Research Article

Light Repository Blockchain System with Multisecret Sharing for Industrial Big Data

Hefeng Chen ¹, Hsiao-Ling Wu,² Chin-Chen Chang ³ and Long-Sheng Chen²

¹Computer Engineering College, Jimei University, Xiamen 361021, China

²Department of Information Management, Chaoyang University of Technology, Taichung 413, Taiwan

³Department of Information Engineering and Computer Science, Feng Chia University, Taichung 407, Taiwan

Correspondence should be addressed to Chin-Chen Chang; alan3c@gmail.com

Received 1 March 2019; Accepted 4 August 2019; Published 16 October 2019

Guest Editor: Fagen Li

Copyright © 2019 Hefeng Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Blockchain is an emerging technology that promises many exciting applications in various fields, including financial, medical, energy, and logistics management. However, there are still some limitations in the existing blockchain framework that prevents its widespread adoption in the commercial world. One important limitation is the storage requirement, wherein each blockchain node has to store a copy of the distributed ledger. Thus, as the number of transactions increases, this storage requirement grows quadratically, eventually limiting the scalability of a blockchain system. Moreover, the public ledger in a blockchain framework allows anyone in the network to audit the transaction, which may not be favourable in some privacy-sensitive applications. In this paper, a secret-sharing scheme is proposed to reduce the size of the blockchain transactions. Each transaction block is divided into t parts, and the size of each part is $1/t$ size of transaction block. We use the secret-sharing mechanism to share t parts into n shares. Hence, each node stores not one transaction but one share in the blockchain system. The proposed scheme can eventually reduce the storage cost of a blockchain transaction by $1/t$ without introducing an additional recovery communication cost; however, robustness is reduced in node failure as a tradeoff. Meanwhile, the proposed scheme was more efficient and secure compared to other state-of-the-art schemes that aim to reduce blockchain storage for industrial big data.

1. Introduction

Internet commerce has become relatively entirely dependent on financial institutions that are trusted third parties to handle electronic payments. Although the system is suitable for most transactions, it still has the inherent weaknesses in trust-based models. Blockchain is a paradigm-shifting technology that has emerged over the past decade, which is based on peer-to-peer communication technology, network theory, and cryptography [1]. In a traditional centralized database model, the central server stores every transaction record, which can be disastrous if the central server is being attacked. In contrast, blockchain technology is a distributed solution without any third-party trust problem. Since every node in the blockchain network keeps a copy of the public ledger, it is possible to audit the transactions locally without referring to a centralized authority. Therefore, blockchain can also be viewed as a distributed ledger system, which

eliminates the disadvantage of a traditional centralized database model. However, the storage cost in blockchain is huge [2], which is one of the factors currently limiting the widespread adoption of blockchain technology. Since each node needs to store the public ledger locally, the storage cost of an entire blockchain network grows quadratically.

To resolve the storage issue in blockchain, Dai et al. [3] proposed a low-storage blockchain system that employs network coding theory to divide the transaction data into multiple blocks and then stores the blocks at different nodes. This can realize distributed storage (DS) by recovering the transaction data through network coding (NC). They proposed two kinds of NC-based DS, one is deterministic rate (NC-DRDS) and the other is rateless (NC-RLDS), to deal with a fixed and variable number of blockchain nodes, respectively. In another approach, Dorri et al. [4] proposed a memory-optimized and flexible blockchain (MOF-BC) that allows the user to summarize or remove part of the “aged”

blockchain transactions. This eventually reduces the ability to perform a public audit, as some of the information is eventually erased. The idea of distributed storage blockchain (DSB) is first introduced by Raman and Varshney in 2018 [5]. The blockchain transaction block is first encrypted and then stored at different nodes (together with the encryption key) using a secret-sharing scheme. In this way, the storage of a blockchain is reduced significantly. Later, Kim et al. [6] improved DSB by proposing a local secret-sharing (LSS) scheme, which can tolerate single peer failure.

