IoT Security Model and Performance Evaluation: A Blockchain Approach

Ziyan Wang[1], Xinghua Dong[2], Yi Li[1], Li Fang[1] , Ping Chen[1]

[1]Beijing University of Posts and Telecommunications, Beijing 100876, China
[2]Industrial and Commercial Bank Beijing branch, China
Contact information of corresponding author: liyi@bupt.edu.cn

Abstract: It is a research hotspot that using blockchain technology to solve the security problems of the Internet of Things (IoT). Although many related ideas have been proposed, there are very few literatures with theoretical and data support. This paper focuses on the research of model construction and performance evaluation. First, an IoT security model is established based on blockchain and InterPlanetary File System (IPFS). In this model, many security risks of traditional IoT architectures can be avoided, and system performance is significantly improved in distributed large capacity storage, concurrency and query. Secondly, the performance of the proposed model is evaluated through the average latency and throughput, which are meaningful for further research and optimization of this direction. Analysis and test results demonstrate the effectiveness of the blockchain-based security model.

Keywords: Internet of Things; Blockchain; Security model; Performance evaluation

1 Introduction

According to the current academic and industry circles, the Internet of Things (IoT) is one of the most promising frontier technologies in the future. However, the security of IoT has obvious limitations, which has not yet reached a convincing solution [1]. With the development of IoT technologies, the application of the IoT continues to expand. Accordingly, security risks and vulnerabilities are being exposed constantly. On December 29, 2016, the U.S. Food and Drug Administration found that there are widespread network security vulnerabilities in wireless embedded medical devices such as pacemakers and insulin pumps in the currently market [2]. The botnet virus named "mirar" in October 2016 caused half of the Internet in the United States to become paralyzed by centralizing the control of IoT devices. The media called this attack "the most serious DDos attack in history"[3]. In general, IoT devices transmit information through various wired and wireless network. Due to the open nature of the network, devices are vulnerable to attacks and the security of the devices' identity is not guaranteed. Moreover, the information transmitted between devices is also vulnerable to intrusion, capture, and tampering, resulting in Dos attacks and DDos attacks [4]. Security problems of the IoT have become an important issue that needs to be resolved in the development of the IoT.

Blockchain technology refers to a distributed shared database in which functional entities within a chain work together to maintain a distributed book whose data records cannot be tamper with and faked. This mechanism was considered to improve IoT security [4]. Meanwhile, the blockchain and the IoT have natural similarity and heterogeneity in distributed features. The blockchain uses the typical point-to-point network. The characteristics of the two determine that if the blockchain technologies are effectively utilized, the IoT application can achieve many network security features such as device privacy protection, information authentication, access control and data encryption [5].

In previous work, the research on IoT with blockchain mainly focused on how to design a combined solution. Thomas and Ned [6] proposed a method for anonymously sharing IoT devices using blockchain technology. Zhao Kuo et al. [7] proposed several practical IoT application scenarios that can use blockchain. Guo Xiong-wei et al. comprehensively considered the technical characteristics of the two and proposed a structural reference framework [8]. However, so far there was no specific application model or implementation, lack of relevant theoretical and data support.

This paper researches the combination of IoT and blockchain from the perspective of model construction and performance evaluation. Firstly, a blockchain-based IoT security model is designed. The model uses the blockchain and InterPlanetary File System (IPFS) [9] to realize four functions: transaction generation, block packaging, blockchain generation and transaction query. The blockchain isolates IoT devices and service providers, thereby eliminating the correlation between the real devices requests and the service providers' responses. As the underlying database, IPFS provides better distributed large-capacity storage, concurrency, and query performance for model. Secondly, related performance testing and evaluations are conducted on the model. The test environment can be changed by setting the number of distributed nodes and transactions. In different test environments, the performance evaluation indicator is the average latency and the throughput. The results show that the blockchain-based IoT security model can avoid many security risks of traditional IoT architectures, and has good performance in storage and query.

