Data Management and Searching System and Method to Provide Increased Security for IoT Platform

Moon Yong Jung
Department of Electronic Engineering
Sogang University
Seoul, South Korea
myjung@sogang.ac.kr

Ju Wook Jang
Department of Electronic Engineering
Sogang University
Seoul, South Korea
jjang@sogang.ac.kr

***Abstract*— Existing data management and searching system for Internet of Things uses centralized database. For this reason, security vulnerabilities are found in this system which consists of server such as IP spoofing, single point of failure and Sybil attack. This paper proposes data management system is based on blockchain which ensures security by using ECDSA digital signature and SHA-256 hash function. Location that is indicated as IP address of data owner and data name are transcribed in block which is included in the blockchain. Furthermore, we devise data manegement and searching method through analyzing block hash value. By using security properties of blockchain such as authentication, non-repudiation and data integrity, this system has advantage of security comparing to previous data management and searching system using centralized database or P2P networks.**

***Keywords*— Internet of Things, Blockchain, Proof-of-Work, Data Management, Signature, Hash function**

I. INTRODUCTION

In IoT(Internet of Things) environment, the amount of data in world is rapidly increasing, because heterogeneous sensor devices are being developed. According to a recent report, the expected amount of data will be up to 4.4 trillion gigabytes by 2020[1]. This enormous set of data is composed of atypical types. Therefore, it is very important to sort out that kind of data.

However, it is a problem when it comes to data management in IoT platform. Currently, billions of devices are being connected to each other and the data within them are stored and analyzed every day. Since there are such a large amount of data, it is virtually impossible to find specific data is wanted by a particular user. As a result, it takes some doing to provide efficient service for managing and customizing data. IoT also has a problem regarding data security. Most of the generated data are usually managed by certificate management system and are stored in the central server for analyzing data. Therefore, IoT platform is at risk for information leakage and falsification due to serious vulnerabilities of security of central database. Also as shown in figure 1, if the central server is paralyzed or is attacked by outsiders, massive data loss can occur.

IoT platform using centralized server have security vulnerabilities [2] (e. g. NRS system [3]). Vulnerable point of IoT platform is as follows.

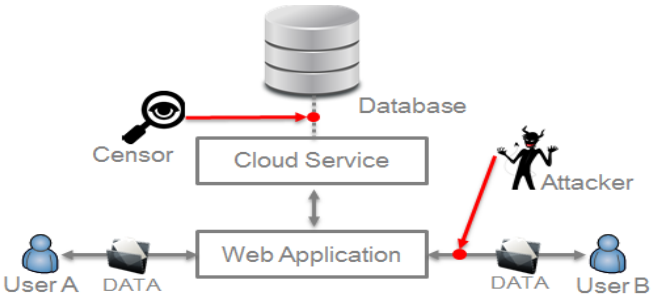

Fig. 1. The Problem with central server

1) IP spoofing [4]: If bad device transmits message that is maliciously changed to its IP address to server, then node cannot acquire right information of IP address of owner who creates data, because wrong information is stored in the server.

2) Sybil attack [5]: When a bad node falsifies itself to a lot of node in the IoT platform, problem that is changed all data in the server is occurred in the system.

3) Single point of failure [6]: If node maliciously changes the data of server by attacking centralized server, every node has misinformation. This system relies on a centralized server to store and manage mapping table for mapping an IP address to data name.

In order to solve above mentioned problems, the distributed storage system (e. g. P2P network system) has been developed and the related studies are in the progress [7]. However, P2P network system also has its own problems of security. For example, if malicious node is falsified to multiple node takes Sybil attack, it would cause the problem that data is unreliable in whole P2P network [8].

Accordingly, we have combined the blockchain [9], security distributed database, with data storage and management system. Blockchain is based technique of Bitcoin, digital currency, which always maintains a continuously growing list of transaction record. When new transaction record is occurred, this record is broadcasted to all node is participated in the blockchain. Then, these nodes attempt to generate block using “Proof-of-Work”. A node that succeeded in generating the block for the first time propagates information of generated block. As blocks is connected to each other in sequence, this connection becomes blockchain.

