IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 25, 2018, accepted February 23, 2018, date of publication March 12, 2018, date of current version May 16, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2814624

A Low Storage Room Requirement Framework for Distributed Ledger in Blockchain

MINGJUN DAI[1,2], SHENGLI ZHANG[1,2], HUI WANG[1,2], AND SHI JIN[3]

[1]College of Information Engineering, Shenzhen University, Shenzhen 518060, China
[2]State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710126, China
[3]National Mobile Communications Research Laboratory, Southeast University, Nanjing 210018, China

Corresponding author: Shengli Zhang (zsl@szu.edu.cn)

This work was supported in part by the Natural Science Foundation of China under Grant 61771315 and Grant 61575126, in part by the Natural Science Foundation of Guangdong Province under Grant 2017A030310462, in part by the Key Project of Department of Education of Guangdong Province under Grant 2015KTSCX121, in part by the Natural Science Foundation of Shenzhen City under Grant JCYJ20160226192223251, and in part by the Natural Science Foundation of Shenzhen University under Grant 00002501.

ABSTRACT Traditional centralized commerce on the Internet relies on trusted third parties to process electronic payments. It suffers from the weakness of the trust-based model. A pure decentralized mechanism called blockchain tackles the above problem and has become a hot research area. However, since each node in a blockchain system needs to store all transactions of the other nodes, as time continues, the storage room required to store the entire blockchain will be huge. Therefore, the current storage mechanism needs to be revised to cater to the rapidly increasing need for storage. Network coded (NC) distributed storage (DS) can significantly reduce the required storage room. This paper proposes a NC-DS framework to store the blockchain and proposes corresponding solutions to apply the NC-DS to the blockchain systems. Analysis shows that the proposed scheme achieves significant improvement in saving storage room.

INDEX TERMS Blockchain, distributed storage, network coding.

I. INTRODUCTION

Commerce on the Internet has come to rely relatively exclusively on financial institutions serving as trusted third parties to process electronic payments. While the system works well enough for most transactions, it still suffers from the inherent weaknesses of the trust-based model [1]. The blockchain technique, which is a distributed solution of the trust problem without any third party, is a promising substitution [2]. In the following sections, we briefly illustrate the history of the technique development.

A. THIRD-PARTY DRAWBACKS

In recent years, the Internet-based economy has developed rapidly. In such Internet-based entities, there normally exists a certain trusted authority or third party (TP), which is actually a centralized agency. All online transactions go through this TP for safety. There are many forms of TP. It can be an email service provider indicating that email has been delivered; it can be a certification authority proving that a certain digital certificate is trustworthy; it can be a social network such as Facebook or WeChat indicating that posts regarding life events have been shared only with specified friends; it can be a bank verifying that money has been delivered reliably to family in a remote country; or it can be an electronic commerce platform such as Alibaba telling the e-commerce shop that a customer has paid for a product [3].

All the above situations count on normal operations of TP. Note that TP may have the following drawbacks [4]: (1) The data stored in TP may be damaged or altered, which may collapse the whole system. Actually, this situation is practically unavoidable for large and complex systems since small data manipulation may produce huge profits. (2) The existence of TP incurs construction and operational costs, which confines the application volume to be above a certain threshold. (3) The centric processing of large data is complicated and inefficient when all processes are managed by a data center.

B. BRIEF INTRODUCTION TO BLOCKCHAIN

In general, blockchain is a set of protocols/mechanisms that can help to verify and recognize each transaction by enabling all of the untrusted participants to come to a distributed consensus. Blockchain has the potential to revolutionize the world, especially the digital world, by realizing distributed