978-1-5386-6067-6/18/$31.00 ©2018 IEEE

The remaining of the paper is organized as follows. Section II describes in detail the blockchain-based IoT security model. Section III is the test and performance evaluation of the model in different test environments. Section IV is summaries. Section V is future work.

2 Blockchain-based IoT Security Model

The functions and principles of the blockchain-based IoT security model are described in this section. According to the characteristics of the blockchain and traditional IoT architecture [10], the blockchain-based IoT security model is shown in Figure 1. The model involves three identities, including IoT devices, blockchain nodes, and service providers.

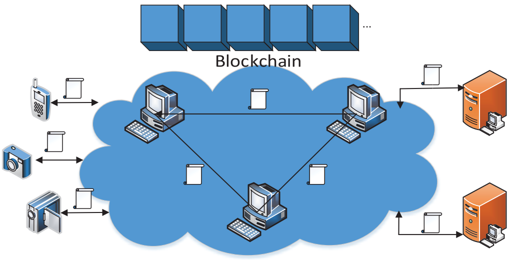

Figure 1 Blockchain-based IoT security model

In the model, IoT devices, blockchain nodes, and service providers all use their own public keys as proof of their identity. A blockchain-based IoT service flow is shown in Figure 2. One, IoT device initiates a transaction and signs it using the private key. The signed transaction contains the device public key, the service provider public key, the service request data, and the signature. Next, the transaction is packaged by the blockchain nodes and sent to the blockchain. Two, the service provider uses its public key to query related transaction records in the blockchain to receive service request and three, write the service response in the same way. Four, the device gets the service response through its public key. Throughout the back and forth process, the private key identifies ownership of the data, and the public key indicates the parties of the transaction.

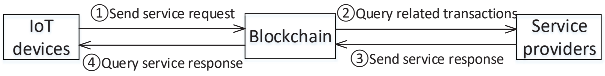

Figure 2 Blockchain-based IoT service flow

The current consensus algorithms mainly include POS, POW for the public chain, PBFT for the coalition chain and mixture of them [5]. This paper combines the mechanism of POS [11] and PBFT [12], assuming that the entire network node is trusted. The device-generated transactions are broadcast to the blockchain network. The blockchain nodes verify them using the public key and corresponding signature. Verified transactions can be processed or they will be abandoned. When transactions reach the threshold, one node is randomly selected from all blockchain nodes, and it is responsible for generating a new block. Similar to generating a transaction, the new block contains the hash value of all valid transactions, the public key and signature of the selected blockchain node. The final chained block contains the new block and the hash values of previous one, thereby ensuring the block generation process information is traceable and the tampering is costly. The referred example is based on the IoT device sending service requests. The service provider sends the service responses are the same way.

According to above description, the functional modules are divided into transaction generation, block packaging, blockchain generation and transaction query, which together form an interface layer of the data.

2.1 Transaction generation

This paper uses the Elliptic Curves Cryptography (ECC) asymmetric encryption algorithm [13] to generate unique identity key pairs for the three model identities. The private key is saved by itself, and the public key is stored in the database as an identifier in the blockchain network.

The transaction generation is used to generate service requests and service responses, like one and three in Figure 2. Using the sender's public key, receiver's public key, service request data or service response data and timestamp as a basic information, the transaction ID is generated through the SHA256 encryption algorithm. The transaction ID and the sender's private key create a signature. As shown in Figure 3. Transaction ID, basic information, and signature make up the transaction.

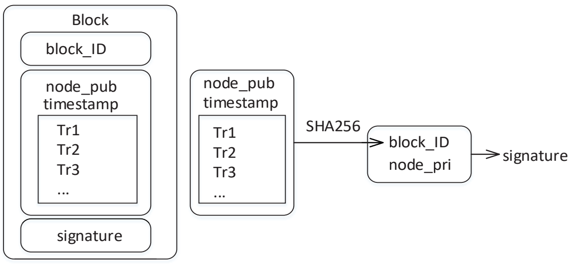

Figure 3 Data structure of transaction

2.2 Block packaging

The block structure is shown in Figure 4. Transactions are broadcast and the blockchain nodes firstly verify them. The verification method is, based on the basic information and the sender's public key, to verify whether the signature matches. If matches, it confirms that the data is indeed sent by the sender.