Secret sharing is used as a basic cryptographic primitive in computer science, including for electronic voting [7], key-aggregate authentication [8], distributed cloud computing [9], secret image sharing [10, 11], and data hiding [12]. In 1979, the secret sharing was first introduced by Shamir [13] based on the Lagrange interpolating polynomial, while Blakley [14] independently took another approach based on the hyperplane geometry. In 1983, Mignotte's scheme [15] and Asmuth-Bloom's scheme [16] were proposed based on the Chinese remainder theorem. A perfect (t, n) secret-sharing scheme [13] has two properties: (1) any t or more shares can recover the secret; (2) any $t - 1$ or fewer shares reveal no information about the secret. To share another secret, the dealer must redistribute every participant's secret share. Later, several schemes have been presented for multiple secret sharing. In a multisecret sharing scheme, every user only needs to keep one share and many secrets can be shared independently without updating the share.

Most multisecret sharing schemes require the disclosure of large amounts of public information. Chien et al. [17] proposed a multisecret scheme based on systematic block codes, while Yang et al. [18] improved the amount of public information required by Chien et al. Further, Yang's scheme is based on Shamir's secret-sharing scheme such that fewer than the threshold number of pieces do not leak any information. Dehkordi and Mashhadi's scheme [19] focused on improving the efficiency of computations involved in share creation and secret reconstruction rather than space efficiency. Our multisecret sharing is based on recursion, in which a secret is first divided into t pieces and then the pieces are encoded one by one in such a manner that the shares of the already encoded pieces are reused to create new shares for the next piece.

In this paper, we improve existing work [3, 5, 6] by proposing a low-storage scheme with multisecret sharing based on polynomial interpolation. Our idea is different from Raman and Varshney [5] and Kim et al. [6] as we do not encrypt the blockchain transaction block. Instead, our proposed scheme divides the blockchain transaction block directly into multiple shares and stores it in different nodes. The advantages of our work can be summarized as follows.

- (1) Efficiency: the size of share is on the order of $|S|/t$; that is, the storage room and communication cost are reduced effectively.
- (2) Robustness: no side information needs to be stored with the shares, and no public information needs to be disclosed.

2. Blockchain and Storage Issues

Blockchain is a distributed ledger system that requires each participating node to store a copy of ledger for the transaction records. Every time a transaction block is created, it is first verified by the neighbouring nodes and then goes through the consensus (mining) process by solving a difficult cryptographic puzzle. One of the nodes that successfully solve this puzzle is rewarded with some incentives. This transaction block is then added to the ledger, wherein each block is related to the previous block through a chain of hash values. This data structure suffers from scalability issue, as the storage room required to keep the entire ledger is growing quadratically when the number of blockchain nodes increases.

3. Overview of Shamir's Secret-Sharing Scheme

3.1. Shamir's (r, n) -Threshold Secret Sharing. In 1979, Shamir developed a (r, n) -threshold secret-sharing scheme [13] based on polynomial interpolation, wherein a univariate polynomial $y = f(x)$ of degree $r - 1$ is uniquely defined by r points (x_j, y_j) with distinct x_j , for $j = 1, 2, \dots, r$. The scheme can decompose one secret into n shares, with r shares required to recover the original secret, where $r \leq n$. However, the secret cannot be recovered with less than r shadows. It consists of the following two phases.

3.2. Shadow Distribution Phase. The trusted dealer starts with a secret integer, $S \geq 0$, that is, to be distributed among n users. Thus, the dealer runs the following:

- (1) Choose a prime $p > \max(n, S)$.
- (2) Randomly select $r - 1$ independent coefficients $a_1, a_2, \dots, a_{r-1} \in \mathbb{Z}_p$, to constitute a random polynomial with $r - 1$ degree over $\mathbb{Z}_p$:

$$f(x) = S + \sum_{j=1}^{r-1} a_j x^j \mod p. \quad (1)$$

- (3) Choose n distinct nonzero elements of $\mathbb{Z}_p$, denoted as x_i , for $1 \leq i \leq n$.
- (4) Compute $s_i = f(x_i) \mod p$, $1 \leq i \leq n$ and securely transfer the share s_i to user U_i , along with the public index x_i .