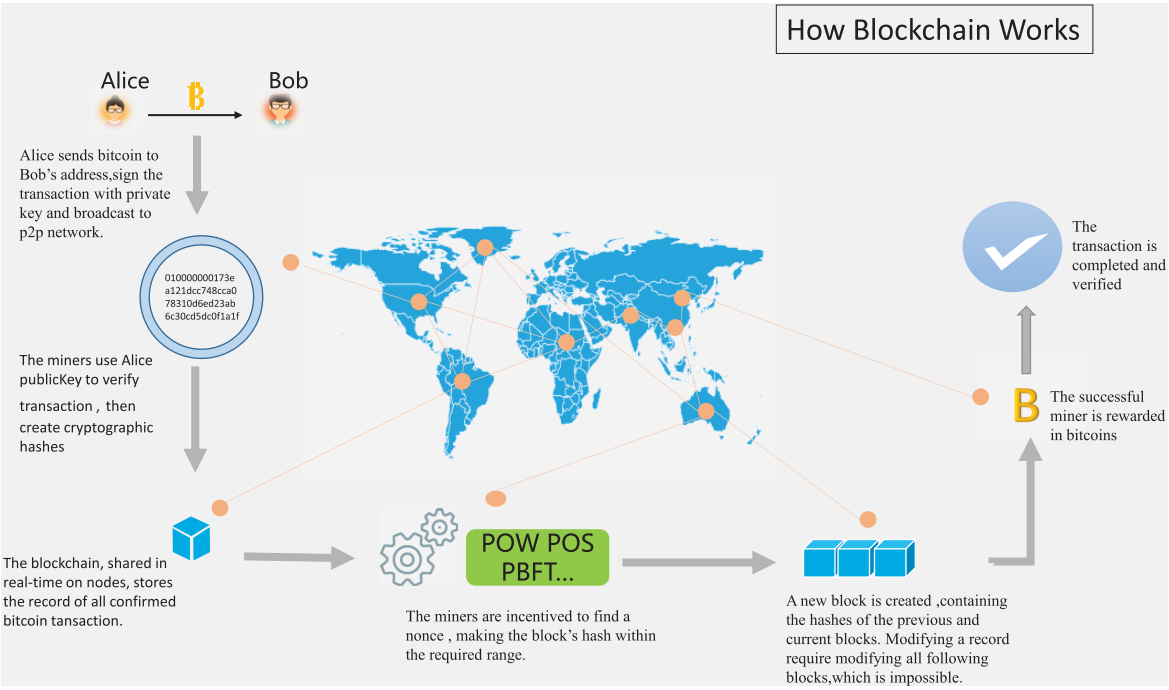

FIGURE 1. How blockchain works.

transactions, or a set of transactions, between unknown participants. The basic idea of blockchain can be explained by its two typical applications as follows.

Mechanism of blockchain in Bitcoin: Each transaction is broadcast to every node in the Bitcoin network and is then recorded in a public ledger after verification. Every single transaction needs to be verified for validity before it is recorded in the public ledger. A verifying node needs to ensure the following two things before recording any transaction [1]: (1) The spender owns the cryptocurrency-digital signature verification on the transaction. (2) The spender has sufficient cryptocurrency in his/her account, thus checking every transaction against the spender's account (public key) in the ledger to ensure that he/she has sufficient balance in his/her account. For an example of the procedure of blockchain in the Bitcoin network, please refer to Fig. 1.

Mechanism of blockchain in Ethereum: Ethereum is actually a decentralization application platform based on blockchain [2]. Compared to Bitcoin networks, Ethereum adds additional functions that the Bitcoin network lacks, so a developer can apply decentralization to blockchain. To implement decentralization, two fundamental basics are cryptography and game theory. Cryptography is easy to understand, and game theory is implemented as a means of achieving economic incentive. Another important concept in Ethereum is called a contract, which is actually one set of program code published/posted by a particular user. This code cannot be revised after publishing/posting. Each contract has an associated address, so when this address receives one transaction, its associated program code will be executed.