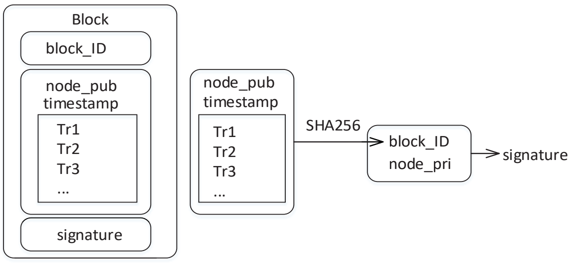

Figure 4 Data structure of block

When block packaging, the elected node writes its own public key into the block. Similar to the transaction generation, the basic information here are multiple transactions' ID, the elected node's public key, and the timestamp. Block consists of block ID, basic information and signature.

2.3 Blockchain generation

As shown in Figure 5, the hash value of the previous block is combined with the new block to form a blockchain. Each transaction of the sender is stored in the blockchain and cannot be tampered with.

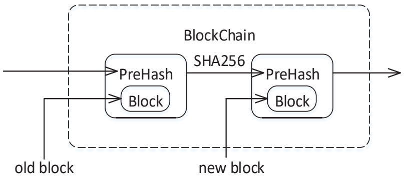

Figure 5 Data structure of blockchain

2.4 Transaction query

The service provider queries the service request based on the public key. Similarly, the device uses the public key to query the blockchain for the service response. The database consists of four tables, which are public key retrieval table, transaction table, block table and blockchain table. Store the relevant information of the corresponding name. Figure 6 shows the relationship between functional modules.

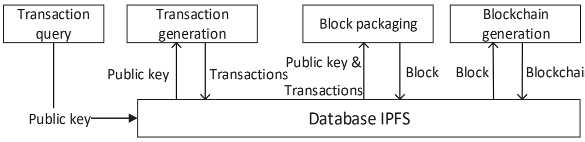

Figure 6 Relationship between functional modules

3 Performance Evaluations

This section is mainly divided into two parts. The first part is the method and process of the experiment. The second part is the performance evaluation of the model through two indicators: average latency and throughput.

3.1 Experimental methods and procedures

LAN segments are constructed through routers to simulate the different areas of the IoT where devices and service providers are located. The experiment does not distinguish between IoT devices and service providers. To avoid the impact of host, deploy two blockchain nodes per LAN segment. The number of nodes in the experiment is set to 6, 8 and 12, respectively.

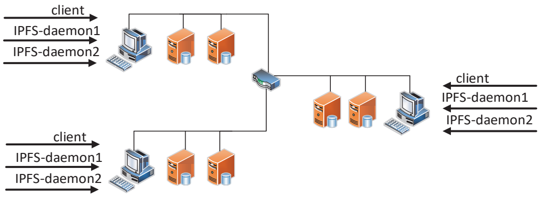

Figure 7 Blockchain network with 6 nodes

Taking Figure 7 as an example, three processes on each LAN, including one client and two IPFS processes, form a point-to-point blockchain network of six nodes.

With reference to [14] [15], some changes have been made to the source code of IPFS. First, the config module is modified so that the experiment nodes can invoke the IPFS access. Second, the warehouse address path of the Swarm module and the local nodes of the IPFS are modified so that the storage and querying of the operation object originated from the experimental nodes, which is in line with the actual needs. Then, in Bootstrap module, configure the boot node address correspondingly. Finally, remove the limit on the number of threads in the IPFS to prevent the upper thread limit that may occur in the experiment.

