Available online at www.ijpe-online.com

International Journal of Performability Engineering

vol. 15, no. 12, December 2019, pp. 3187-3195
DOI: 10.23940/ijpe.19.12.p10.31873195

A New Network Intrusion Detection System based on Blockchain

Jinhua Fu[a,b], Mixue Xu[a], Yongzhong Huang[a], and Hongwei Tao[b,*]

[a]*State Key Laboratory of Mathematical Engineering and Advanced Computing, Information Engineering University, Zhengzhou, 450001, China*
[b]*School of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou, 450002, China*

Abstract

With the increasing application of computers and networks, several network security problems have emerged, and thus network intrusion detection systems have become the focus of network security research. In order to achieve the purpose of intrusion detection and protection, the traditional network intrusion detection system extracts features from the data of the network data stream according to the feature recognition algorithm and compares the extracted features with those in the training set to recognize the behavior. However, if a user wants to effectively detect malicious behaviors in the network, a large feature library is needed, and it cannot be shared with other users, which makes the quality of single user detection lower than the highest detection quality of the whole network. Blockchain, which is a new network system of decentralization, de-trust, tamper-proof, anti-counterfeiting, and traceability, plays an important role in the transmission and sharing of high-value data. In this paper, a new network intrusion detection system is designed based on blockchain, which can enable users to share feature libraries over the whole network by means of P2P network transmission. Meanwhile, its network structure and consensus algorithm are presented, and its security and performance are analyzed. Analysis results show that this system has lower false negative rates.

Keywords: pattern recognition; intrusion detection; blockchain; P2P network

(Submitted on August 7, 2019; Revised on November 3, 2019; Accepted on November 14, 2019)

1. Introduction

In 1980, P. A. James proposed the concept of intrusion detection and the classification of system threats [1]. In 1990, NSM (network and security manager), which was developed by Heberlein et al., made NIDS (network intrusion detection) an important part of intrusion detection. Network intrusion detection refers to the detection of various network behaviors that endanger the security of computer systems. At present, it is mainly divided into anomaly-based network intrusion detection and feature-based network intrusion detection. Anomaly-based network intrusion detection checks intrusion behaviors by defining normal user behaviors and identifying the operations that are inconsistent with normal behaviors. Feature-based network intrusion detection first extracts the characteristics of the behaviors of received packets and then compares them with the features in the existing feature library. If the characteristics extracted are in the library, then the behavior will be considered an intrusion behavior. Since we cannot define normal user behaviors or malicious behavior characteristics, the existing network intrusion detection systems usually combine the two.

In this paper, feature-based network intrusion detection based on blockchain technology is expanded, which can achieve the goal of sharing whole network feature libraries and improve the detection security. Section 2 introduces the intrusion detection architecture on the basis of the feature network. Section 3 describes blockchain infrastructure, including P2P networks, smart contracts, consensus algorithms, and other parts related to the network intrusion detection system. Section 4 presents our new network intrusion detection system. In Section 5, the security analysis of our new system is proposed, and its performance is compared with other network intrusion detection systems. Comparison results show that our system has lower false negative rates. Section 6 provides the summary and conclusion.

* Corresponding author.
E-mail address: tthhww_811@163.com

2. Feature-based Network Intrusion Detection System

A feature-based network intrusion detection system (also known as misuse detection system) usually consists of three components, information capture, feature analysis, and result response [2], as shown in Figure 1. Information capture collects various information on the network and passes the collected information as parameters to the feature analysis engine. Feature analysis includes feature extraction and feature judgment. Feature extraction is a pretreatment process that transforms the existing intrusion behaviors (training set) to obtain the feature library and detection model. Feature judgment assesses whether the existing network data information conforms to the features of the feature library. The result is a response to the output of the feature judgment, such as notifying the system administrator, breaking the network intruder's link, and collecting intrusion information.