C. BLOAT PROBLEM IN BLOCKCHAIN

As described in the previous subsection, blockchain requires that each node stores all the transactions generated by all the nodes. Therefore, the volume of data stored at each node increases linearly with respect to n^2, where n denotes the number of nodes in the system. Moreover, it also increases linearly with time. The above two factors may cause the following problems: (1) a lack of storage room at each node, especially for portable devices [4], and (2) non-sustainable development of the system size, which may require much bandwidth for the initial synchronization of the new participant. The above two problems can be called the storage bloating problem. For example, currently, one entire blockchain in a Bitcoin network takes up 66 GB, and this number is continually increasing by 0.1 GB per day. It is estimated that one entire blockchain will take up 40 TB in 20 years, which is a tremendously heavy burden for either storage or downloading.

A number of attempts have been made to deal with this problem. One suggestion [1] is to reclaim disk space, namely, one node can delete old blocks to save storage room. Another suggestion is called the simplified payment verification (SPV) mechanism [5], also called a lightweight client. In SPV, a node does not store the entire blockchain; instead, it only stores all elementary information of each block,

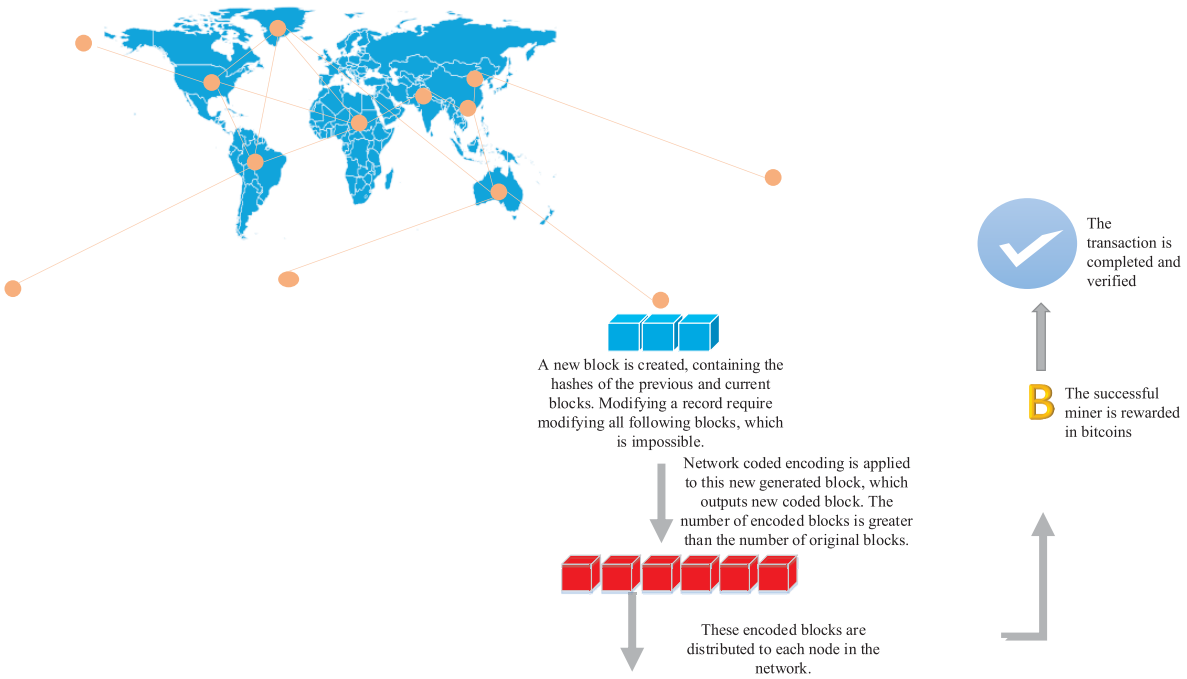

FIGURE 2. Incorporate network coded storage into blockchain systems.

including hash, proof of work, etc. In summary, the above works lose certain amounts of information.

