Secure Provenance-Aware Internet of Things

Nidhi Patel
College of Informatics
Northern Kentucky University
pateln21@nku.edu

Dharmang Hansaliya
College of Informatics
Northern Kentucky University
hansaliyad1@nku.edu

Kena Patel
College of Informatics
Northern Kentucky University
patelk20@nku.edu

***Abstract*—A network is a digital telecommunication between different nodes which enables them to share resources and exchange information. These nodes could be any digital device like computer or networking equipment's like routers or switches. The network of physical devices, vehicles, home appliances, and other items embedded with electronics, software, sensors, actuators, and connectivity which enables these things to connect, collect and exchange data are called Internet of Things. IoT devices are digital nodes which are designed in a such a way that it blends into surrounding environment. As the number of IoT devices in our environment increases, increases the problem of secure provenance. According to Google dictionary, provenance means the place of origin or earliest known history of something. The word "Provenance-Aware" strongly suggest a secure way of generating, storing, and retrieving data in the process of service delivery. This data can be used for any purpose like, authentication, authorization, identification, accountability verification, etc. In this paper, we discuss about provenance chain in IoT devices using blockchain technology.**

Keywords—Provenance, Blockchain, Internet of Things (IoT), BigchainDB, Tendermint, JWT (Json Web Token).

I. INTRODUCTION

With new innovations and inventions in technological field, number of digital electronic devices are increasing day by day. In 2010, the number of devices connected to the Internet was bigger than the population of the earth. In current era, the manufacturing cost of these connected devices are low because of which the number of digital devices will keep growing. There was a time where these devices were only used for computing and personal computers. Today, it is used everywhere. These devices are in cars, in your refrigerator, in your home security system and other home automation devices. The possibilities and capabilities of these devices are endless. The primary fuel of the vision of the internet of things is continuously decreasing manufacturing cost of sensors and actuators. These IoT devices autonomously exchange information and data to make our everyday life easier. At the same time, it makes boundaries between the cyber and the physical worlds even more blurred.

It is a trend of Information Technology, every new technology in IT comes with new challenges and problems and IoT brings such challenges which cannot be resolved with traditional security design for internet. In this paper, we will discuss about data security using provenance chain using blockchain technology.

IoT devices are widely used in industrial, health, agricultural and many other sectors beside home. Wide range of usage of IoT devices, makes it deal with huge amount of objects that generates and uses different data and entities. Provenance can be used to detect the creation and propagation of data generated by all of these IoT devices. As mentioned in paper, Security & privacy in IoT Data Provenance, "Provenance-aware system has the ability to keep track of ownership of data, origin from when and where, It can be defined as the process of detecting the lineage and the derivation of data and data objects data provenance provide wide range of application like estimating the level of quality and trust of data, audit trail, debugging, reusability and reproducibility, and analyzing performance bottlenecks. It can also be used in areas such as ownership, security, citation, and copyright". [11] As you can see, data provenance is important with IoT devices. If I summarize the requirement of provenance-aware system, system requires below quality:

- Quality and trustworthiness of the data
- Security of the data
- Traceability of generated data
- Ownership of the data

So, our approach to this problem was blockchain technology.

The name, "Blockchain", clearly represents itself. Blockchain is a chain of blocks. A block in a blockchain is made up of data, hash and hash of previous block. First item in a block is data. This data could be anything depending on a type of blockchain. Blockchain of bitcoin contains transaction information as a data. Second item in a block is a hash of a block which identifies a block itself, and all of its contents and is always unique. If any data in block is changed, it's hash is recalculated. Changing something in block will change its hash. Third item in a block is hash of previous block. This creates a chain of block and strengthens the blockchain security. The basic blockchain is shown in figure 1.