3.3. Secret Reconstruction Phase. Assume that r users, $U_{i_1}, U_{i_2}, \dots, U_{i_r}$, pool their shares to compute the secret S . Their shadows provide r distinct points (x_{i_j}, s_{i_j}) 's, $1 \leq j \leq r$, which allow the computation of the coefficients of $f(x)$ by Lagrange interpolation. The secret, S , can be expressed as

$$S = f(0) = \sum_{j=1}^r s_{i_j} c_{i_j} \mod p, \quad (2)$$

where $c_{i_j} = \prod_{1 \leq k \leq r, k \neq j} (x_{i_j}/x_{i_k} - x_{i_j}) \mod p$, for $1 \leq j \leq r$.

Figure 1 shows an example of Shamir's (3, 6) secret sharing, which can be adopted by blockchain. Assume that the transaction block α is divided into three equal length packets, which are expressed as integers α_1 , α_2 , and α_3 . We use them as the coefficients to construct a 2-degree polynomial, which is expressed as $f(x) = \alpha_1 + \alpha_2x + \alpha_3x^2$. The shadows are then generated and distributed to the blockchain nodes with ID = 1, 2, 3, 4, 5, 6.

4. Blockchain Transactions with Secret-Sharing Scheme

In this section, we describe our proposal of using secret-sharing scheme to reduce the storage cost of blockchain. In our sharing scheme, each block is stored in a distributed manner among all nodes. Consider the transaction block α , our idea inspired by Parakh and Kak [20]. The flowchart of sharing process and reconstruction process is shown in Figure 2.

Firstly, we divide the block α into t pieces of size $|\alpha|/t$, denoted as $\alpha_1, \alpha_2, \dots, \alpha_t$, such that their concatenation $\alpha_1\|\alpha_2\|\dots\|\alpha_t = \alpha$. All the arithmetic performed is finite field arithmetic on $\mathbb{Z}_p$, where p is larger than α_i ($1 \leq i \leq t$) and n . Then, we share them recursively as follows. The first step is to run Shamir's (2, 2) secret sharing for the piece α_1 . That is, generate randomly a 1-degree polynomial $g_1(x)$ whose constant term is α_1 , and compute $g_{1,1} = g_1(1)$ and $g_{1,2} = g_1(2)$ to obtain two shares for α_1 . Next, generate a 2-degree polynomial $g_2(x) = \alpha_2 + g_{1,1}x + g_{1,2}x^2 \pmod{p}$ whose constant term is the next piece α_2 , and the coefficients are the previous two shares. By induction, assume that we have obtained i shares of the piece α_{i-1} ($2 \leq i \leq t-1$), denoted as $g_{i-1,1}, g_{i-1,2}, \dots, g_{i-1,i}$, we can generate a i -degree sharing polynomial for the next piece α_i by using $g_{i-1,j}$ as the j -th coefficient term for $j = 1, 2, \dots, i$. At the last step, we generate a t -degree sharing polynomial:

$$g_t(x) = \alpha_t + g_{t-1,1}x + g_{t-1,2}x^2 + \dots + g_{t-1,t}x^t \pmod{p}, \quad (3)$$

and the final n shares are

$$g_t(x_1), g_t(x_2), \dots, g_t(x_n), \quad (4)$$

where $x_1, x_2, \dots, x_n$ are n public indexes of nodes. The sharing scheme is shown in Algorithm 1.