Application technology using blockchain algorithm is utilized to store information regarding trading record. However, in this paper, we utilize blockchain for data management and searching system as opposed to store information of trading. A data is composed of information such as data creation, modification and deletion in the blockchain. This data is called "transaction". By using blockchain, It is easy to sort out by recording information of data in the block, and it solves problems which are security vulnerabilities of data management system of existing IoT platform such as IP spoofing, Sybil attack and single point of failure through security properties of blockchain as authentication, non-repudiation and data integrity,.

In this paper, we address our blockchain-based data management and searching system. In chapter 2, we explain fundamental properties of blockchain. In chapter 3 and 4, how we develop this system and show simulation results. Lastly, chapter 5, we mentioned conclusion.

II. PROPERTIES OF BLOCKCHAIN

A. *Blockchain*

The blockchain network timestamps transactions by hashing them into an ongoing chain of hash-based Proof-of-Work, forming a record that cannot be changed without redoing the Proof-of-Work [9]. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. All nodes participating in blockchain network store duplicated form of blockchain.

B. *Transactions*

Each owner of digital coin like Bitcoin transfers the coin to the by digitally signing a hash of the previous transaction and the public key of the next owner and adding these to the end of the coin [9]. A payee can verify the signatures to verify the chain of ownership.

C. *Proof-of-Work*

Blockchain network implements the Proof-of-Work by incrementing a nonce, a random variable, in the block until a value is found that gives the block's hash the required zero bits [9]. Once the CPU effort has been expended to make it satisfy the Proof-of-Work, the block cannot be changed without redoing the work. As later blocks are chained after it, the work to change the block would include redoing all the blocks after it. To compensate for increasing hardware speed and varying interest in running nodes over time, the Proof-of-Work difficulty is determined by a moving average targeting an average number of blocks per hour. If they are generated too fast, the difficulty increases.

D. *Network*

The steps to run the network are as follows [9]:

1. New transactions are broadcast to all nodes.
2. Each node collects new transactions into a block.
3. Each node works on finding a difficult Proof-of-Work for its block.
4. When a node finds a Proof-of-Work, it broadcasts the block to all nodes.
5. Nodes accept the block only if all transactions in it are valid and not already spent.
6. Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

III. SYSTEM DESIGN OVERVIEW

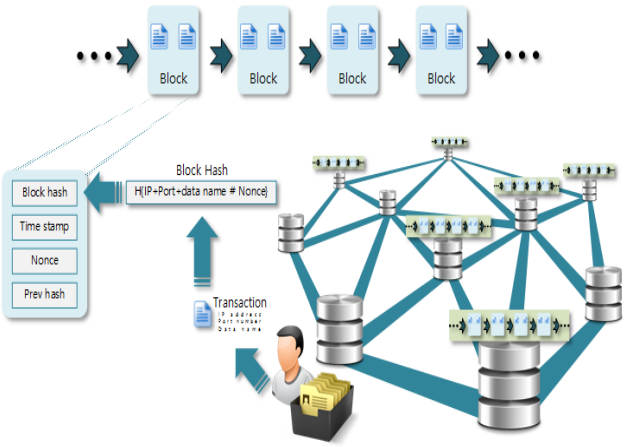

Fig. 2. Overall data management and searching system

As shown in Figure 2, it represents the blockchain-based data management and searching data owner system. If user generates data in specified directory, information of data such as data name, IP address and TCP port number of data owner is converted into ASCII character encoding. Then, hash value is created by hashing data name, IP address and port number using SHA-256 hash function. Through this, it creates transaction in order to transmit it to all nodes participating in the blockchain network.