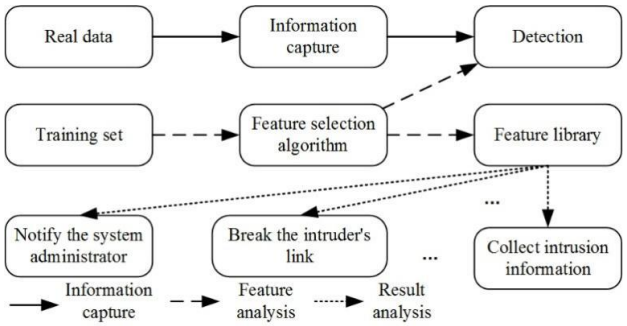

Figure 1. Feature-based network intrusion detection system

With the development of data mining technology, feature libraries can be updated through neural network training, so that the accuracy will increase and the false negative rates will decrease. However, due to the different training sets and intrusion detection methods of different terminals, the value of accuracy and false negative rates will float up and down. Therefore, a network intrusion detection system under single node conditions may not fully utilize resources.

3. Blockchain Infrastructure

Blockchain originates from the concept of "block of chain" in Nakamoto's Bitcoin White Paper [3]. With the rapid development of block chain technology, it can effectively guarantee the authenticity, security, and reliability of data. It has been widely used in Bitcoin Litecoin [4], Monroe [5], Zcash [6], medical data [7], personal data protection [8], and data allocation schemes [9]. The blockchain's architecture of Bitcoin is comprised of six parts, which include a data layer, network layer, consensus layer, incentive layer, contract layer, and application layer, as shown in Figure 2.

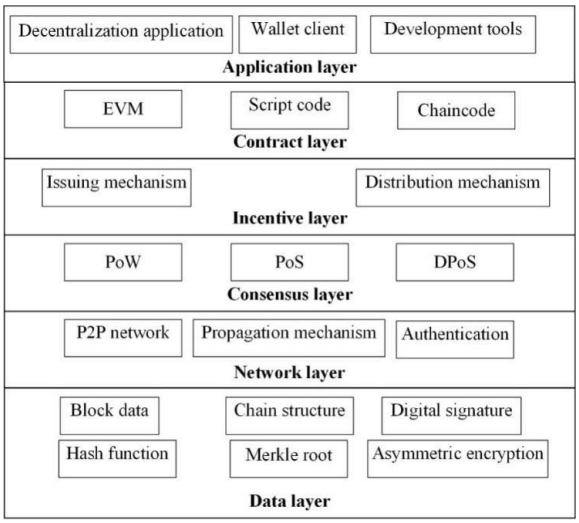

Figure 2. Blockchain system infrastructure

This section mainly introduces the P2P network of the network layer in the blockchain infrastructure, the consensus algorithm of the consensus layer, and the EVM smart contract of the contract layer.

3.1. P2P Network

Centralized network technology is usually used to undertake data dissemination, including a centralizing node and other nodes. Each node should be connected with the centralized node. However, it is necessary to undertake message transmission by sending a packet to the centralized node and then re-transmitting to each node through the centralized node. The centralized network is shown in Figure 3.

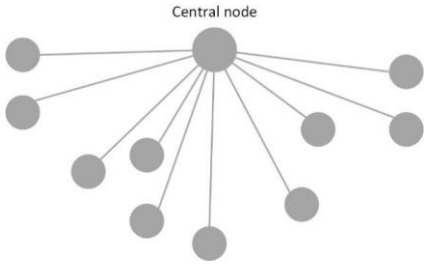

Figure 3. Centralized network

From the above picture, we can see that the centralized node is capable of mastering all the information of nodes on the network, including the IP address, the information of re-transmission between two mods, and so on. The security and privacy of the network are based on the security of the centralized node and network morality.

However, the network technology of P2P reforms the traditional centralized communication, which not only effectively solves the security problem of centralized nodes, but also reduces network delays when the server experiences a performance bottleneck. It also prevents the destructive effect of the whole network caused by the downtime of a few nodes so as to increase the robustness of the system [10]. In addition, it is capable of improving the transmission speed of the whole network as the numbers of users and shared resources increase. In accordance with the structure relation, P2P can be divided into three kinds of topological form: a fully distributed unstructured network, fully distributed structure topology, and semi-distributed topology.

Bitcoin makes use of the fully distributed unstructured network. Each node randomly selects an existing node link when it joins. The form of the network is shown in Figure 4.

