Hindawi
Security and Communication Networks
Volume 2021, Article ID 6694281, 7 pages
https://doi.org/10.1155/2021/6694281

Research Article

Blockchain-Based Automated System for Identification and Storage of Networks

Deepak Prashar,[1] Nishant Jha,[1] Muhammad Shafiq,[2] Nazir Ahmad,[3] Mamoon Rashid,[1] Shoeib Amin Banday,[4] and Habib Ullah Khan[5]

[1]*School of Computer Science and Engineering, Lovely Professional University, Phagwara, India*
[2]*Cyberspace Institute of Advanced Technology, GuangZhou University, Guangzhou, China*
[3]*Department of Information System Community College, King Khalid University, Muhayel, Saudi Arabia*
[4]*Department of Electronics & Communication Engineering, Islamic University of Science & Technology, Awantipora, India*
[5]*Department of Accounting and Information Systems, College of Business and Economics, Doha, Qatar University, Qatar*

Correspondence should be addressed to Muhammad Shafiq; srsshafiq@gmail.com

Received 19 December 2020; Revised 22 January 2021; Accepted 3 February 2021; Published 20 February 2021

Academic Editor: Omar Cheikhrouhou

Network topology is one of the major factors in defining the behavior of a network. In the present scenario, the demand for network security has increased due to an increase in the possibility of attacks by malicious users. In this paper, a blockchain-based system is suggested for securely discovering and storing networks. Techniques such as cloud-based storage systems are not efficient and are lacking in trust, privacy, security, and data control. The blockchain-based technique suggested in this paper is capable of resolving these challenges. Experiments were performed using Mininet, Cisco Packet Tracer, and Ethereum blockchain with the network inference algorithm. This algorithm is capable of inferring the network topology even when only partial information regarding the network is available. The results obtained clearly show that the network is resistant to malicious users and various external attacks, making the network robust.

1. Introduction

Managing networks has become a challenging task due to the complex nature and different structures of networks in different systems. Managing networks manually is not feasible due to factors such as a limited amount of time, challenges in tracing the configuration states of a large number of devices, need for specialists from various backgrounds, and defining an efficient strategy for network configuration management [1]. These factors are responsible for increasing the costs and efforts required for managing networks. Moreover, network topology is the extended version of the total resources in the network. Collection of information from the techniques such as software-defined networks (SDNs) has become a challenging task for improving QoS, network management, and routing [2]. The number of devices in connection with IP networks is forecasted to be increased upto 29 billion by the end of 2022. This growth is directly proportional to the growth in machine-to-machine communications [3]. These communications do not require any human intervention. By 2022, M2M connections are expected to be more than half of the global linked devices and connections [4]. IP networks are becoming much more common and dynamic as a result. This is affecting the internet causing the internet service providers to face challenges of increasing demands of bandwidth along with the LAN which also takes care of devices requiring machine-to-machine communications. To resolve the challenges faced in mapping the network, an automated system is present in this paper. This system is derived from the current techniques used for deducing the network information, which is capable of extracting information from the network when only partial information is available or if there are certain changes in the network.

A vast number of features are included in network control systems (NCSs) that allow automated and centralized device management. However, a suitable system discovery mechanism is needed in order to further expand the functionality of the NCS and to reduce network management effort. NCS currently supports connecting devices to the system manually, so the user has to provide the device's address and specify its form. The device type in the NCS defines the (internal) NCS interface should be used to communicate with the device (NETCONF, SNMP, Cisco Command Line Interface (CLI), etc., are used for this internal interface to configure the device). Various tools and techniques that are used on a large scale are based on the concept of traceroute at the IP level [5]. Systems such as skitter use twenty-four monitors which target the order of one million destinations. Other systems such as NCC TTM, RIPE, and NLANR AMP work on a mesh of traceroutes between a few hundred monitors [5]. There are challenges in scaling these techniques to higher levels. However, various large-scale techniques are being used in the present [5]. A number of critical network management tasks, such as network diagnosis and resource management, are mandatory for maintaining accurate knowledge of network topology. The system suggested in this paper will check the reachability of the nodes at each step and helps in efficiently finding the faulty nodes and helping in analyzing the traffic.