Each node receiving transaction performs Proof-of-Work [9] for generating new block. Proof-of-Work is iteration process to find nonce to satisfy required zero bits by applying SHA-256 hash function to data which is combined by transaction includes IP address, TCP port number, data name, previous block hash and nonce. The node which succeed to find legitimate hash value by doing Proof-of-Work transmits block that includes hash value and nonce to all nodes. The block is connected to the main blockchain after each node receiving block certifies validity of block.

Also, how to check the owner of data stored in the blockchain is as follows. All blocks are stored as JSON format and searcher checks block hash values. When searcher input data name to blockchain, blockchain-based data searching system is able to find data owner through combining data name, IP address and port number. Finally, searcher can find the data owner and can require data.

IV. CONSTRUCTION OF DATA MANAGEMENT AND SEARCH SYSTEM

We have developed data management and searching system which has excellent security performance, because existing data management system of IoT platform is exposed to security vulnerabilities. Existing technology using blockchain is usually used to store trading record. Instead of, we apply security

properties of blockchain to data management system. For effective explanation, we assume that each node has fixed IP address environment such as office and laboratory, and knows each other's IP address.

For generating block in the blockchain, we use algorithm which have three step process. First, each node that generates data sends factors which are name of data, IP address and TCP port number of owner to other nodes. This data is called the transaction. Next, the node is owner of data generates signature before owner sends transaction to other nodes. This signature which is included in the transaction is transmitted to other nodes. The last step is generating block through the process that is Proof-of-Work. We will explain the results according to the process described above.

A. *transactions*

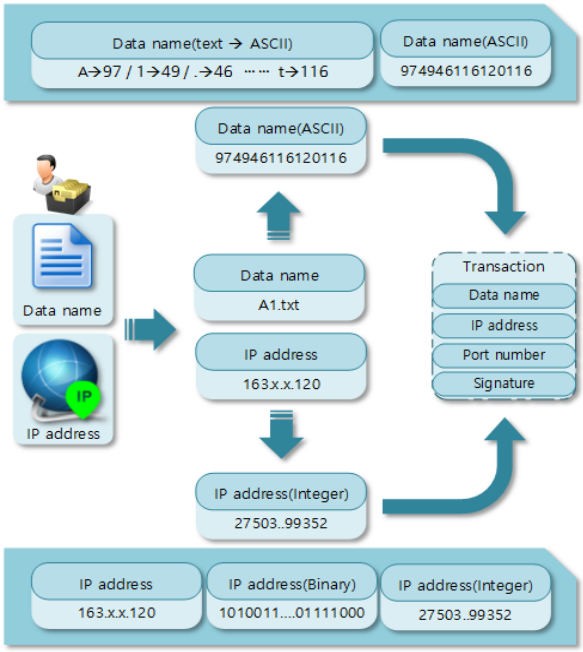

Fig. 3. Process of generating transaction.

As shown in Figure 3, it illustrates a process of generating a transaction. A node is ready to generate the transaction by extracting the name of generated data and owner information which are IP address and port number. The name of generated data which is represented by character converts to a number in order to use a SHA-256 hash function. As shown in the figure above, using the ASCII character encoding, a1.txt which is data name is converted numeric data such as 974946116120116. In order to list the data which are IP address and name, IP address of owner also have the similarly conversion process. For example, when IP address is 163.239.195.120, IP address is listed as a binary number. This converted number is represented as 1010011….01111000. And then, the binary number is, after being converted to a decimal number, included in the transaction. This case is represented by 2750399352. A data that is IP address which is applied to this conversion process is combined with the TCP port number of the node is included in the transaction. It simply lists the IP address and port number, such as 275039935212000. Transaction that is generated by the combination of data name, IP address and port number such as 27503993521200097494611612011 6 is transmitted to all other nodes participating in the blockchain networks.