As far as we know, there is no work tackling the potential storage bloating problem with network coding (NC) in which the framework does not lose any information. NC [6] applied to distributed storage (DS) [7], [8] can significantly reduce the storage space compared with replication. In this work, we suggest NC-based DS (NC-DS) to serve as the storage framework for blockchain to relieve the bloating problem. Correspondingly, within a NC-DS structured blockchain system, as there may frequently emerge new nodes that need to download the entire blockchain from existing nodes and then perform decoding to reconstruct the blockchain, low complexity decoding [11] needs to be considered to save both energy and time.

Existing works use the idea of blockchain to build a file storage system [12], [13]. However, no NC is applied; hence, the problem of bloating still remains.

II. PROPOSED NC-DS FRAMEWORK FOR BLOCKCHAIN

In this section, we propose a network coding-based distributed storage (NC-DS) framework for blockchain. This part is embeded into existing blockchain system as Fig. 2 illustrated. Compared to Fig. 1, only the process of inserted part is shown, and the remaining part is the same as that shown in Fig. 1 and hence are not drawn in this firgure. For the example in Fig. 2, we may adopt the $(6, 3)$ code illustrated in Fig. 3. In this example, the new generated block is partitioned into 3 equal-length sub-blocks, namely A, B, C as shown in blue color. These three sub-blocks are then

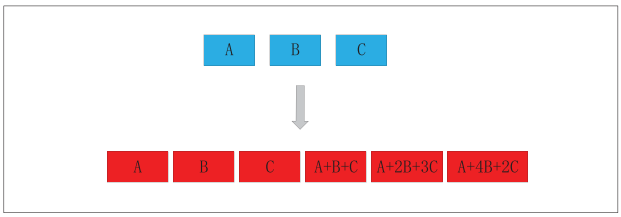

FIGURE 3. An optional $(6, 3)$ code for the red sub-blocks in Fig. 2.

encoded into 6 sub-blocks as shown in red color according to a particular $(6, 3)$ combination property (CP) code illustrated in Fig. 3. Detailed illustration of the $(6, 3)$ code may refer to subsec. II-A. Correspondingly, when the system fetches these encoded blocks, they need to perform decoding before normal blockchain procedure.

A. PRELIMINARIES ON NC-DS

Suppose one file takes up a storage room of G, where the file refers to the chain of the transaction blocks to be replicated to each participating node in the blockchain system. With respect to the storage of this file, there are two implementations of NC-DS. One is deterministic rate (DR) and the other is rateless (RL) [14]. We briefly illustrate the main idea of deterministic NC-DS (NC-DRDS) and rateless NC-DS (NC-RLDS).

1) BASICS OF ENCODING

Consider some verified blocks of a blockchain with size G, which is divided into k equal length packets that each take up

G/k storage room. We use $x_1, x_2, \ldots, x_k$ to denote these k packets. These k packets are encoded into n encoded packets which are denoted by $y_1, y_2, \ldots, y_n$. The rate of this code is k/n. Normally, there are two ways to compose the code, one is by linear combination and the other is by shift and add operation. For the first linear combination method, each encoded packet is a linear combination of $x_1, x_2, \ldots$, and x_k, namely,

$$y_i = \sum_{j=1}^{k} c_{i,j} x_k, \tag{1}$$

where $c_{i,j}$ is chosen from a finite field GF(q) with q being the size of this finite field. Detailed determination of $c_{i,j}$ will be illustrated later. For the second shift and add method, each encoded packet is obtained by first shifting x_i's to the right by distinct numbers of bits and then add them together in a bitwise manner [11].

2) NC-DRDS

If both k and n are fixed, the rate of this code is k/n, which is a deterministic number, hence the name. Choose one encoding method, say Reed-Solomn (RS) [15] codes, to encode these k packets, including $x_1, x_2, \ldots, x_k$, into n coded packets such that arbitrary k of these n packets are able to recover the original file. We call the code that possesses the above property as (n, k) combination property (CP) code. In this case, $c_{i,j}$'s are deterministic.