The major contributions of this paper include the following:

(1) A blockchain-based system is suggested for securely discovering and storing networks

(2) Experiments were performed using Mininet, Cisco Packet Tracer, and Ethereum blockchain with network inference algorithm

(3) Different tools and techniques that are used for network management and how to store a network safely and with security in a blockchain framework are analyzed and compared

The rest of the paper is structured as follows. Section 2 explains the literature reviewed, Section 3 explains the significance of work, Section 4 explains the methodology, Section 5 explains the experimental results, and Section 6 concludes the paper.

2. Literature Reviewed

As the internet is becoming more complex, there is a stronger need to study these complexities since manual identification of topology is not feasible for large and complex networks, and various studies have been done to suggest an automated system of inferring complex topologies [1]. In [6], SNMP algorithm for the discovery of network topology is analyzed, and its challenges are explained. In [7], the authors suggested topology control techniques for construction and management of IoT networks on large scale in smart cities. A novel approach is suggested for resolving the challenges and increasing the efficiency of the existing community detection algorithms by considering the network topology and other contents [8]. Analysis of combinatorial topology is done in an arbitrary structure of failure-free networks [9]. An analytical algorithm for finding the shortest paths having a common scheme of a family of networks on generating function was developed [10]. The authors in [11] suggested a SLDP protocol for efficient discovery and extraction of information about the topology of SDN. The authors in [12] presented a novel approach which helps in enabling a distributed discovery of topology in a 2-layer fashion without the knowledge of previous network configuration.

Khan et al. [13] presented a systematic survey of topology findings and related safety consequences for SDNs. Their survey highlighted the role of topology discovery in the conventional network and SDN, introduced a thematic taxonomy of topology discovery in the SDN, and offered insights into possible challenges to topology discovery. Azzouni et al. [14] implemented and described a new protocol called OpenFlow Discovery Protocol sOTDP, which is safe and efficient. sOFTDP needs to adjust the OpenFlow switch architecture minimally, eliminates major vulnerabilities, and enhances its performance during topology discovery. Deshpande et al. [15] developed an efficient BTCmap system for exploring and mapping the bitcoin topology network that is built and implemented. Delgado-Segura et al. [16] presented TxProbe, a modern bitcoin network topology reconstruction technique. They also performed bitcoin test network studies that show that their methodology correctly reconstructs topology and recalls more than 90%. Sharma et al. [17] suggested blockchain-based BIRD (Intercloud Resource Discovery) to resolve the limitations of current solutions for the nonfederated intercloud environment. This is an initial example that the blockchain concept is used to minimize the need for a trustworthy third partner or broker to be utilized by CSPs, thus ensuring the services are found and chosen in the best way possible.

Zheng et al. [18] primarily studied the automated node detection system based on the algorithm of Kademlia, including the protocol theory, the coordination handshake method, and the specific algorithm process. Finally, they observed the effects of automated discovery of nodes by the user in Ethereum and used Python for a quick experiment to locate Ethereum nodes automatically. Essaid et al. [19] suggested a topology discovery method that gathers and analyzes data for bitcoin P2P connections in real time utilizing a modified variant of the PageRank algorithm that assembles incoming graph research input nodes. In the same fashion, some other related works in the direction of security have been done by many researchers using the blockchain technology pertaining to the Internet of Things (IoT) and other latest applications [20–25]. In present works, we found that the network tomography is not a valid solution to traceroute-based techniques for inference of network topology. Thus, through this paper, we want to resolve the drawbacks of these studies and to suggest a better tool or technique with traceroute-based methods.

From the discussion of the existing literature mentioned above, the authors feel that there is a requirement of a

blockchain-based system for securely discovering and storing networks. The presence of this system will make network resistant to malicious users and various external attacks, making the network robust.

3. Significance of the Work

Current studies use the cloud for storage purposes. Major challenges of cloud-based storage systems are lack of transparency, trust, and data control. Since cloud-based storage is inherent, its challenges cannot be resolved fully [26]. However, blockchain-based storage systems, as suggested in this paper, can resolve these challenges along with providing a secure storage environment. A comparison between the cloud-based storage system and the blockchain-based storage system is shown in Table 1.