B. *Signature*

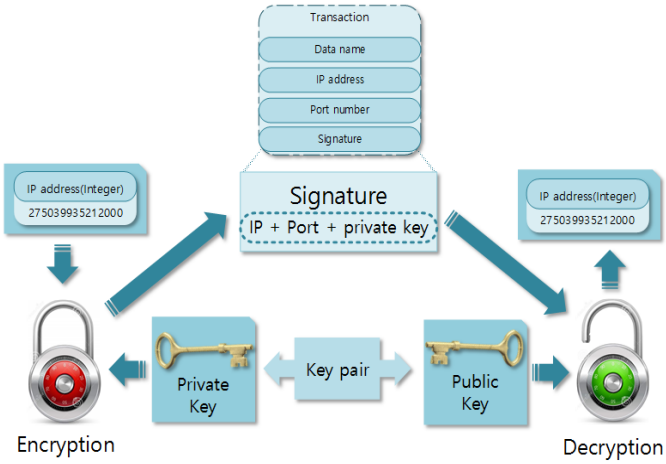

Fig. 4. Process of creating signature.

Before sending the transaction, the node generates the signature using its private key. Algorithms in order to generate the signature are RSA [10] and ECDSA [11], and we uses ECDSA algorithm. Each node has its private key and public key, and has public key of other nodes. For cryptography, the node encrypts the message by using the public key of another node. However, the node encrypts message with its private key in order to generate a signature. The signature is generated by combination of transaction which are data name, IP address and port number and ECDSA private key in order to apply to this system. Signature which is included in transaction is transmitted to other nodes, and the receiving node is able to certify the validity of transaction by decrypting signature with public key of transmitting node. If signature verification result is true, then the transaction is verified that transmitting node is not changed.

By using signature, this system is secured from IP spoofing and Sybil attack of malicious node. As each node is able to certificate identity of other nodes through encryption and decryption process of signature, this data management system prevents to combine data name with maliciously modified IP address thereby the IP spoofing of bad node. Similarly, this data management service system is able to defend Sybil attack. When a malicious node that falsifies as a lot of nodes transmits signature, an honest node get a result is false by decrypting signature using public key of falsified node because signature is created by unique private key.

The signature is transmitted while being included in transaction, and receiving node decrypted the signature using public key of transmitting node. As shown in Figure 5, after receiving transaction, it is the result of decryption process. Fig. 5-1 is simulation result of transmitting node and Fig. 5-2 is simulation result of receiving node. Fig. 5-1 denotes factors of transaction that includes data name, IP address, port number and signature. In addition, it was also appearance of the private key of data owner. But, the private key is not actually transmitted. Fig. 5-2 denotes process of verification of transaction. After

receiving node checks the IP address and port number of the transmitting node, Recipient decrypts the signature using public key by checking the list that stores IP address, port number and public key. This case, the result of decryption is 275039935212000 that is message (IP address + port number). By checking result of decryption is true, verification of signature of transmitting node is completed.

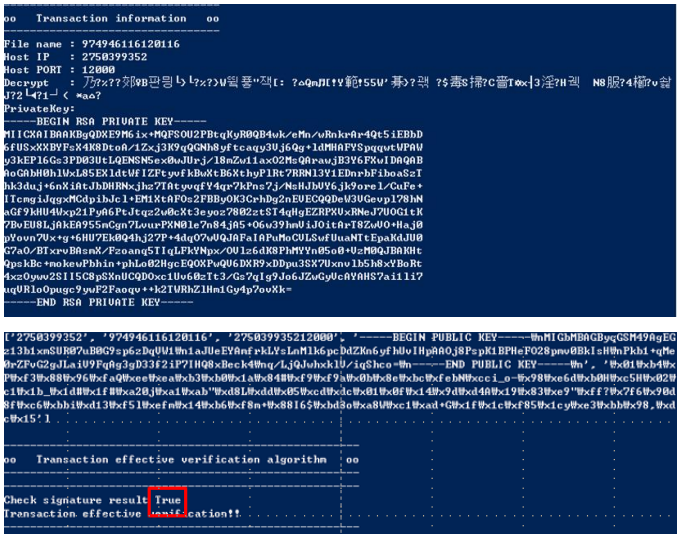

Fig. 5. Simulation result (process of generating signature)