3) NC-RLDS

Different from the DR code, the number of coded packets, n, can be arbitrary for the k original packets with RL code. Therefore, the code rate of NC-RLDS can be changed as needed, hence the name of rateless code. In this case, n varies as needed. Each encoded packet is a random linear combination of $x_1, x_2, \ldots$, and x_k, namely, $c_{i,j}$ is randomly chosen from a finite field GF(q). This encoding method is called random linear network coding (RLNC) [16].

Since $c_{i,j}$ are chosen in a random manner, they may not be able to guarantee that arbitrary k of the n encoded packets are able to recover the original file. Therefore, more encoded packets should be utilized for recovering the original file. In other words, more than k encoded packets may need to be fetched for data reconstruction.

B. PROBLEMS OF TRIVIAL APPLICATION OF NC-DS TO BLOCKCHAIN

Let m denote the number of nodes that the blockchain system contains. If the value of m can be previously obtained, then the NC-DS technique can be readily applied to the blockchain system. We simply let $n \leq m$, choose an appropriate value k that is smaller than n, perform the encoding, and place the n encoded packets generated as described previously into these m nodes, with each node storing exactly one encoded packet. The storage room requirement for each node is G/k, which is $1/k$ that for simple replication adopted by the existing blockchain system. In some private blockchain systems, the upper bound of the system supporting users is given beforehand, where the DR scheme can be applied.

In the public blockchain system, due to its fully distributed nature, the value of m varies with time and cannot be obtained previously. In this case, NC-DRDS cannot be directly applied since we do not know how to set the value of n. On the contrary, NC-RLDS can still be directly applied since its rateless nature does not require knowing the value of m. However, trivial application of RLNC requires a large field size q to ensure high probability of recovery based on arbitrary k encoded packets. A large q indicates high encoding and decoding complexity [9], [10], which may not suit application scenarios where energy and computing capability are restricted, especially the scenario where mobile terminals serve as nodes. Therefore, low complexity design is desired [11].

C. PROPOSED SOLUTIONS

To apply NC-DRDS and NC-RLDS to blockchain, we propose some suggestions to tackle the problems listed above.

1) NC-DRDS

We may consider designing an encoding method that outputs the encoded packets sequentially instead of simultaneously. For example, we may first select a value for n that is greater than k. We encode the original k packets into n encoded packets, which are placed in n nodes. If $m - n > 0$, there are remaining nodes that have not had encoded packets placed in them. We perform another step of encoding to encode the original k packets into $2n$ encoded packets, which are then placed in $2n$ nodes within the remaining nodes. This process continues until all nodes have had one encoded packet placed in them.

2) NC-RLDS

The high decoding complexity of RLNC is mainly due to the following two reasons: (1) The operating finite field size q is large, which is CPU consuming especially for multiplication and division operations. (2) The decoding is actually equivalent to solving a set of linear equations. The main operations include the following two steps: In step 1, a set of elementary row operations is performed on the coefficient matrix to transform it into a row reduced echelon form (REF). In step 2, based on the REF resultant, backward substitution is applied to solve the original information. The complexities of these two steps are cubic and square with respect to the dimension of the coefficient matrix, respectively. In other words, step 1 dominates step 2 [11].

Corresponding to these two reasons, we design methods to achieve low decoding complexity. On the one hand, we restrict the operation to within a binary field to reap the lower decoding complexity within the operation finite field size. On the other hand, we try to avoid the time-consuming step 1 that dominates step 2 in complexity. In other words, the decoding is actually a backward substitution process.

To avoid step 1 and leave the backward substitution only, distinct packets should be misaligned by several bits when they are summed up for constructing an encoded packet. By combining the above two methods, we propose a method called binary field random shift encoding. Binary refers to all the operations that are within the binary field. Random shift means that to compose one encoded packet, all original packets first shift to the right by a random number of bits and are then summed in a bitwise manner.

III. ANALYSIS OF PROPOSED SCHEME

We analyze the proposed framework from the viewpoint of storage room, bandwidth, consensus speed, and scalability.