The reconstruction is an inverse process that is carried out in a backward and first-out manner. Any $t+1$ of the shares can interpolate a t -degree polynomial $g_t(\cdot)$ with the constant term α_t . Then, interpolate a $t-1$ -degree polynomial $g_{t-1}(\cdot)$ with the constant term α_{t-1} , by taking the j -th coefficient of $g_t(\cdot)$ as the point at $x = j$, for $j = 1, 2, \dots, t$. The recursion repeats until α_1 is obtained. Then, the whole block is given by the concatenation $\alpha_1\|\alpha_2\|\dots\|\alpha_t = \alpha$. Algorithm 2 shows how to reconstruct the shared transaction block.

In Figure 3, a block α is divided into 4 pieces $\alpha_1 = 19$, $\alpha_2 = 27$, $\alpha_3 = 4$, and $\alpha_4 = 15$. Assumed that α is to be shared between 7 parties such that any 5 of them can reconstruct all the 4 secrets. We can now use a prime $p = 31$ and random number $r = 11$.

FIGURE 1: Optional (3, 6) Shamir secret sharing.

Theorem 1. Our proposed scheme divides the original file α into n pieces in such a way that (i) any $t+1$ packets can recover the original file α , but (ii) no group of t pieces can do so.

Proof. Based on the Lagrange interpolation, given t points $(u_1, y_1), (u_2, y_2), \dots, (u_{t+1}, y_{t+1})$ with distinct u_i 's, there is a unique polynomial of degree t such that $y_{ij} = g(u_{ij})$, for $1 \leq j \leq t+1$, i.e.,

$$g(x) = \sum_{j=1}^{t+1} y_{ij} l_{ij}(x) \pmod{p}, \quad (5)$$

where $l_{ij}(x) = \prod_{1 \leq k \leq t+1, k \neq j} (x - u_{ik} / u_{ij} - u_{ik}) \pmod{p}$.

Therefore, $\alpha_t = g(0)$ can be reconstructed correctly. To reconstruct the next piece α_{t-1} , a $(t-1)$ -degree polynomial is generated by taking the k -th coefficient of $g(x)$ as the point at $x = k$, for $k = 1, 2, \dots, t$. Correctness is still guaranteed by Lagrange interpolation. Then, based on recursion, $\alpha_{t-2}, \dots, \alpha_2, \alpha_1$ can be correctly reconstructed one by one.

On the contrary, if only t of these n packets are available, it is not suffice to compute α_i 's. For each candidate value $\alpha_t \in \{0, 1, \dots, p-1\}$, we can construct a unique polynomial $g(x)$ of degree t such that $g(0) = \alpha_t \pmod{p}$ and $g(x_{ij}) = y_{ij} \pmod{p}$ for $1 \leq j \leq t+1$. These p possible polynomials have equal probability; therefore, the real value of packets cannot be determined.

Moreover, we can improve the above proposed scheme to achieve the verifiable property. In general, each transaction block in the blockchain system (e.g., Bitcoin) consists of a block header and a data block. All transactions are included in the data block as leaf nodes of a Merkle tree [21], and the hash root of the Merkle tree is computed in the block header (see Figure 4). Denote the block header and data block of a transaction block, α , as β and δ , respectively. Now, we perform the above sharing scheme for β and δ , independently. After reconstructing β and δ , the correctness of δ can be verified by the root hash value in block header β . $\square$

FIGURE 2: Flowchart of sharing process and reconstruction process.