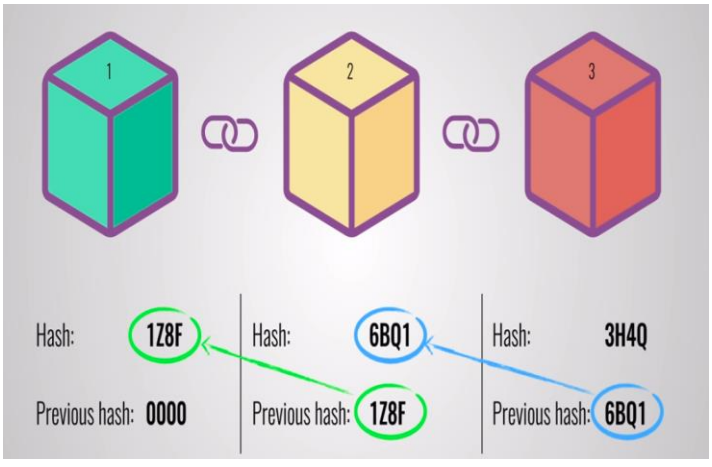

Figure 1. Blockchain

As you can see, block three's previous hash is the hash of block two and block two's previous hash is the hash of block one. This is how blocks are chained together. Let's suppose, an intruder tries to change the data contained in block two. This will change the hash of block two because that hash is generated by encrypting the data of block two and previous hash contained in block two. Since data is changed in block two, which indeed change the hash of block two, previous hash mentioned in block three is no longer same. This will break the chain.

As you can see, blockchain, can provide quality and trustworthiness of data. It can also provide little security and trackability. But, using just blockchain, we cannot find the ownership of the data. In addition to that, blockchain technology is vulnerable to 51% attack[12]. Also it on P2P network which means anyone can join the network which we don't want. In addition to that, blockchain requires verification of chain every time someone adds data to it, which requires lots of compute power and storage which IoT device cannot provide. So, we came up with different approach which involves blockchain.

In this paper, we discuss about BigchainDB, which is different approach to achieve provenance-aware system involving blockchain technology. We will discuss all components of BigchainDB and create a system model of provenance-aware system using BigchainDB technology. To better understand the system model, we will also discuss the data follow in our proposed provenance-aware system.

The reminder of this paper is organized as follows:

- Section 2, challenges with data provenance in IoT system: In this section we will discuss about all the challenges faced during the implementation of provenance-aware system for IoT environment.
- Section 3, our solutions towards challenges: In this section, we will discuss about the solution to the challenges which we will discuss in previous section.
- Section 4, System model description: In this section, we will use our solution and implement it to generate provenance-aware IoT environment.
- Section 5, Life of a Data in our proposed system model: In this section, we will discuss the life cycle of a data in our proposed system model.
- Section 6, Data read and write matrices: In this section, we will present the read and write matrices of BigchainDB.
- Section 7, Conclusion: In this section, we will conclude our findings.

II. CHALLENGES WITH DATA PROVENANCE IN IOT SYSTEM

In this section, we will discuss about challenges that came across with the implementation of data provenance model in IoT infrastructure. The resolution to these challenges discussed here guarantees the fulfilment of all qualities of provenance-aware system discussed in introduction section. Some of the challenges with provenance-aware system in IoT infrastructure are:

- Data security
- Storage and processing of data
- Indexing generated data in provenance-aware system in IoT infrastructure
- Identification of entity who generates data (Ownership)

A. Data Security

IoT devices generate huge amount of data because of wide and big IoT network in any given sector which involves multiple nodes of data generation. Some of the data generated by IoT devices has the quality of data sensitivity which increases the importance of data security. Furthermore, unauthorized access to data leads to exploitation of generated data and confidentiality. In addition to that, Secure data increases the trustworthiness of data. As mentioned earlier, trustworthy of data is one of the requirement of data provenance. The solution of this challenge is addressed in our proposed solution.

B. Storage and processing of data

Provenance not only means data trustworthy and data security. Provenance also means traceability of data origin and history. For which, provenance chain is required which can be used to trace the data origin and history. To achieve that quality, our proposed solution was blockchain which requires tons of storage and data processing. With IoT device being small and having less storage and processing power, it is not feasible for them to process, store and validate blockchain to add new generate data. In addition to that, blockchain requires each node have consensus before adding new data to a chain. The solution to this challenge is also addressed in our proposed solution.