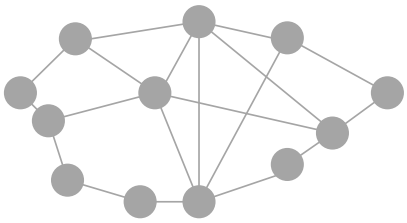

Figure 4. Fully distributed unstructured network

This way may cause the problems of the so-called "the circulation of the universal flood", which means that the messages given out from node A will go through nodes B and C and then back to node A, and "the storm of responses", where the messages required by node A are cropped in many nodes and sent back to node A, resulting in the breakdown of the node. For these reasons, the decentralized structured topology is applied in Ethereum [11]. This is how new nodes can be linked with the existing nodes in a good order, as shown in Figure 5.

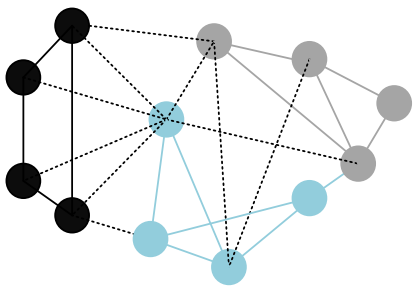

Figure 5. Decentralized structured network

The networking node in the decentralized structured topology is so obvious that it is vulnerable to network attackers. Therefore, EOS draws on the advantages of centralized structure and decentralized unstructured topology, improving the network efficiency and security with the random interconnected structure of super nodes and the structured link of general nodes [12]. The result is a new networking mode named semi-distributed topology, as shown in Figure 6.

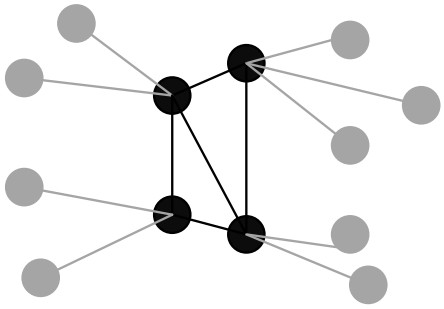

Figure 6. Semi-distributed network

On the basis of the new network intrusion of blockchain, the network layer is tested. As an improved semi-distributed topology, it can effectively strengthen the efficiency of nodes and the stability of systems.

3.2. Smart Contract

The smart contract is a commitment defined in the form of numbers; once the predetermined conditions are met, the subsequent preset steps are executed sequentially until the parties of the contract are complete. The smart contract was first put forward in the 1990s. It is an application hypothesis based on the problems of slow execution of real contracts, cheating in the middle link, and so on. However, it was impossible for the old style to assume a trusted third party when it encountered a trust problem at that time. It was not enough to solve the problem until the emergence of blockchain, which allowed the trust problems to be solved. The emergence of Ethereum further made smart contracts practical.

As a way of understanding smart contracts, a simple voting process of the smart contract emerged. The contract stipulates that authorized users may vote through a one-person-one-vote mechanism. Its workflow is shown in Figure 7.

Figure 7 is based on the running example of the Ethereum voting smart contract. The network intrusion detection contract described by us is similar to Ethereum, which is Turing-complete, so that it can support the deployment of various pattern detection algorithms.