Blockchain-based storage systems, as suggested in the paper, have the following advantages:

Privacy control: in blockchain-based storage systems, every user can create their own identity in a decentralized fashion making the identity of the user anonymous from the real world. One of the major techniques used by cloud for resolving the privacy challenges is attribute-based encryption [26]. Blockchain-based storage resolves the same challenge by giving users the ability for generation and distribution of the secret keys, thus keeping the privacy in the network.

Security: the data must be encrypted before transmitting on the blockchain network. Although centralized storage network also provides encryption, the advantage of blockchain is that a single file is divided into fractions and distributed among various users called nodes in a network. This increases the security of the network as a malicious node is not able to affect the network. Centralized storage systems do not assure data integrity and processing techniques of the data [26], but a blockchain-based system does all these functions.

Bandwidth: in cloud-based storage systems, in case of downloading a file from the server, it gets downloaded fully from a single connection only, whereas in a blockchain-based storage system, each fraction of the file is downloaded from different storage providers, thereby making the downloads to run in parallel and increasing the bandwidth to the maximum and minimizing the download time [27].

Reputation mechanism: blockchain-based storage systems use the reputation mechanism [28]. This mechanism will allow the network to validate the space provider's sincerity automatically for ensuring that the hosts function upto their claims if they are not eliminated from the network. It makes the network and the storage providers trustworthy.

All these features make a blockchain-based storage system efficient and more secure than centralized storage systems.

4. Methodology

This section covers the algorithms used for the network topology and the associated blockchain concept that is associated with it so that the proposed system becomes more secure and robust. In a SDN, blockchain can be used for securing the application from various outside attacks [29]. The controller applications can be guarded from being tampered by the malicious users. This is summarized by the CPSA (Control Plane Security Algorithm) [29] in the form of Algorithm 1.

The experiments were carried out on a system installed with Windows 10, Home Single Language, Node.js used as a controller, and Ethereum as the blockchain. Cisco Packet Tracer and Mininet were also used. In studies done by authors in [1, 29] and other works as discussed in Section 2 above, the allocation of monitors was done at network edge with the goal of each monitor is a collection of traces of each monitor. After collection of these traces, these are sent to the network operating system for processing. The use of blockchain is that it merges various topologies into a single topology at the end of each round of consensus. Network topology acquisition process involves various steps that are discussed in this section in the form of algorithms. Let us consider a node running the inference algorithm for the topology. It can be from a monitor or a sensor. The algorithm [30] is given in the form of Algorithm 2.

Hop distance in the targeted node list between that node and other node is calculated by the function compute_distance(). If the nodes are running the iTop server, then the target nodes are defined as the monitors. If the list of the goal nodes is made up of network nodes, then these are classified as sensors. In this paper, we have calculated the distances by pinging the target host and exploiting a network not containing firewalls. This was the similar consideration taken in [29] and is possible always to extract the gaps between any nodes. The information regarding the trace between the nodes is collected and stored by using the store_trace() function. No assumptions were made on the network at this point. The resultant traces will be full of asterisks if any blocking routers are present [1]. The construction of the virtual topology [1] takes place using the create_virtual_topology() function. This is shown by Algorithm 3.

The collection of all the traces is checked by Algorithm 3. The identity of the node is used for the collection of traces from the source to the destination (Steps 3 and 4). After identification, it checks whether the query can be processed by the destination or not (Step 5). In this case, the anonymous routers can be present along the path [1]. After construction of the virtual topology, merging of topology [1] is done as described in Algorithm 4.

We have to first calculate the merge topology from the virtual topology. We have to select a nonempty set of merger options beginning with the lower merging choices. Each of the chosen merging options is then sorted in order and is compared with other edges belonging to the unsorted group. Our goal is to find the edge that is in consistency with the chosen edge to perform merging. Now, we will change the

Table 1: Comparison between the cloud-based storage system and the blockchain-based storage system [26].