Given the block α and n public indexes $\{x_1, x_2, \dots, x_n\}$
 Divide α into t equal length pieces $\alpha_1, \alpha_2, \dots, \alpha_t$
 Select a large prime $p > \max\{\alpha_1, \alpha_2, \dots, \alpha_t, n\}$
 Select a random number $r \in \mathbb{Z}_p$, and construct a 1-degree polynomial $g_1(x) = \alpha_1 + rx \pmod{p}$
 Compute $g_{1,1} = g_1(1)$ and $g_{1,2} = g_1(2)$
for $i = 2$ to $t - 1$ **do**
 Construct i -degree polynomial $g_i(x) = \alpha_i + g_{i-1,1}x + g_{i-1,2}x^2 + \dots + g_{i-1,i}x^i \pmod{p}$
 for $j = 1$ to $i + 1$ **do**
 Compute $g_{i,j} = g_i(j)$
 end for
end for
 Construct t -degree polynomial $g_t(x) = \alpha_t + g_{t-1,1}x + g_{t-1,2}x^2 + \dots + g_{t-1,t}x^t \pmod{p}$
for $i = 1$ to n **do**
 Compute $\beta_i = g_t(x_i)$
end for
 Distribute the share β_i to the corresponding node with public index x_i , for $i = 1, \dots, n$

ALGORITHM 1: Sharing process.

Given any $t + 1$ shares $\beta_{x_1}, \beta_{x_2}, \dots, \beta_{x_{t+1}}$
 Interpolate $t + 1$ points, $(x_1, \beta_{x_1}), (x_2, \beta_{x_2}), \dots, (x_{t+1}, \beta_{x_{t+1}})$, to generate t -degree polynomial $g_t(x) = \alpha_t + g_{t-1,1}x + g_{t-1,2}x^2 + \dots + g_{t-1,t}x^t \pmod{p}$
 Compute the constant term $\alpha_t = g_t(0)$ and the coefficients
for $i = t - 1$ to 1 **do**
 Interpolate $i + 1$ points $(1, g_{i,1}), (2, g_{i,2}), \dots, (i + 1, g_{i,i+1})$ to generate i -degree polynomial $g_i(x) = \alpha_i + g_{i-1,1}x + g_{i-1,2}x^2 + \dots + g_{i-1,i}x^i \pmod{p}$
 Compute $\alpha_i = g_i(0) \pmod{p}$
end for
 return $\alpha = \alpha_1 \parallel \alpha_2 \parallel \dots \parallel \alpha_t$

ALGORITHM 2: Reconstruction process.

5. Practical Implementation Issues

5.1. Storage Cost. Assuming that the block header is stored in $\mathbb{F}_\eta$ and the data block is stored in $\mathbb{F}_q$, then the blockchain's storage cost per transaction per peer is

$$\log_2 \eta + \log_2 q. \quad (6)$$

The storage cost of the DSB proposed by Raman and Varshney [5] is

FIGURE 3: Example of proposed (5, 7) secret sharing where the block $\alpha = 19270415$, the prime modules $p = 31$, and the random number $r = 11$.

FIGURE 4: Transactions hashed in a Merkle tree.

$$\frac{\log_2 \eta}{t+1} + 2 \log_2 q. \quad (7)$$

This is effectively smaller compared to the traditional blockchain. Kim et al. [6] proposed the LSS scheme to further reduce the storage cost to

$$\frac{\log_2 \eta}{t} + \log_2 q. \quad (8)$$

The comparison of storage cost for different schemes is detailed in Table 1.

Compared to the traditional blockchain, our proposed scheme can effectively reduce the storage cost by $1/t$, so the storage cost becomes $(\log_2 \eta + \log_2 q)/t$. From Table 1, it is clear that our proposed scheme is more efficient than other prior work including Kim et al.'s [6]. This can effectively reduce the deployment and maintenance cost in blockchain.

5.2. Recovery Communication Cost. Our proposed scheme can effectively reduce the storage cost in blockchain

communication, but it also introduces some additional communication cost. In traditional blockchain, each node keeps a public ledger locally. If any node failure happens, the failed node can recover the ledger by getting it from other peers, so the communication cost is proportional to the storage cost. The minimum communication cost is essentially

$$\log_2 \eta + \log_2 q. \quad (9)$$

When there is a node failure, our proposed scheme requires $(t+1)$ secret shares to recover the transaction block, so the recovery communication cost is dependent on the number of shares $(t+1)$ and proportional to the storage cost:

$$\frac{(t+1)(\log_2 \eta + \log_2 q)}{t+1} = \log_2 \eta + \log_2 q. \quad (10)$$

Figure 5 shows the tradeoff between storage and recovery communication cost for a specific case. It is clearly shown that our proposed scheme can achieve lower storage cost but

TABLE 1: Comparison of storage/communication costs and robustness.