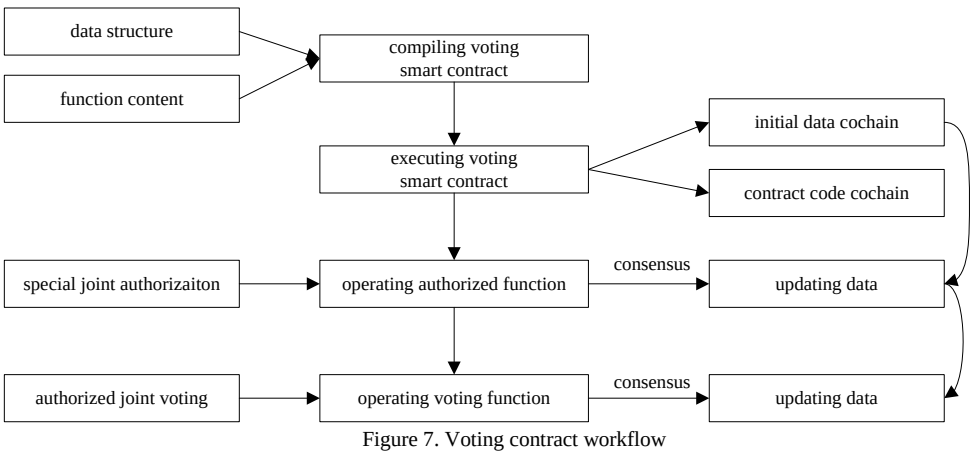

Figure 7. Voting contract workflow

3.3. Consensus Algorithm

The consensus problem is the most important problem in distributed computing, because the consensus algorithm that can be completed under synchronous conditions cannot be completed under asynchronous conditions. This is called the Byzantine generals problem [13]. Later, with the development of the consensus algorithm, paxos appeared. Consensus

algorithms such as paxos [14] and pbft [15] can effectively coordinate the problem of coordinated calculation of a small number of asynchronous nodes. However, the traditional consensus algorithm cannot satisfy the requirements of large-scale nodes requiring equal participation in consensus. Therefore, the PoW consensus algorithm proposed by Bitcoin provides a solid foundation for the application of large-scale nodes to reach a consensus. PoW determines the operation result of each node through independent resolution of each node and can be used as the final result. Other nodes are guaranteed through the incentive mechanism. However, because PoW has the problems of wasting resources and re-centering, there are limitations for some practical application scenarios. Therefore, the PoS mechanism was proposed. PoS was originally proposed by PPCoin [16]. According to the number of CoinDays (time of possession of coins) owned by the node, the difficulty of mining is determined. Later solutions include PoS algorithms such as Casper, which is agreed through multiple rounds of bill mortgages. However, due to its slower speed and the existence of attacks, Casper is still being tested. DPoS mainly serves the alliance chain and requires all the consensus nodes to have a strong driving force to maintain the block. The idea is to make the nodes reach a consensus quickly by narrowing the consensus group, which is more in line with the actual demand of high throughput applications.

Our new network intrusion detection consensus algorithm is an improved DPoS that can determine whether a system is threatened by means of proxy node voting. It can remove bad proxy nodes by 1/2 full-node rebellion out of the system of blockchain, which effectively prevents malicious proxy nodes from attacking the entire system.

4. New Network Intrusion Detection System

The application of network intrusion detection systems can be simplified into five layers by the blockchain system infrastructure, including a data layer, network layer, common understanding layer, contract layer, and application layer, as shown in Figure 8.

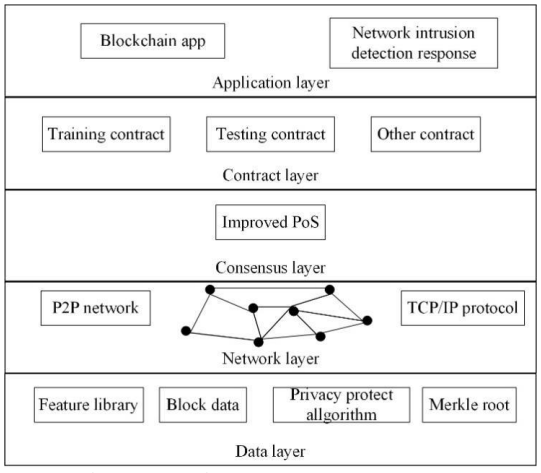

Figure 8. Infrastructure of new network intrusion detection system

The data layer includes feature library data for network intrusion detection, contract data recorded in blocks, algorithms related to privacy protection, and Merkle roots for simplifying data information. The network layer contains a P2P network in blockchain structure and a TCP/IP protocol used in network intrusion detection information capture. Considering the problem of single point attack, the design of the consensus layer uses the improved PoS for input information. The application layer contains the blockchain interface and the network intrusion detection response. Through the interface of the blockchain part, the performance of the real data stream in the intrusion algorithm verification of different nodes can be uniformly summarized, in order to determine whether to send a warning message to the user.

This section introduces a new network intrusion detection system mainly from the aspects of the network layer and consensus layer. The network structure of the blockchain part, the proxy node processing algorithm, and the common node processing algorithm in the consensus are given, and the operation principle of the system is explained.

4.1. Network Structure of New Network Intrusion Detection System

The new network intrusion detection system has a two-layer network structure, and each ordinary node establishes a two-way link with the proxy node and a one-way link with the adjacent and spaced ordinary nodes. The specific link mode is shown in Figure 9.