Type of storage	Is it open source?	Scalable	Privacy	Method of payment	Processing of data	Cost of implementation	Facility for choosing the type of hardware
Cloud-based	No	Highly scalable	Lower	Fiat money	Yes	Costly	No
Blockchain-based	Yes	Complex	Higher	Cryptocurrency	Yes	Low	Yes

```
Read (ID, Y), Y ⟶ Config. Data store
  If ID ⟶ read, then
    Allow ⟶ Read
  End
End if
    If write (ID, Y) ⟶ then
  (Hash value, Y) ⟶ Alert
  Validate using X, X ⟶ App. Blockchain
  End if
    If write (ID, Z), Z ⟶ Controller
    Alert (ID, Hash value (Z)
    Validate using X
End
```

Algorithm 1: CPSA.

```
Input: collection of monitors and sensors in a network
Output: topology inferred
Step 1: call function compute_distance()
Step 2: call function store_trace()
Step 3: call function create_virtual_topology()
Step 4: call function compute_merge_option()
Step 5: call function create_merge_topology()
Step 6: call function return save_topology()
```

Algorithm 2: iTop algorithm.

```
    Input: T ⟶ collection of traces
      D ⟶ distance matrix between the nodes
    Output: TO ⟶ virtual topology
(1)   function ⟶ create TO
(2)   for each trace ∈ ∈∈traces do
(3)   source ⟶ get_source (trace)
(4)   D ⟶ get_D (trace)
(5)   if get_answer ⟶ D (trace) then
(6)   R ⟶ get_router (trace)
(7)   (TO, R) ⟶ add_router
(8)   (Path, R) ⟶ add_path
(9)   else
(10)  D ⟶ distance{source}{D}
(11)  R ⟶ get_router (D, source, traces)
(12)  RD ⟶ get_router (D, source, traces)
(13)  N ⟶ nonresponding router (traces, source, D)
(14)  add router (TO , R U N U )
(15)  add path (path, R U N U reversed (N))
(16)  return (T_O, Paths)
```

Algorithm 3: Virtual topology construction.

Input: T ⟶ collection of traces
P ⟶ paths of virtual topology
TO ⟶ virtual topology
Output: X ⟶ merged table options
Y ⟶ endpoint compatibility table
(1) Function ⟶ compute_merge():
(2) E ⟶ edges in path
(3) Z ⟶ merge_table (TO)
(4) Preserve_trace (E, Z)
(5) D_Preservation (TO, paths, E)
(6) Y ⟶ compatible table()
(7) Endpoint compatible ⟶ (Y, X, TO)
(8) Return (X, Y)

Algorithm 4: Merging option calculation.

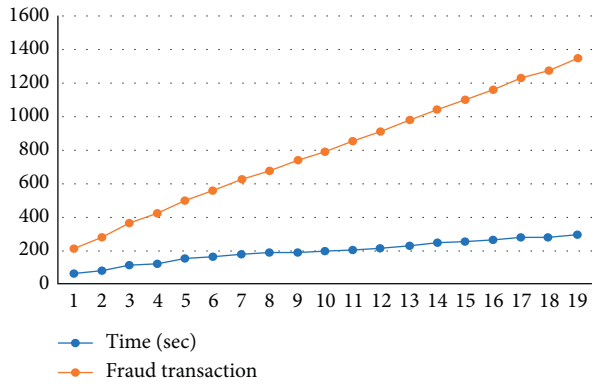

Figure 1: Time of execution of a consensus round of fraudulent transaction for simulation 1.

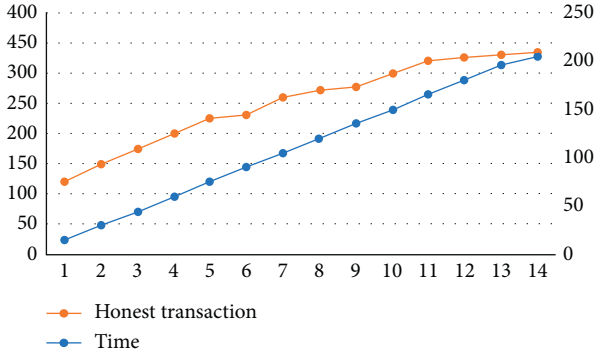

Figure 2: Honest transactions.

topology table and choices of fusion accordingly. We will continue this operation until all the edges of the merge options become empty.