Yahoo Cloud Services Benchmark Test (YCSB) is used for testing. Modify the YCSB workload module and data interface layer. The transformation of the data interface layer is described in section II. The workload module is used to build different numbers of transactions and test environments. The number of transactions is 100, 200, 500, 1000 and 5000. The test environment is S1 and S2. The transaction operation of S1 is 100% insertion as shown in the one and three of Figure 2. S2's transactions operations are mixed read and write, in which 80% read 20 writes, used to simulate real-world situations that the demand for transaction query is more than write. Include the entire process of Figure 2. The two scenarios are the main scenarios used in the current IoT environment.

3.2 Performance evaluation

The final metric we used to evaluate are the average latency and the throughput of transactions in each of ten independent operations for each group of experiments.

Average Latency

According to the data obtained by YCSB. When the workload are S1 and S2, the average latency of each transaction operation with 6, 8 and 12 nodes is shown in Table 1.

Table I When the workload are S1 and S2, the average latency of each transaction with nodes 6, 8 and 12

	S1			
	Number of nodes			
		6 nodes	8 nodes	12 nodes
Number of transactions	100	0.96	0.88	0.69
	200	1.72	1.54	1.13
	500	2.95	2.77	1.56
	1000	5.78	5.23	3.22
	5000	27.87	24.01	16.34
	S2			
	Number of nodes			
		6 nodes	8 nodes	12 nodes
Number of transactions	100	0.42	0.39	0.31
	200	0.86	0.77	0.65
	500	1.45	1.35	0.96
	1000	2.66	1.69	1.12
	5000	4.82	3.73	3.23

When S1, as the number of transactions increases, the average latency under the same number of nodes shows an increase. When transactions are small, such as 100, the average latency is relatively small and approximate, which is 0.96s, 0.88s, and 0.69s, respectively. After 1000 transactions, the average latency significantly increased. From Figure 8, the greater the number of transactions, the greater the average latency. At 12 nodes, the average latency of 5000 transactions is 5.07 times that of 1000 pens. Look laterally when the same number of transactions, the more nodes the smaller the average latency of each transaction operation. For 1000, the average latency of 6, 8 and 12 nodes are 5.78s, 5.23s, and 3.22s.

As shown in Figure 9, with the increase transactions, the average latency of S2 workload shows a slowdown in growth with the same number of nodes. At 12 nodes, the average latency of 5000 transactions is 2.88 times that of 1000 pens. This is because in S2, the probability of querying the same object is greater. This compares with querying the same object multiple times and IPFS will store the objects' local characteristics while local queries are very fast. When transactions are the same, the more nodes there are, the smaller the average latency is. The query operation is more, the model performed is better.

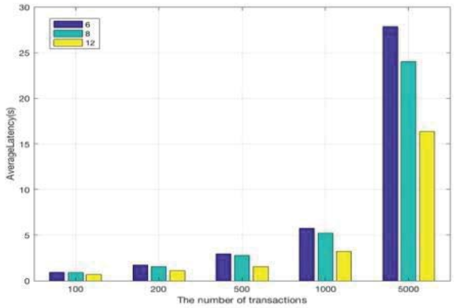

Figure 8 When S1, comparison of average latency for each transaction with nodes 6,8,12

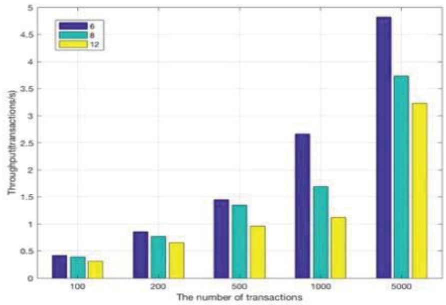

Figure 9 When S2, comparison of average latency for each transaction with nodes 6,8,12.