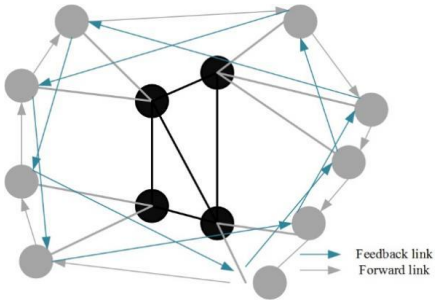

Figure 9. Network structure of new network intrusion detection system

The network structure of the new network intrusion detection system is divided into two layers. The first layer is the same semi-distributed link as DPoS, and the other part is a circular feedback link. To judge whether the agent node is evil or not, all the information in the network is collected by communication with the surrounding nodes. Once more than $1 < 2$ nodes propose that the proxy node commit evil, the proxy node will be excluded from the detection system so that the proxy node can no longer achieve the best detection effect in the network. Because of the downtime and failure of the distributed node, the feedback link is set. Once the node does not receive it within a certain time or receives an error message, the node will automatically exclude the previous one and move to the next one.

Due to the difference in information transmission of the two networks, proxy nodes, with their lower quantities, can achieve consensus agreement and carry out operations. Common nodes can achieve consensus agreement only after one of them collects more than half of the signatures. If the number of common nodes N is M times that of the proxy node whose number is n, the time transference period is $M \times$ epoch, that is, the proxy node can change once after M times of consensus proxy.

4.2. Consensus Algorithm of New Network Intrusion Detection System

The consensus algorithm of the new network intrusion detection system is the improved PoS consensus algorithm based on the above network layer. The consensus algorithm is composed of the proxy node processing algorithm and common node processing algorithm. The proxy node processing algorithm is given in Algorithm 1. In Algorithm 1, the data of the previous block is denoted as $Block_{i-1}$, the block header is represented by *header*, $H(.)$ is the hash algorithm, $sig(.)$ is the signature algorithm, $msgs = \{msg_1, msg_2, \cdots, msg_{nr}\}$ is the data stream collected by the proxy node, *test* is the proxy node detection algorithm, *feat* is the feature processing algorithm, and *f_data* is the feature library. The final detection result 0 represents no threat, and 1 represents a threat. Note that $H(msgs)$ represents many hash values of the messages. Upon termination of Algorithm 1, the block will add some new messages or a signature of the proxy node.

Algorithm 1. Proxy node processing algorithm

Proxy node processing algorithm

Input: $Block_{i-1}$, raw_Block_i, $msgs$, f_data
Output: $Block_i$
Algorithm:
if $raw\_Block_i \neq \emptyset$
  $num = 0$;
  for $j = 1$ to n_r:
    if $H(msg_j)$ in raw_Block_i && $test(msg_j, f\_data) = 1$
      num++;
      $f\_data = f\_data \| \text{feat}(msg_j)$
    endif;
  endfor;
  if $num > 1/3\ n_r$
    $Block_i = raw\_Block_i \| sig(header)$
  elif
    $Block_i = H(msgs)$
    $Header = H(Block_{i-1}) \| Merkle(Block_i)$
    $Block_i = header \| Block_i \| sig(header)$
  endif
  return $Block_i$
else
  $Block_i = \{\}$;
  for $j = 1$ to n_r:
    if $test(msg_j, f\_data) = 1$:// the results are threatening
      $Block_i = Block_i \| H(msg_j)$;

```
            f_data = f_data||feat(msg_j)
          endif;
        endfor;
      header = H(Block_{i-1})||Merkle(Block_i);
      Block_i = header||Block_i||sig(header)
    endif
    return Block_i
```

For a data *msg* that needs to be checked by common nodes, in order to prevent the delay from causing the data to be unprocessed, the block height is *i* when this data is sent, and the block is judged when the block height is increased to *i* + *s*. That is, when *msg* exists in *s* blocks received by normal nodes and the block is signed by more than half of the proxy nodes, normal nodes will identify it as threatening data and give an alarm. When there is no *msg* in the received block, the normal node considers the data to be safe. Assuming the number of proxy node is *m*, the processing algorithm of normal nodes is given in Algorithm 2.