	Traditional blockchain	Original DSB [5]	DSB with LSS [6]	Proposed scheme
Storage cost	$\log_2 \eta + \log_2 q$	$(\log_2 \eta/t + 1) + 2\log_2 q$	$(\log_2 \eta/t) + \log_2 q$	$(\log_2 \eta + \log_2 q)/t$
Communication cost	$\log_2 \eta + \log_2 q$	$\log_2 \eta + 2(t+1)\log_2 q + \rho$	$\log_2 \eta + t\log_2 q$	$\log_2 \eta + \log_2 q$
Robustness	$n - 1$	$(n/t + 1) - 1$	$(2n/t + 1) - 1$	$n - t - 1$

FIGURE 5: Tradeoff between storage cost and recovery communication cost ($\eta = 2^{350}$, $q = 2^{30}$, and $\rho = 200$).

does not increase the recovery communication cost. This is a huge advantage compared to original blockchain and the state-of-the-art work by Kim et al.

5.3. Robustness to Peer Failures. In regard to the robustness to peer failures, traditional blockchain can tolerate $(n - 1)$ node failure, in the expense of large ledger stored in every node. The scheme proposed by Raman and Varshney [5] and Kim et al. [6] can only tolerate $(n/t + 1) - 1$ and $2(n/t + 1) - 1$ failures, respectively, which is significantly smaller than the traditional blockchain. Our proposed scheme can tolerate $(n - t - 1)$ node failure, which is better compared to these two state-of-the-art schemes [5, 6]. Figure 6 shows the comparison of robustness where $n = 900$.

Another interesting work was presented by Dai et al. [3] recently based on network coding theory. Compared to them, our proposed scheme shares the same storage cost, communication cost, and robustness. However, their scheme can guarantee the recovery of transaction block if the collected packets is no less than t . It may leak information to malicious node if the collected packets is less than t . On the

contrary, our proposed scheme utilizes Shamir's secret-sharing scheme, which can assure that if the collected packets is less than r , no additional information is being leaked. Hence, it is more advantageous compared to the scheme proposed by Dai et al.

6. Conclusions

Traditional blockchain requires huge storage room when the number of nodes increases, which limits the scalability of this emerging technology. In this article, we proposed a low-storage scheme based on secret sharing to reduce the storage cost of blockchain. The proposed scheme is able to reduce the storage cost of traditional blockchain by $1/t$ without introducing additional communication cost. It is also more efficient compared to the other recent studies [5, 6] that attempted to solve the same problem. Although the scheme proposed by Dai et al. [3] is as efficient as our proposed scheme, it leaks partial information and is considered less secure. The proposed scheme requires both r and n to be fixed, so it is more suitable for private and consortium blockchain systems where the number of nodes is

FIGURE 6: Comparison of robustness ($n = 900$).

predetermined. This can be a limitation as the number of nodes in public blockchain is changing dynamically. In future, we plan to extend this to support public blockchain.

Data Availability

The relevant analysis data used to support the findings of this study are included in the article.

Conflicts of Interest

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

Acknowledgments

This work was partly supported by the National Natural Science Foundation of China (Grant no. 11804114), the Natural Science Foundation of Fujian Province, China (Grant nos. 2017J01761 and 2018J01537), the Science and Technology project of Fujian Province (Grant no. 2019H0021), and the Science and Technology project of Xiamen Municipal (Grant no. 3502Z20173028).