5. Experiments and Results

Now, we have configured the virtual host and the router by using Mininet and Cisco Packet Tracer. A network is created similar to the real LAN. Since they are not very large, we will be able to differentiate between the original and inferred topology just by seeing them. The criteria used for evaluation [1] are shown in Table 2.

Simulation 1: in the first simulation, we have set up a test network with 1 router and 2 subnets. Two sensors are used in running the inference algorithm. Each sensor is placed in different subnets, and each sensor contains 4 hosts. In the second, each includes 12 hosts. In the third subnet, there are 40 hosts. We found that both precision and recall are greater than 85%. The values inferred from this simulation are given in Tables 3 and 4.

Modification of the values of Ethereum with the number of malicious nodes present in the network is done for calculating the time required by the nodes for achieving consensus [1]. A set of 25 experiments were done, keeping the parameters the same for each simulation, and then the mean time for execution of the experiment was evaluated. The time required for reaching consensus is also dependent on the constants present in the configuration file of each node. These constants are responsible for influencing the time required (minimum or maximum) for achieving consensus [1]. The constants are as follows:

LEDGER_MINIMUM_CLOSE = 40 sec

LEDGER_MAXIMUM_CLOSE = 60 sec

LEDGER_MINIMUM_CONSENSUS = 20 sec

LEDGER_MAXIMUM_CONSENSUS = 60 sec

600 transactions are sent to the nodes present in the blockchain network. We have considered Ethereum of 70% of the UNL nodes as each UNL is having 5 nodes. A node will notify about the consensus if and only if at least 3 nodes are contained in the UNL. Figure 1 shows that the time of execution is maximum when fraudulent transactions are inserted into the nodes.

Simulation 2: we have considered a firewall with 3 subnets and 3 sensors with precision values between 0.80 and 1.00. A large number of sensors are used for reducing the number of false negatives with an increase in the number of false positives for the same purpose by

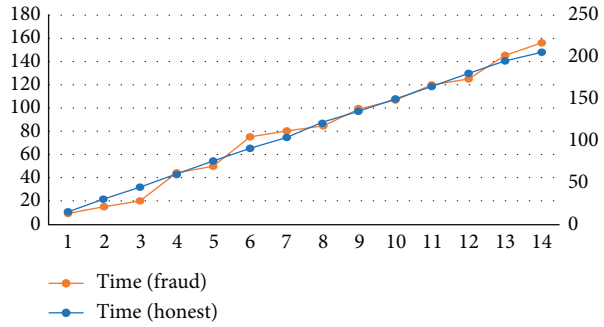

Figure 3: Time variation between fraudulent transactions (malicious nodes) and honest transactions.

Table 2: Criteria for evaluation.

T^+	Nodes belonging to both real and inferred topology
T^-	Nodes not belonging to either of the real or inferred topology
F^+	Nodes belonging to the inferred topology but not the real topology
F^-	Nodes belonging to the real topology but not the inferred topology
Precision value	Helps in measuring the precision of the topology inferred
X (recall)	Measure of completeness
F-1 measure	Harmonic average of the precision and recall value

Table 3: Values inferred from the topology for simulation 1.

No. of sensors	F^+	F^-
1	1	1
1	2	2
2	2	4
2	2	4
3	1	4
3	1	5

Table 4: Values inferred from the topology for simulation 2.

No. of sensors	F^+	F^-
1	2	2
1	2	2
2	4	4
2	5	4
3	5	5
3	6	5

keeping the precision values constant. Another fact we found is the recall value reached its maximum, i.e., equal to 1, while using 3 sensors. The high number of false negatives is due to the fact that it is used by one random sensor only and the networks being isolated making the sensor in the LAN subnet unable to capture.