Algorithm 2. Normal node processing algorithm

```
Common node processing algorithm
Input:Block_i, Block_{i+1},…, Block_{i+s}, msg
Output:0, 1
Algorithm:
    for t = i to t = i + s
      if msg in Block_t
        num = #sig(header);
        if num > 1/2 m
          return 1;
        endif
      endif;
    endfor;
    return 0;
```

5. Analysis of New Network Intrusion Detection System

5.1. The Security Analysis of the New Network Intrusion Detection System

The security of the new system includes the attack success rate and the attack cost. It is generally believed that the attack method of a distribution system is the damage of the proxy mechanism by node union. On this basis, the attack success rate and success cost of the new network intrusion detection system will be analyzed in this section. In a distribution consensus system, the amount of nodes that are shut down does not exceed one-third.

Suppose event *E* indicates that the bad proxy node is successful and cannot be detected by the common node, event E_1 expresses that the amount of bad proxy nodes is more than half, and event E_2 represents that common nodes detect the proxy nodes with consensus on certain information.

Due to the large number of common nodes, the upper limit distribution of bad proxy nodes can be considered to follow a binomial distribution $B(n, 2/3)$, and then we have

$$\Pr(E_1) = \sum_{i=0}^{n/2} \left(\frac{2}{3}\right)^i \left(\frac{1}{3}\right)^{n-i}$$

Where *n* is the number of proxy nodes. The upper limit distribution of bad proxy nodes is shown in Figure 10.

From Figure 10, we can see that when the number of proxy nodes reaches 120, the success rate of bad proxy nodes is only 8.65×10^{-5}, while the probability of bad proxy nodes avoiding detection by common nodes is

$$\Pr(E) = \Pr(E_1 \cap \overline{E_2}) = \Pr(E_1) - \Pr(E_1 \cap E_2)$$

Proxy nodes are operated with the PoS algorithm. In the case of the same number of participating nodes, the PoS algorithm runs faster than the PoW and PBFT algorithms. Therefore, in order to improve safety, the quantity of proxy nodes should be increased.

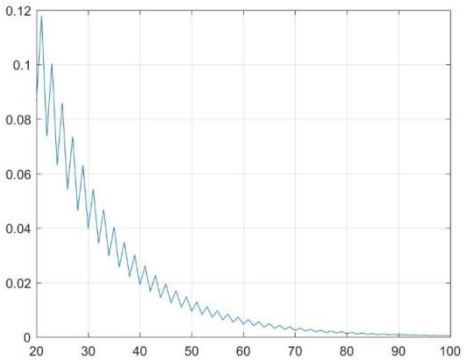

Figure 10. Upper limit probability of bad proxy nodes

5.2. The Performance Analysis of the New Network Intrusion Detection System

The new network intrusion detection system is different from the traditional network intrusion detection system; it can fully utilize the advantages of distributed networks and synthesize the detection effect of each node through blockchain technology to make the accuracy and false negative rates reach the network limit. Furthermore, with the expansion of the network scale, the performance of the system will be improved. In the following, an experiment is conducted to simulate the delay of network transmission and random shutdown of nodes and set a small amount of repeated template sets. The comparison results of the detection accuracy rate between the single node intrusion detection system and that based on blockchain are shown in Figure 11.

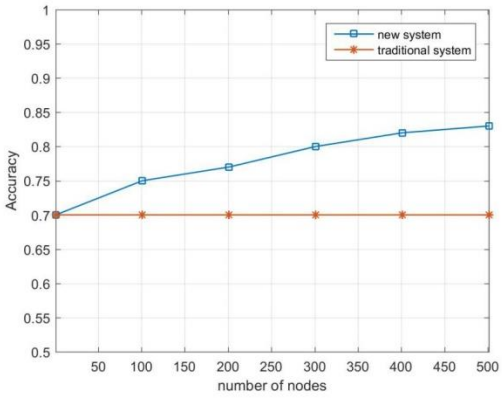

Figure 11. Contrast diagram of accuracy

From Figure 11, we can deserve that the new network intrusion detection system not only has a higher accuracy rate but also has scalability when the number of nodes is not 1, that is, when the number of nodes increases, the indicators in each part will still be improved.

