IEEE International Conference on Blockchain, Atlanta, GA, USA, 14–17 July 2019; pp. 405–410.
7. Jiang, S.; Cao, J.; Wu, H.; Yang, Y.; Ma, M.; He, J. Blochie: A blockchain-based platform for healthcare information exchange. In Proceedings of the IEEE International Conference on Smart Computing (Smartcomp), Taormina, Italy, 18–20 June 2018; pp. 49–56.
8. Dey, T.; Jaiswal, S.; Sunderkrishnan, S.; Katre, N. HealthSense: A medical use case of Internet of Things and blockchain. In Proceedings of the International Conference on Intelligent Sustainable Systems (ICISS), Palladam, India, 7–8 December 2017; pp. 486–491.
9. Dilawar, N.; Rizwan, M.; Ahmad, F.; Akram, S. Blockchain: Securing Internet of Medical Things (IoMT). *Int. J. Adv. Comput. Sci. Appl.* **2019**, *10*, 82–89. [CrossRef]
10. Połap, D.; Srivastava, G.; Jolfaei, A.; Parizi, R.M. Blockchain Technology and Neural Networks for the Internet of Medical Things. In Proceedings of the IEEE Infocom 2020-IEEE Conference on Computer Communications Workshops (Infocom Wkshps), Toronto, ON, Canada, 6–9 July 2020; pp. 508–513.
11. Sharma, A.; Tomar, R.; Chilamkurti, N.; Kim, B.G. Blockchain based smart contracts for internet of medical things in e-healthcare. *Electronics* **2020**, *9*, 1609. [CrossRef]
12. Liu, Y.; Zhang, F.; Zhang, J. Attacks to some verifiable multi-secret sharing schemes and two improved schemes. *Inf. Sci.* **2016**, *329*, 524–539. [CrossRef]
13. Chen, D.; Lu, W.; Xing, W.; Wang, N. An efficient verifiable threshold Multi-Secret sharing scheme with different stages. *IEEE Access* **2019**, *7*, 107104–107110. [CrossRef]
14. Blakley, G.R. Safeguarding cryptographic keys. In Proceedings of the National Computer Conference, New York, NY, USA, 4–7 June 1979; Volume 48, pp. 313–317.
15. Shamir, A. How to share a secret. *Commun. ACM* **1979**, *22*, 612–613. [CrossRef]
16. Chor, B.; Goldwasser, S.; Micali, S.; Awerbuch, B. Verifiable secret sharing and achieving simultaneity in the presence of faults. In Proceedings of the 26th Annual Symposium on Foundations of Computer Science, Portland, OR, USA, 21–23 October 1985; pp. 383–395.
17. Amroudi, A.N.; Zaghain, A.; Sajadieh, M. A verifiable (k, n, m)-threshold multi-secret sharing scheme based on NTRU cryptosystem. *Wirel. Pers. Commun.* **2017**, *96*, 1393–1405. [CrossRef]
18. Ersoy, O.; Pedersen, T.B.; Kaya, K.; Selçuk, A.A.; Anarim, E. A CRT-based verifiable secret sharing scheme secure against unbounded adversaries. *Secur. Commun. Netw.* **2016**, *9*, 4416–4427. [CrossRef]
19. Feldman, P. A practical scheme for non-interactive verifiable secret sharing. In Proceedings of the 28th Annual Symposium on Foundations of Computer Science, Los Angeles, CA, USA, 12–14 October 1987; pp. 427–438.
20. Zhao, J.; Zhang, J.; Zhao, R. A practical verifiable multi-secret sharing scheme. *Comput. Stand. Interfaces* **2007**, *29*, 138–141. [CrossRef]
21. He, J.; Dawson, E. Multisecret-sharing scheme based on one-way function. *Electron. Lett.* **1995**, *31*, 93–95. [CrossRef]
22. Harn, L. Efficient sharing (broadcasting) of multiple secrets. *IEE Proc. Comput. Digit. Tech.* **1995**, *142*, 237–240. [CrossRef]
23. Hwang, R.J.; Chang, C.C. An on-line secret sharing scheme for multi-secrets. *Comput. Commun.* **1998**, *21*, 1170–1176. [CrossRef]
24. Rivest, R.L.; Shamir, A.; Adleman, L. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM* **1978**, *21*, 120–126. [CrossRef]
25. Diffie, W.; Hellman, M. New directions in cryptography. *IEEE Trans. Inf. Theory* **1976**, *22*, 644–654. [CrossRef]
26. Tamo, I.; Barg, A. A family of optimal locally recoverable codes. *IEEE Trans. Inf. Theory* **2014**, *60*, 4661–4676. [CrossRef]
27. MacWilliams, F.J.; Sloane, N.J.A. *The Theory of Error Correcting Codes*; Elsevier: Amsterdam, The Netherlands, 1977; Volume 16.

28. Dehkordi, M.H.; Ghasemi, R. A lightweight public verifiable multi secret sharing scheme using short integer solution. *Wirel. Pers. Commun.* **2016**, *91*, 1459–1469. [CrossRef]
29. El Bansarkhani, R.; Meziani, M. An efficient lattice-based secret sharing construction. In *IFIP International Workshop on Information Security Theory and Practice*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 160–168.
30. Pilaram, H.; Eghlidos, T. An efficient lattice based multi-stage secret sharing scheme. *IEEE Trans. Dependable Secur. Comput.* **2015**, *14*, 2–8. [CrossRef]
31. Rajabi, B.; Eslami, Z. A verifiable threshold secret sharing scheme based on lattices. *Inf. Sci.* **2019**, *501*, 655–661. [CrossRef]
32. Lyubashevsky, V.; Micciancio, D. Generalized compact knapsacks are collision resistant. In *International Colloquium on Automata, Languages, and Programming*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 144–155.
33. Chen, C.; Danba, O.; Hoffstein, J.; Hülsing, A.; Rijneveld, J.; Schanck, J.M.; Schwabe, P.; Whyte, W.; Zhang, Z. Algorithm Specifications And Supporting Documentation. *arXiv* **2019**, arXiv:1904.00357v1.