C. Indexing generated data provenance-aware infrastructure

Indexing a data is very important once data is generated. It helps to find and query the data once it is generated for further analysis. Also, with large datasets generated by large IoT network, it will be difficult to search for data without indexing. Property like, indexing the data, can be found in structured databases like SQL. But it does not provide a chain for data security and traceability. Also mentioned in paper, Security & privacy in IoT Data Provenance, "Data Citation can be done but it required efficient lookups in all dimensions to retrieve data" [11]. In short, this challenge summaries that we cannot have both functionalities (Indexing and Chain) in the system. We have to decide whether to use indexing or chaining data. Not using one of the feature will result in not fulfilling provenance-aware system requirement, since provenance-aware system requires both functionalities. The solution to this challenge is also addressed in our proposed solution.

D. Identification of entity who generates data (Ownership)

With the importance of security, quality, trustworthiness, and traceability of data, ownership of data is also important. Without ownership, life of generated data can be traced using chain, but a chain cannot identify who generated data which is required feature to have in provenance-aware system. Since secure chain functionality comes from blockchain technology, blockchain doesn't provide identity of generated data. But, the solution of this challenge is addressed in our proposed solution.

Now that, we know all the challenges towards building a provenance-aware system, In next section, we will discuss how did we overcome these challenges.

III. Our solution toward challenges

In this section, we will discuss how we managed to solve the challenges which we mentioned above. After listing out all the requirements of provenance-aware system and the problems related to it, we concluded that we needed something which has some properties of traditional database and some properties of blockchain. we were looking for blockchain database and we came across this new concept called BigchainDB. BigchainDB is the hybridization of blockchain technology and traditional database**.** Table I as shown from the paper "BigchainDB 2.0 The Blockchain Database", mentions the differences between blockchain technology, traditional database, and BigchainDB.

TABLE I

Difference between blockchain, distributed database, and BigchainDB

	Typical Blockchain	Typical Distributed Database	BigchainDB
Decentralization	✓		✓
Byzantine Fault Tolerance	✓		✓
Immutability	✓		✓
Owner-Controlled Assets	✓		✓
High Transaction Rate		✓	✓
Low Latency		✓	✓
Indexing & Querying of Structured Data		✓	✓

BigchainDB is a decentralized infrastructure in peer-to-peer network. In addition to that, BigchainDB is also Byzantine Fault tolerant which means the BigchainDB P2P network will stay up even 1/3rd of the BigchainDB nodes gets down. Furthermore, in BigchainDB creator/generator of the data has control over the data. That means, developer can write a query which user can use it which gives control of his data which he provides. With such control, node/user can specify that no one can update his generated records except him.

BigchainDB also database properties like high transaction rate, low latency and indexing and querying data structures. The developers of the BigchainDB has published their findings on paper. In below sub-section, we discussed how BigchainDB and our proposed solution (Proposed solution is discussed in detail in upcoming section) will solve the challenges we mentioned above. Figure 3 can be used as a reference for upcoming sub-sections.

A. Data Security

Data stored in BigchainDB are stored in MongoDB in JSON format. Furthermore, each JSON formatted data added to BigchainDB are stored using blockchain technology. This means data can be validated and data updated by intruder can be detected. In addition to that, data generated from IoT node is transported to BigchainDB network over HTTPS protocol. For IoT node to connect to BigchainDB network, requires a authentication token which authorizes that a node is a validate node. If node is not provided, connection cannot be made. Furthermore, for a new BigchainDB node to connect to BigchainDB network, requires X25519 key-pair of all the nodes present in a network and new BigchainDB node X25519 key-pair are shared with all the nodes present in a network. Authentication token and key pairs are used to authorize IoT nodes and BigchainDB nodes which indeed protects from attacks like 51% attack and secures data.

B. Storage and processing of data

If blockchain which consist of JSON format data is stored on each IoT devices, it will consume huge amount of storage and processing power which IoT devices don't have. We discussed this in challenging section. Moving the storage and processing services to cloud decentralized P2P infrastructure, solves this issue. Since each IoT nodes will connected to this cloud infrastructure over HTTPS protocol and secured using authentication token, the only responsibility the nodes will have is generating and transferring data and which indeed solves our problem of storage and processing.