Throughput

The throughput of 6, 8 and 12 nodes under S1 and S2 is shown in Table 2. Figure 10 is a S1 comparison of broken lines. At the same nodes, when the transactions increase to around 500, the model's throughput reaches a large value within the fluctuation range. After 500 transactions, the throughput no longer increases as the number of transactions. In the same transactions, there is a significant throughput difference between different nodes. For 500 transactions, the throughput of 12 nodes is 1.89 times that of 6 nodes.

Table II When the workload are S1 and S2, the throughput with nodes 6, 8 and 12

		S1		
		Number of nodes		
		6 nodes	8 nodes	12 nodes
Number of transactions	100	104.17	113.64	144.93
	200	116.27	129.87	176.99
	500	169.49	180.51	320.51
	1000	173.01	191.20	310.55
	5000	179.40	208.24	306.00
		S2		
		Number of nodes		
		6 nodes	8 nodes	12 nodes
Number of transactions	100	230.10	256.41	322.58
	200	232.56	259.74	307.69
	500	344.82	370.37	520.83
	1000	375.94	591.72	892.86
	5000	1037.34	1340.48	1547.99

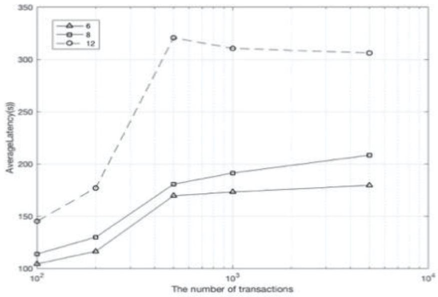

Figure 10 When S1, comparison of throughput with nodes 6,8,12

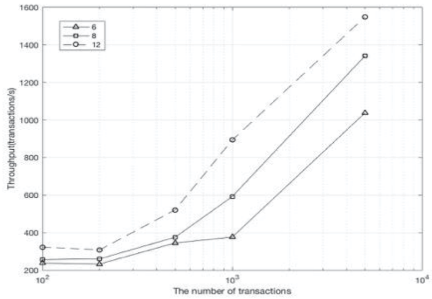

Figure 11 When S2, comparison of throughput with nodes 6,8,12

Figure 11 is a S2 comparison chart of broken lines. At the same node, the number of transactions increased from 100 to 5,000, and the throughput decreased first and then increased. In 200 transactions, the throughput slightly decreased. As the transactions continue to increase, the throughput shows an increasing trend. This is related to the threshold setting of block packing. The packing process includes node' initialization and block initialization. It is not difficult to predict that due to the IPFS query mechanism, the more query operations the model contains, the greater the throughput. Limiting factors will appear in hardware performance.

4 Summaries

This paper designs a blockchain-based IoT security model and evaluates the performance of the model. The simulation results show that the model can provide good performance support in terms of the average latency and the throughput. In addition to this, the heavier the proportion of query operations in execution process, the better the performance of the model, which is in line with the actual situation of most people in the life as replicators rather than creators.

The blockchain technology used in this model provides the possibility for decentralization of the IoT. Data is no longer controlled by a single cloud service provider. The traditional IoT-centered service architecture has tremendous pressure on computing, storage and bandwidth as the number of devices grows geometrically. The model uses blockchain to provide point-to-point data transmission, effectively utilizing the distributed resources of each node. At the same time, the distributed architecture of the model can avoid the single point failure, DOS attack and DDOS attack caused by the centralized service architecture of the IoT [10]. Besides, the key pair is used as proof of identity and all transmitted data is signed and encrypted in the model. The devices' data and privacy will be more secure. Once written into the blockchain, there is no possibility of being illegally tampered with and lost, thereby increasing the reliability of the entire system. This is something that traditional IoT architecture cannot provide.