C. *Generating block*

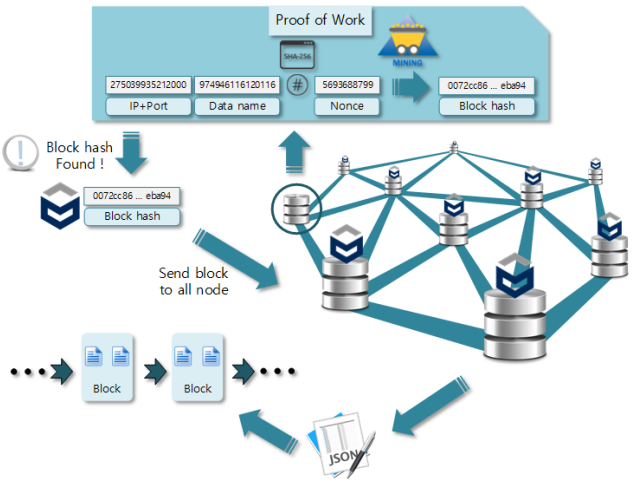

Fig. 6. Process of generating block

As shown in Figure 6, it illustrates a process of generating a block. The node receiving the transaction executes Proof-of-Work to generate block. We also design to similar difficulty with original Bitcoin blockchain using same number of zero bits. But, the number of transaction in the block is different to original blockchain. A block in the original blockchain has many transaction. However, a block in this blockchain for applying to data management system has only one transaction. So, one information of owner and one data name are stored in a block. This approach has several advantages. We know easily that the number of files which are able to share with other nodes are participated in blockchain only by counting the number of blocks. And, searcher can find easily data owner because searcher only need to analyze block hash value without the need to analyze a lot of transactions inside the block. Also, Office or laboratory members that use this data management system will only store data for sharing on specific directory, therefore, the time at which a block having only one information of data owner is generated can sufficiently cover the time at which data to be shared is generated.

It is able to discover owner information of data, when we search data name by analyzing only block hash, since the block hash is made up of factors which are combined with IP address, port name and data name. Block hash consist of hash algorithm result between factors and nonce in progress. Nonce is a random variable that apply to SHA-256 hash function with factors in order to find hash value which is smaller than standard hash value which is set up to control difficulty of generating block such as 0x00000…12a2dcf8, and this process that finds legitimate nonce is called Proof-of-Work or mining. The reason for using Proof-of-Work is to hide the node that will generate block in order to prevent an attack of malicious nodes, because the node that generates the block hash is selected at random. Finding hash value is smaller than standard hash value is similar to a first-preimage attack. In a first-preimage attack, a node knows a hash value but not the message that created it, and a node wants to discover any message with the known hash value [12]. For finding perfectly same hash value mentioned above, it takes a lot of time. When hash length is denoted as "L", A first-preimage attack allows an attacker who knows a desired hash value to find a message that results in that value in fewer than 2^L attempts [12]. In case of SHA-256 hash function, the node should try 2^{256} times. A time which finds hash value is satisfied with the number of zero bits is faster than a time which finds perfectly same hash value. However, level of difficulty is able to be increased by increasing the number of zero bits of hash value. In this way of generating block, this data management system is able to select a node which generates block randomly, because it does not know which node generates the block as level of difficulty is increased.

Fig. 7. Simulation process to generating block

The node that generates block hash broadcasts information of block such as block hash and nonce to all nodes participating in blockchain network. After a node receiving block hash conducts process that is verification of block, and receiving node add to block in their blockchain.

For making block, the node collects information of transaction. As shown in Figure 7, it denotes process to generate block. Every nodes are ready to create block, because both transmitting node and receiving node obtain data name, IP address and port number from information of transaction. ‘hashReady’ which is a variable of python code is a list information for creating hash value. This variable consists of previous block hash, IP address, port number and nonce. If the node finds appropriate nonce value that is able to make block hash value, then that node announce value of block hash to other nodes which are participated in data management system using blockchain.