A. STORAGE ROOM

As previously described, our proposed NC-DRDS uses $1/k$ of the storage room used by existing blockchain systems.

With respect to our proposed NC-RLDS, especially the random shift method, the utilization of storage room is slightly larger than $1/k$ due to storage room overhead induced by shift.

B. BANDWIDTH CONSUMPTION FOR DATA DISSEMINATION WITHIN THE NETWORK

After the first user finishes his computation, he needs to disseminate his data to all nodes in the system. In the traditional blockchain system, the bandwidth consumption is n times that of the data volume of the disseminated data. For our proposed framework, the volume of data to be disseminated is $1/n$ of the original.

C. CONSENSUS SPEED

The processing order is as follows: After each new block reaches consensus, this block is then network coded and appended in each blockchain and stored using our proposed storage framework. Therefore, our proposed storage framework has no effect on consensus speed. For the case when there are multiple parallel sub-chains, we need to first perform decoding from those coded blocks and then perform consensus process. There is only certain additional decoding procedure.

D. SCALABILITY

Our proposed NC-DRDS and NC-RLDS can scale with respect to network size automatically. Firstly, for NC-DRDS, the encoded packets are output sequentially which suits scalability. Secondly, for NC-RLDS, the random shift also suits scalability since the scope of the number of shift can be adjusted and properly designed.

IV. CONCLUSIONS AND OPEN RESEARCH ISSUE

A. CONCLUSIONS

In this paper, we adopt a network coded (NC) distributed storage (DS) framework to store the blockchain to save storage room. Trivial application of NC-DS to blockchain is difficult because the parameter is difficult to set. We propose two solutions to tackle this problem. Analysis shows that the proposed scheme indeed achieves significant improvement in saving storage room.

B. ISSUES THAT MAY ARISE DUE TO THE NEW FRAMEWORK

In existing blockchain systems, hijackers may exist that want to modify the data stored within the system, also called a pollution attack. This may cause a data inconsistency problem, namely, the same previously stored data fetched from different nodes are different from each other. A key assumption of existing blockchain systems is that the hijackers are not able to compromise more than half of the computing resources of the system. Based on this assumption, if there is data inconsistency among different nodes, the majority rule handles this problem and always returns the correct data.

The above procedure does not readily apply to our proposed NC-DS framework since different nodes store distinct encoded packets. Note that an error in an NC-based system will spread to a large area. Therefore, the problem is how to mimic the majority rule to output correct data.

Works [17]–[19] focus on how to detect the pollution. Network error correcting codes mainly target correcting errors for NC-based systems [20], [21]. However, the above method is based on adding checking bits and incurs many checking operations. A small number of checking bits and simple checking operations are needed.

REFERENCES

[1] S. Nakamoto. (Oct. 2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. [Online]. Available: https://www.cryptovest.co.uk/

[2] C. Xu, K. Wang, and M. Guo, "Intelligent resource management in blockchain-based cloud datacenters," *IEEE Cloud Comput.*, vol. 4, no. 6, pp. 50–59, Nov. 2017.

[3] M. Crosby, P. P. Nachiappan, S. Verma, and V. Kalyanaraman, "BlockChain technology—Beyond bitcoin," SCET, Berkeley, CA, USA, Tech. Rep. Rev 2:6–19, Oct. 2015.

[4] K. Wang, J. Mi, C. Xu, Q. Zhu, L. Shu, and D. J. Deng, "Real-time load reduction in multimedia big data for mobile Internet," *ACM Trans. Multimedia Comput., Commun. Appl.*, vol. 12, no. 5, Oct. 2016, Art. no. 76.

[5] Null. (May 2017). *Scalability: Bitcoinwiki*. [Online]. Available: https://en.bitcoin.it/wiki/Scalability

[6] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.

[7] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *IEEE Trans. Inf. Theory*, vol. 56, no. 9, pp. 4539–4551, Sep. 2010.