6. Conclusions

Security is of great importance to network development. To a great extent, a network intrusion detection system can prevent malicious code, virus software, and DDoS attacks. Improving the accuracy and false negative rates of network detection systems plays an important role in personal computers and center servers. In this paper, the detection algorithms of each node can be integrated together so that they can exceed the original maximum detection effect of the whole network through the blockchain consensus algorithm. Moreover, each node can choose to make the detection algorithm public to achieve the purpose of automatic detection on the blockchain. Each node can also retain the detection algorithms and disclose only the processing results. If the detection effect of a node is very accurate, then the node is more likely to become a proxy node.

Acknowledgments

This research work is supported by the Innovative Research Groups of the National Natural Science Foundation of China

(No. 61521003), Intergovernmental Special Programme of National Key Research and Development Programme (No. 2016YFE0100300, 2016YFE0100600), National Scientific Fund Programme for Young Scholar (No. 61672470), and Science and Technology Project of Henan Province (No. 182102210617).

References

1. P. A. James, "Computer Security Threat Monitoring and Surveillance," James P. Anderson Co., Fort Washington, 1980
2. M. Padmavathi and R. M. Suresh, "Secure P2P Intelligent Network Transaction using Litecoin," *Mobile Networks and Applications*, Vol. 24, No. 2, pp. 318-326, April 2019
3. I. Bentov and R. Kumaresan, "How to Use Bitcoin to Design Fair Protocol," in *Proceeding of the 34th Annual Cryptology Conference*, pp. 421-439, Santa Barbara, CA, USA, August 2014
4. P. Katsiampa, "Volatility Estimation for Bitcoin: A Comparison of GARCH Model," *Economics Letters*, Vol. 158, pp. 3-6, September 2017
5. Q. Xia, E. B. Sifah, K. O. Asamoah, J. Gao, X. Du, and M. Guizani, "MeDShare: Trust-Less Medical Data Sharing among Cloud Service Providers via Blockchain," *IEEE Access*, Vol. 5, No. 99, pp. 14757-14767, July 2017
6. G. Liang, S. R. Weller, F. Luo, J. Zhao, and Z. Dong, "Distributed Blockchain-based Data Protection Framework for Modern Power Systems Against Cyber Attacks," *IEEE Transactions on Smart Grid*, Vol. 10, No. 3, pp. 162-173, May 2019
7. W. Pennington and J. Evans, "Blockchain-Enabled, Subscriber-based Capital Markets Index Data Distribution," *The Journal of Index Investing*, Vol. 7, No. 4, pp. 83-87, July 2017
8. A. K. Ghosh and A. Schartzbard, "A Study in using Neural Networks for Anomaly and Misuse Detection," in *Proceedings of the 8th Conference on Usenix Security Symposium*, pp. 1-11, Washington, D. C., August 1999
9. S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," (https://bitcoin.org/en/bitcoin-paper, 2008)
10. L. H. Chun, "Classification of P2P Networks and Analysis of Key Technologies", *Microcomputer Information*, DOI 10.3969/j.issn.1008-0570.2008.09.044, 2008
11. G. Wood, "Ethereum: a Secure Decentralised Generalised Transaction Ledger," *Ethereum Project Yellow Paper*, Vol. 32, No. 10, pp. 1365-1367, October 2018
12. EOS.IO technical white paper v2, (https://steemit.com/eos/@eosio/eos-io-technical-white-paper, 2018)
13. L. Lamport, R. Shostak, and M. Pease, "The Byzantine Generals Problem," *ACM Transactions on Programming Languages and Systems*, Vol. 2, No. 3, pp. 382-401, May 1982
14. L. Lamport, "The Part-Time Parliament," *ACM Transactions on Computer Systems*, Vol. 16, No. 2, pp. 133-169, May 1998
15. M. Castro and B. Liskov, "Practical Byzantine Fault Tolerance," in *Proceedings of the Third Symposium on Operating Systems Design and Implementation*, pp. 1-14, New Orleans, USA, February 1999
16. K. Sunny, "A P2P (Peer-to-Peer) Proof of Stake Cryptocurrency," (https://peercoin.net/whitepaper, 2013)