In terms of databases, blockchain nodes mostly participate in transaction processing in the network with the identity of peer nodes. They do not do optimization design for high concurrent services and thus cannot support high concurrent access. Traditional blockchain databases increase the throughput, concurrent traffic, and storage capacity of the system linearly by scaling the number of nodes horizontally. At present, most blockchain platforms decrease their overall system performance as the number of nodes increases [16]. The model uses IPFS to change from full to partial storage and can connected to the same file system independent of the central server. Moreover, model-stored objects are based on content addressing. It reduces redundancy and increases network space. The relevant backups on valid paths significantly improve the query speed of the model. What needs to be raised is, compared to the traditional IoT architecture, this model sacrifices some system overhead (memory, CPU, etc.) in exchange for security, which is an inevitable problem with blockchain technology.

5 Future Works

The security model and the evaluation performance given in this paper provide a reference for further research on using blockchain to solve the security problems in the IoT. In the future work, it is necessary to further improve the election of nodes in the model and consider adding incentives to make it more fair and efficient. In addition to this, more network environments need to be built to perform various tests and optimize the performance of the model. Finally, since IPFS can only store invariant objects, dynamic objects can only be stored via InterPlanetary Name Space (IPNS), which is also a problem worthy of study.

References

[1] Zhang Yuqing, Zhou Wei, Peng Ani. Internet of Things Security Overview [J]. Computer Research and Development, 2017, 54 (10): 2130-2143.

[2] MARC GOODMAN. Hacking the HumanHeart [EB/OL].[2017-04-24].http://bigthink.com/future-crimes/hacking-the-human-heart.

[3] Wikipedia. 2016 dyn cyberattack [EB/OL]. [2017-05-09]. http://en.wikipedia.org/w/index.php?title=2016_Dyn_cyberattack&oldid=763071700.

[4] HE Yu-Jun, GONG Guo-Cheng.Research on the Technology of Block Chain in Security of IoT [J] .Telecommunications Engineering Technology and Standardization, 2017,30 (5): 12-16..

[5] Zhu Yan, Gan Guohua, Deng Di, et al.Study on security in key technologies of blockchain [J] .Information Security Research, 2016, 2 (12): 1090-1097

[6] Hardjono T, Smith N. Cloud-Based Commissioning of Constrained Devices using Permissioned Blockchains[C]// ACM International Workshop on Iot Privacy, Trust, and Security. ACM, 2016:29-36.

[7] ZHAO Kuo, XING Yongheng. Security Survey of Internet of Things Driven by Block Chain Technology [J].Netinfo Security, 2017(5):1-6..

[8] GAO Xiong-wei, YAN Bin-Feng. Block-chain thinking, IOT blockchain and its reference frame and application analysis [J] .Telecommunications Technology, 2017 (5): 61-65.

[9] J. Benet, "IPFS – Content Addressed, Versioned, P2P File System," Protocol Labs, Inc., Tech. Rep., 2014.

[10] Yang Guang, Geng Guining, Du Yu, Liu Zhaohui, Han He. Security threats and measures of Internet of Things [J]. Journal of Tsinghua University (Science and Technology), 2011, 51(10):1335-1340.

[11] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system. White Paper, 2008.

[12] Cachin C. Architecture of the hyperledger blockchain fabric//Workshop on Distributed Cryptocurrencies and Consensus Ledgers. 2016

[13] Blake, G. Seroussi, and N. Smart, editors, Advances in Elliptic Curve Cryptography, London Mathematical Society 317, Cambridge University Press, 2005.

[14] B. Confais, A. Lebre and B. Parrein, "Performance Analysis of Object Store Systems in a Fog/Edge Computing Infrastructures," 2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom), Luxembourg City, 2016, pp. 294-301.

[15] B. Confais, A. Lebre and B. Parrein, "An Object Store Service for a Fog/Edge Computing Infrastructure Based on IPFS and a Scale-Out NAS," 2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC), Madrid, 2017, pp. 41-50. doi: 10.1109/ICFEC.2017.13

[16] SHAO Qi-Feng,JIN Che-Qing,ZHANG Zhao,QIAN Wei-Ning,ZHOU Ao-Ying,Blockchain: Architecture and Research Progress, 2017, Vol.40,Online Publishing No.157