[8] A. G. Dimakis, K. Ramchandran, Y. Wu, and C. Suh, "A survey on network codes for distributed storage," *Proc. IEEE*, vol. 99, no. 3, pp. 476–489, Mar. 2011.

[9] P. Vingelmann, P. Zanaty, F. H. P. Fitzek, and H. Charaf, "Implementation of random linear network coding on OpenGL-enabled graphics cards," in *Proc. IEEE EW*, Aalborg, Denmark, May 2009, pp. 1–5.

[10] M. Shahabinejad, M. Khabbazian, and M. Ardakani, "An efficient binary locally repairable code for Hadoop distributed file system," *IEEE Commun. Lett.*, vol. 18, no. 8, pp. 1287–1290, Aug. 2014.

[11] M. Dai, C. W. Sung, H. Wang, X. Gong, and Z. Lu, "A new zigzag-decodable code with efficient repair in wireless distributed storage," *IEEE Trans. Mobile Comput.*, vol. 16, no. 5, pp. 1218–1230, May 2017.

[12] S. Wilkinson, J. Lowry, and T. Boshevski. (Dec. 2014). *Metadisk A Blockchain-Based Decentralized File Storage Application*. [Online]. Available: https://www.metadisk:org

[13] S. Wilkinson, T. Boshevski, J. Brandoff, and V. Buterin, "Storj A peer-to-peer cloud storage network," Atlantic USA Inc, New York, NY, USA, Tech. Rep. Tech. Rep. V2.0, Dec. 2014.

[14] J. Castura and Y. Mao, "Rateless coding and relay networks," *IEEE Signal Process. Mag.*, vol. 24, no. 5, pp. 27–35, Sep. 2007.
[15] S. B. Wicker and V. K. Bhargava, *Reed-Solomon Codes and Their Applications*. Hoboken, NJ, USA: Wiley, 1994.
[16] T. Ho *et al.*, "A random linear network coding approach to multicast," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413–4430, Oct. 2006.
[17] C. Anglano, R. Gaeta, and M. Grangetto, "Securing coding-based cloud storage against pollution attacks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 5, pp. 1457–1469, May 2017.
[18] M. K. Kiskani and H. Sadjadpour. (2017). "Secure and private cloud storage systems with random linear fountain codes." [Online]. Available: https://arxiv.org/abs/1706.05604
[19] L. Buttyan, L. Czap, and I. Vajda, "Detection and recovery from pollution attacks in coding-based distributed storage schemes," *IEEE Trans. Depend. Sec. Comput.*, vol. 8, no. 6, pp. 824–838, Nov./Dec. 2011.
[20] X. Guang and Z. Zhang, *Linear Network Error Correction Coding*. New York, NY, USA: Springer, Feb. 2014.
[21] Z. Zhang, "Theory and applications of network error correction coding," *Proc. IEEE*, vol. 99, no. 3, pp. 406–420, Mar. 2011.

MINGJUN DAI received the Ph.D. degree in electronic engineering from the City University of Hong Kong in 2012. He joined the Faculty of the College of Information Engineering, Shenzhen University, Shenzhen, China, where he is currently an Associate Professor. His research interests include block chain, network coding design, distributed storage, and visible-light communication.

SHENGLI ZHANG received the B.Eng. degree in electronic engineering and the M.Eng. degree in communication and information engineering from the University of Science and Technology of China, Hefei, China, in 2002 and 2005, respectively, and the Ph.D. degree from the Department of Information Engineering, The Chinese University University of Hong Kong, in 2008. After that, he joined the Communication Engineering Department, Shenzhen University, where he is currently a Full Professor. From 2014 to 2015, he was a Visiting Associate Professor with Stanford University.

HUI WANG is currently a Full Professor and a Vice President with Shenzhen University, Shenzhen, China. His research interests include block chain and distributed storage.

SHI JIN is currently with the National Mobile Communications Research Laboratory, Southeast University, Nanjing, China, where he is also a Full Professor.