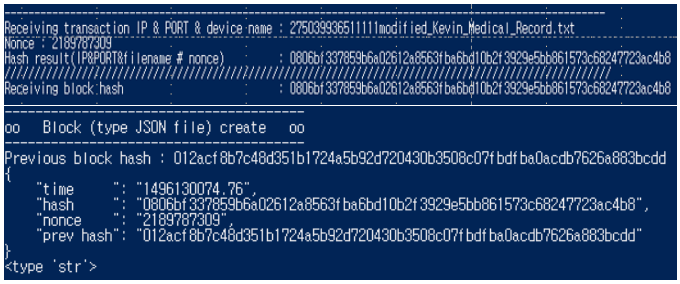

Fig. 8. Generating block

As shown in Figure 8, it denotes generating block. A node which receives block hash value and nonce from transmitting node checks a fact that block hash value corresponds to result of SHA-256 hash function of transaction and nonce. If above condition is satisfied, then the block is connected to blockchain. The block is composed to four factors which are timestamp indicates generating time of block, block hash value, previous hash value and nonce. This block is stored as JSON file, and data management system retrieves JSON files are stored in personal database when this system searches data name.

D. Data searching method and system

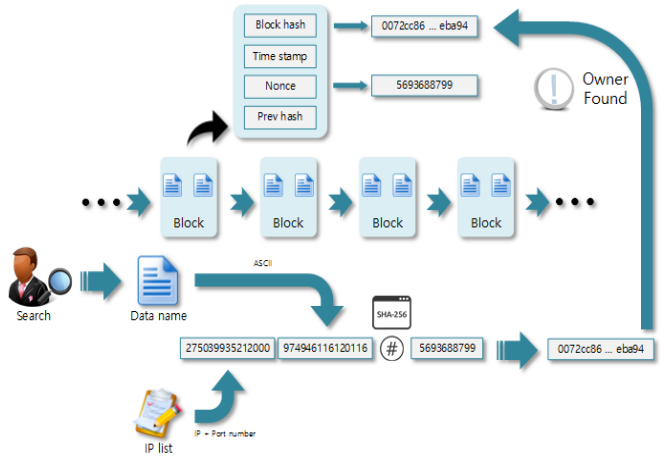

Fig. 9. Search method

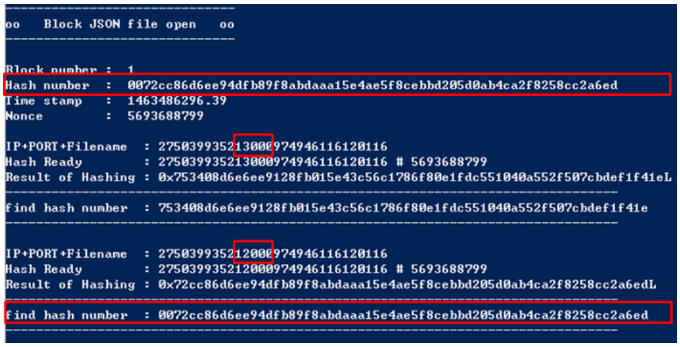

Fig. 10. Simulation result (Finding data owner to use searching algorithm)

As shown in Figure 9, it denotes process of searching data owner. As shown in Figure 10, it is the result of searching simulation. A searcher who want to find owner of data writes data name on data searching system is based on blockchain. This proposed searching system lists factors which are IP address, port number of all nodes which are participated in the blockchain network and data name which is converted to ASII character encoding. Then, the searching system extracts nonce from a block and calculates hash value with nonce and listed factors. The system checks that hash value of recorded block corresponds to result of hash value which is calculated by using listed factor and nonce, and is able to speculate that owner is formulated to IP address in the block has data name which is requested by searcher when calculated hash value equal to block hash value. If not same, system proceeds in the same way by analyzing next block over and over again.