C. Indexing generated data provenance-aware infrastructure

As we discussed earlier, indexing is important for provenance-aware system. BigchainDB provides two kinds of ids to each generated data. One is a public key of a node from which data is generated and another is transaction id which is provided by MongoDB database to a data which is inserted. Using this IDs, each data generated can be traced and also node which generated the data can be found. Furthermore, a node can also query the BigchainDB database to find particular data or a transaction. Figure 2 represents data which was inserted using our system model. If you can see the line number 14 in figure 2, shows the public key of a node from which the data is generated. The line number 38 in figure 2 shows the transaction id of the data. In addition to that, line number 5 shows the public key of the previous node who updated the data. In this case, it is the same as the one which created the node. BigchainDB doesn't allow to update the data but they still have update functionality which recreates the data in updated form which user is trying to update. They also have delete/burn functionality where data is not deleted but it is marked as no longer useful data which means no node can use it to update, or read.

D. Identification of entity who generates data (Ownership)

To identify and find the ownership of the data generator, public key shown in figure 2 can be used. Using this public key, parent node from where the data was generated first can be traced.

```
{
    "inputs": [
        {
            "owners_before": [
                "ATVug7dSQ3BBKN3naxteiGCsu8BfqYZ5rCSkmwAioY8h"
            ],
            "fulfills": null,
            "fulfillment": "pGSAIIyC-E7-r1kDDXLR0ySq-rePE4YMfmoW3AoqtEpt9Lz6gUATt_GCbIwb4V64u4eEKAl4ZUQYFopKCZ_eu9kbxEolT8Ie_jKdR2mWt3JM5F9YCweYk1FrZW0VnlSZaclMjiQI"
        }
    ],
    "outputs": [
        {
            "public_keys": [
                "ATVug7dSQ3BBKN3naxteiGCsu8BfqYZ5rCSkmwAioY8h"
            ],
            "condition": {
                "details": {
                    "type": "ed25519-sha-256",
                    "public_key": "ATVug7dSQ3BBKN3naxteiGCsu8BfqYZ5rCSkmwAioY8h"
                },
                "uri": "ni:///sha-256;ANbda3MvpdnAa4GQtR1T_QK_PvnDL1bwUdGeBj9-Oc?fpt=ed25519-sha-256&cost=131072"
            },
            "amount": "1"
        }
    ],
    "operation": "CREATE",
    "metadata": {
        "what": "My first BigchainDB transaction"
    },
    "asset": {
        "data": {
            "city": "Berlin, DE",
            "temperature": 22,
            "datetime": "Wed Dec 05 2018 12:07:40 GMT-0500 (EST)"
        }
    },
    "version": "2.0",
    "id": "5a3464c6efad8e8f0306eb4ee04f24495ba3bce5e3d7e098c246e10b5f877705"
}
```


Figure 2. Data stored in BigchainDB showing identification of data generator

In above sub-section, we saw how BigchainDB can be used to solve all the challenges we had. Since, solving this challenges guarantees the development of provenance-aware system, we used BigchainDB to develop our system model which is discussed in upcoming sections.