The effect of various Ethereum values on time required for achieving consensus is tested in this simulation. Different types of honest transactions are sent to the nodes of the blockchain considering the values such as 70%, 80%, and 100%, respectively, and are plotted in Figure 2.

Simulation 3: the last simulation is set up using 3 routers and 4 subnets with root router blocking. The sensors are placed in between the hosts and the router within every subnet. All of the incoming packets are dropped by the blocking router, preventing the contact between the hosts in the distinct subnets only. A centralized NOC used for collecting the traces from the sensors and running the inference algorithm for the topology can reconstruct the nodes, but this can remove all the benefits that we got using decentralization. Now, we can compare the times of execution when the Ethereum value is 85% with one malicious node. The execution time is increasing with the malicious transactions indicating that the Ethereum values do not affect the time taken in achieving consensus, but it slows the consensus mechanism. This is shown in Figure 3.

6. Conclusion

In this paper, an autonomous system for detecting and storing a network is suggested. For evaluating our system, we have performed various experiments in a LAN scenario with inference algorithms. The algorithms help in analyzing the traffic for discovering newer nodes and testing the reachability of the nodes periodically. This is the way through which we can track the network topology. We have chosen the fraudulent nodes with Ethereum values to achieve consensus in order to make an honest transaction. Upon analysis, we have found that the Ethereum values do not influence the time required in achieving consensus. In fact, the processing time of fraudulent transactions slows down the consensus mechanism as these transactions must be verified in order to get discarded. Also, in case of any exterior attacks by the malicious nodes, these nodes do not get included in the ledger or transaction process to affect the other nodes. This makes the network topology more secure and robust.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] D. Knertser and V. Tsarinenko, "Network device discovery," 2013.

[2] L. Ochoa-Aday, C. Cervelló-Pastor, and A. Fernández-Fernández, "Discovering the network topology: an efficient approach for SDN," 2016.

[3] M. Shafiq, X. Yu, A. K. Bashir, H. N. Chaudhry, and D. Wang, "A machine learning approach for feature selection traffic classification using security analysis," *The Journal of Supercomputing*, vol. 74, no. 10, pp. 4867–4892, 2018.

[4] Cisco, *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2017–2022*, Cisco, San Francisco, CA, USA, 2019.

[5] B. Donnet, T. Friedman, and M. Crovella, "Improved algorithms for network topology discovery," in *Proceedings of the International Workshop on Passive and Active Network Measurement*, pp. 149–162, Springer, Berlin, Germany, 2005.

[6] H. Wang, "Improvement and implementation of wireless network topology system based on SNMP protocol for router equipment," *Computer Communications*, vol. 151, pp. 10–18, 2020.

[7] K. S. Desikan, V. J. Kotagi, and C. S. R. Murthy, "Topology control in fog computing enabled IoT networks for smart cities," *Computer Networks*, vol. 176, Article ID 107270, 2020.

[8] A. Bhih, P. Johnson, and M. Randles, "An optimisation tool for robust community detection algorithms using content and topology information," *The Journal of Supercomputing*, vol. 76, no. 1, pp. 226–254, 2020.

[9] M. Shafiq, Z. Tian, A. K. Bashir, X. Du, and M. Guizani, "Corrauc: a malicious bot-iot traffic detection method in iot network using machine learning techniques," *IEEE Internet of Things Journal*, 2020.

[10] E. A. Monakhova, O. G. Monakhov, A. Y. Romanov, and E. V. Lezhnev, "Analytical routing algorithm for networks-on-chip with the three-dimensional circulant topology," in *Proceedings of the Moscow Workshop on Electronic and Networking Technologies (MWENT)*, pp. 1–6, Moscow, Russia, 2020.

[11] A. Nehra, M. Tripathi, M. S. Gaur, R. B. Battula, and C. Lal, "SLDP: a secure and lightweight link discovery protocol for software defined networking," *Computer Networks*, vol. 150, pp. 102–116, 2019.

[12] L. Ochoa-Aday, C. Cervelló-Pastor, and A. Fernández-Fernández, "eTDP: enhanced topology discovery protocol for software-defined networks," *IEEE Access*, vol. 7, pp. 23471–23487, 2019.