A searcher calculates hash value using data name and factors which are IP address, TCP port number and nonce in the block. In this case, since this system does not compute inverse hash function, it does not take a lot of time to check owner of data by analyzing all blocks which is stored to JSON files in the blockchain.

V. CONCLUSION

Blockchain improves the security of the distributed data storage system. This paper has proposed the blockchain-based data management and searching data system. It is shown from results of simulation that the developed system outperforms the previous distributed storage system. This system provides that it is easy to manage numerous data and security to prevent IP spoofing, Sybil attack and single point of failure.

Signature that is encrypted information using private key is able to guarantee authentication which is the process of checking user identity. For this reason, it can solve security problem that is IP spoofing by checking validity of signature. In addition, generator of signature is not able to repudiate a fact which node generates signature and transaction, because signature is created by private key cannot be known to other users. It is called non-repudiation. Sybil attack is an attack wherein a repudiation system is subverted by forging identities in P2P networks. Therefore, signature prevents that malicious node disguises as honest node. Also, obviously, this data management and searching system is based on blockchain prevents to security problem is single point of failure that occurs from centralized database. Thus, this system has advantage of security comparing

to previous data management and searching system using centralized database or P2P networks in the IoT platform. In the future work, we build a system that provides effective security by applying blockchain to the IoT plaform. For example, the location of IoT devices can be checked by storing the mapping table of combining the UUID which is an extracting result from IoT devices which are connected to a gateway and the IP address of the gateway in the blockchain. Therefore, it is able to support IoT platform by supplying shortest path between IoT devices.

ACKNOWLEDGMENT

The research was supported by the Institute for Information & Communications Technology Promotioni (IITP) grant funded by the Korean government (MSIP) (No. 2015-0-00183, A Study on Hyper Connected Self-Organizing Network Infrastructure Technologies for IoT Service).

REFERENCES

[1] IDC .The Digital Age of Opportunities: The seventh digital universe study, April 2014.

[2] Abdelmalek, A., M. Feham, and A. Taleb-Ahmed. "On Recent Security Enhancements to Autoconfiguration Protocols for MANETs: Real Threats and Requirements." International Journal of Computer Science and Network Security 9.4 (2009): 401-407.

[3] Balakrishnan, H., Lakshminarayanan, K., Ratnasamy, S., Shenker, S., Stoica, I., & Walfish, M. (2004, August). A layered naming architecture for the internet. In ACM SIGCOMM Computer Communication Review (Vol. 34, No. 4, pp. 343-352). ACM.

[4] Tanase, Matthew. "IP spoofing: an introduction." Security Focus 11 (2003).

[5] Douceur, J. R. (2002, March). The sybil attack. In International Workshop on Peer-to-Peer Systems (pp. 251-260). Springer Berlin Heidelberg.

[6] Lynch, Gary S. Single point of failure: The 10 essential laws of supply chain risk management. John Wiley and Sons, 2009.

[7] Lakshman, Avinash, and Prashant Malik. "Cassandra: a decentralized structured storage system." ACM SIGOPS Operating Systems Review 44.2 (2010): 35-40.

[8] Newsome, J., Shi, E., Song, D., & Perrig, A. (2004, April). The sybil attack in sensor networks: analysis & defenses. In Proceedings of the 3rd international symposium on Information processing in sensor networks (pp. 259-268). ACM.

[9] Nakamoto, Satoshi. "Bitcoin: A peer-to-peer electronic cash system." (2008): 28.

[10] Rivest, Ronald L., Adi Shamir, and Leonard Adleman. "A method for obtaining digital signatures and public-key cryptosystems." Communications of the ACM 21.2 (1978): 120-126.

[11] Johnson, Don, Alfred Menezes, and Scott Vanstone. "The elliptic curve digital signature algorithm (ECDSA)." International journal of information security 1.1 (2001): 36-63.

[12] Hoffman, Paul, and Bruce Schneier. Attacks on cryptographic hashes in Internet protocols. No. RFC 4270. 2005.