References

- [1] V. Sharma, "An energy-efficient transaction model for the blockchain-enabled Internet of Vehicles (IoV)," *IEEE Communications Letters*, vol. 23, no. 2, pp. 246–249, 2019.
- [2] K. Croman, C. Decker, I. Eyal et al., "On scaling decentralized blockchains," in *Financial Cryptography and Data Security*, pp. 106–125, Springer, Berlin, Germany, 2016.
- [3] M. Dai, S. Zhang, H. Wang, and S. Jin, "A low storage requirement framework for distributed ledger in blockchain," *IEEE Access*, vol. 6, pp. 22970–22975, 2018.
- [4] A. Dorri, S. S. Kanhere, and R. Jurdak, "MOF-BC: a memory optimized and flexible blockchain for large scale networks," *Future Generation Computer Systems*, vol. 92, pp. 357–373, 2019.
- [5] R. K. Raman and L. R. Varshney, "Distributed storage meets secret sharing on the blockchain," in *Proceedings of Information Theory and Applications Workshop (ITA)*, San Diego, CA, USA, February 2018.
- [6] Y. Kim, R. K. Raman, Y.-S. Kim, L. R. Varshney, and N. R. Shanbhag, "Efficient local secret sharing for distributed blockchain systems," *IEEE Communications Letters*, vol. 23, no. 2, pp. 282–285, 2018.
- [7] S. Iftene, "General secret sharing based on the Chinese remainder theorem with applications in E-voting," *Electronic Notes in Theoretical Computer Science*, vol. 186, pp. 67–84, 2007.
- [8] C. Guo, N. Luo, M. Z. A. Bhuiyan et al., "Key-aggregate authentication cryptosystem for data sharing in dynamic cloud storage," *Future Generation Computer Systems*, vol. 84, pp. 190–199, 2018.
- [9] H. Pirlam and T. Eghlidos, "An efficient lattice based multi-stage secret sharing scheme," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 1, pp. 2–8, 2015.
- [10] C. Guo, Q. Yuan, K. Lu, M. Li, and Z. Fu, " (t, n) Threshold secret image sharing scheme with adversary structure," *Multimedia Tools and Applications*, vol. 76, no. 20, pp. 21193–21210, 2017.
- [11] C. Guo, H. Zhang, Z. Fu, B. Feng, and M. Li, "A novel proactive secret image sharing scheme based on LISS," *Multimedia Tools and Applications*, vol. 77, no. 15, pp. 19569–19590, 2018.
- [12] P. Singh and B. Raman, "Reversible data hiding based on Shamir's secret sharing for color images over cloud," *Information Sciences*, vol. 422, pp. 77–97, 2018.
- [13] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.
- [14] G. R. Blakley, "Safeguarding cryptographic keys," in *Proceedings of the 1979 AFIPS National Computer Conference*, vol. 48, pp. 313–317, New York, NY, USA, June 1979.
- [15] M. Mignotte, "How to share a secret," in *Proceedings of the Workshop on Cryptography, EUROCRYPT*, pp. 371–375, Burg Feuerstein, Germany, April 1982.
- [16] C. Asmuth and J. Bloom, "A modular approach to key safeguarding," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 208–210, 1983.
- [17] H. Y. Chien, J. K. Jan, and Y. M. Tseng, "A practical (t, n) multi-secret sharing scheme," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E83-A, no. 12, pp. 2762–2765, 2000.
- [18] C.-C. Yang, T.-Y. Chang, and M.-S. Hwang, "A (t, n) multi-secret sharing scheme," *Applied Mathematics and Computation*, vol. 151, no. 2, pp. 483–490, 2004.
- [19] M. H. Dehkordi and S. Mashhadi, "New efficient and practical verifiable multi-secret sharing schemes," *Information Sciences*, vol. 178, no. 9, pp. 2262–2274, 2008.
- [20] A. Parakh and S. Kak, "Space efficient secret sharing for implicit data security," *Information Sciences*, vol. 181, no. 2, pp. 335–341, 2011.
- [21] S. Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System*, 2008, <https://bitcoin.org/en/bitcoin-paper>.