[13] S. Khan, A. Gani, A. W. A. Wahab, M. Guizani, and M. K. Khan, "Topology discovery in software-defined networks: threats, taxonomy, and state-of-the-art," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 303–324, 2016.

[14] A. Azzouni, R. Boutaba, N. T. M. Trang, and G. Pujolle, "sOFTDP: secure and efficient topology discovery protocol for SDN," 2017, https://arxiv.org/abs/1705.04527.

[15] V. Deshpande, H. Badis, and L. George, "BTCmap: mapping bitcoin peer-to-peer network topology," in *Proceedings of the 2018 IFIP/IEEE International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks (PEMWN)*, pp. 1–6, IEEE, Toulouse, France, 2018.

[16] S. Delgado-Segura, S. Bakshi, C. Pérez-Solà et al., "TxProbe: discovering bitcoin's network topology using orphan transactions," in *Proceedings of the International Conference on Financial Cryptography and Data Security*, pp. 550–566, Cham, Switzerland, 2019.

[17] M. Sharma, J. Singh, and A. Gupta, "Intelligent resource discovery in inter-cloud using blockchain," in *Proceedings of the 2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing*, pp. 1333–1338, IEEE, Piscataway, NJ, USA, 2019.

[18] L. Zheng, X. Helu, M. Li, and H. Lu, "Automatic discovery mechanism of blockchain nodes based on the kademlia algorithm," in *Proceedings of the International Conference on Artificial Intelligence and Security*, pp. 605–616, Springer, Cham, Switzerland, 2019.

[19] M. Essaid, S. Park, and H. Ju, "Visualising bitcoin's dynamic P2P network topology and performance," in *Proceedings of the 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pp. 141–145, IEEE, Seoul, South Korea, 2019.

[20] M. A. Khan and K. Salah, "IoT security: review, blockchain solutions, and open challenges," *Future Generation Computer Systems*, vol. 82, pp. 395–411, 2018.

[21] P. Mehta, R. Gupta, and S. Tanwar, "Blockchain envisioned UAV networks: challenges, solutions, and comparisons," *Computer Communications*, vol. 151, 2020.

[22] S. Pu, "Industrial applications of blockchain to IoT data," in *Blockchain and Crypt Currency*, pp. 41–58, Springer, Singapore, 2020.

[23] M. Shafiq, Z. Tian, A. K. Bashir, K. Cengiz, and A. Tahir, "Softsystem: smart edge computing device selection method for IoT based on soft set technique," *Wireless Communications and Mobile Computing*, vol. 202010 pages, 2020.

[24] A. Banotra, J. S. Sharma, S. Gupta, S. K. Gupta, and M. Rashid, "Use of blockchain and internet of things for securing data in healthcare systems," in *Multimedia Security*, pp. 255–267, Springer, Singapore, 2021.

[25] F. Syed, S. K. Gupta, S. Hamood Alsamhi, M. Rashid, and X. Liu, "A survey on recent optimal techniques for securing unmanned aerial vehicles applications," *Transactions on Emerging Telecommunications Technologies*, Article ID e4133, 2020.

[26] N. Z. Benisi, M. Aminian, and B. Javadi, "Blockchain-based decentralized storage networks: a survey," *Journal of Network and Computer Applications*, vol. 162, 2020.

[27] D. Vorick and L. Champine, "Sia: simple decentralized storage," 2014, https://sia.tech/sia.pdf.

[28] R. Dennis and G. Owen, "Rep on the block: a next generation reputation system based on the blockchain," in *Proceedings of the 10th International Conference for Internet Technology and Secured Transactions (ICITST)*, pp. 131–138, London, UK, 2015.

[29] B. Lokesh and N. Rajagopalan, "A Blockchain-based security model for SDNs," in *Proceedings of the IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, pp. 1–6, Bangalore, India, 2020.

[30] B. Holbert, S. Tati, S. Silvestri, T. F. La Porta, and A. Swami, "Network topology inference with partial information," *IEEE Transactions on Network and Service Management*, vol. 12, no. 3, pp. 406–419, 2015.