IV. System Model Description

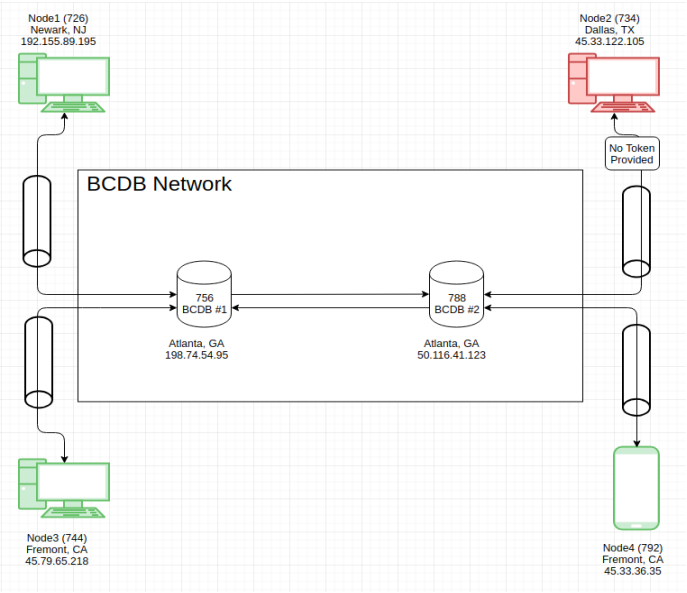

Figure 3. System Model

Our proposed system model consist of four angular nodes simulates as IoT nodes and a P2P network consist of two BigchainDB nodes. All the four angular nodes are connected with BigchainDB nodes on HTTPS protocol. These angular nodes and BigchainDB nodes are installed on ubuntu 18.04 virtual machines provided my linode cloud service provider. These nodes are located in different cities as described in figure 3. All angular nodes are provided with authentication token except Node 2 to communicate with BigchainDB network. The reason not to provide token to Node 2 is to simulate the condition if node without token tries to connect to the BCDB network. It is important to note that angular node 1 and 3 is connected with BCDB node 1 and angular node 4 is connected with BCDB node 2 and angular node 2 is trying to connect with BCDB node 2. Each node is further described in Table II.

TABLE II

Node Description

Node Name	Node Location	Node IP address	Node Type
BCDB #1	Atlanta, GA, USA	198.74.54.95	DB server
BCDB #2	Atlanta, GA, USA	50.116.41.123	DB server
Node 1	Newark, NJ, USA	192.155.89.195	Angular node
Node 2	Dallas, TX, USA	45.33.122.105	Angular node
Node 3	Fremont, CA, USA	45.79.65.218	Angular node
Node 4	Fremont, CA, USA	45.33.36.35	Angular node

Figure 4 shows the closer view of BCDB P2P network. It consists of two BCDB nodes. Each node consist of components listed below:

- NodeJS Server: NodeJS server is responsible for authenticating upcoming requests from IoT/angular nodes. It makes sure that request consist of authentication token and that token is valid. This token is generated using HS256 JWT (Json Web Token) signing algorithm and 256 character long secret key which resides on BCDB network. This secret key is used to validate tokens.
- BC Engine: BC Engine is responsible for maintaining and validating blockchain and storing it in MongoDB database
- MongoDB database: MongoDB database stores blockchain
- Tendermint: Tendermint is Byzantine Fault Tolerant. It makes sure that network stays up and running even 1/3rd of the BCDB nodes goes down. Furthermore, it is responsible for syncing data between two BCDB nodes. It is also responsible for authenticating the connection between two nodes before syncing data. It does that using X25519 key pairs. As described earlier, to establish a connection between two nodes, it requires to share the key pairs of each node with other node.

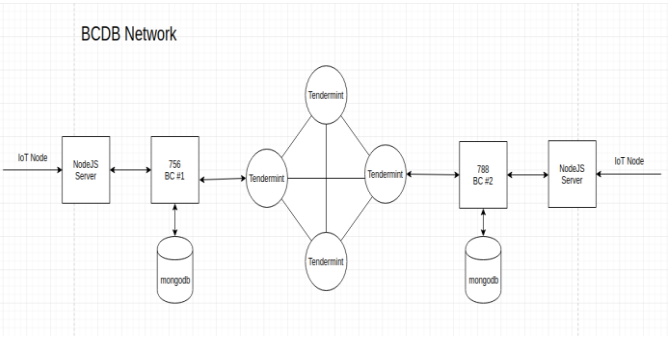

Figure 4. Closer view of BCDB network

In next section, we will discuss the life of a generated data in our proposed solution to understand the system model better.

V. LIFE OF A DATA IN OUR PROPOSED SYSTEM MODEL

In this section, we will discuss the life of a data from the point it is generated to the point it is been stored and synchronized with other BCDB nodes. In addition to the path of data on which it will go, we will also discuss how data is changed on different stages.

- The data will be generated from IoT/angular nodes. Before it is sent to BCDB network, an authentication token is added to the header of the HTTPS request.
- Once authentication token is added, the packet is sent to BCDB network over the internet. When packet reaches the BCDB network, it lands on Node JS server. This server will make sure that authentication token exist in HTTPS header and the token is valid using secret key stored on the server. If token is valid, request is passed through else a response is sent with a message that token is not valid.
- The authorized request is landed on Bigchain engine. Here, a public key of the node is added to the data. A previous public key is also added if data is recreated with update operation. In addition to that, other components shown in figure 2 are also added.
- After that, data is stored in Mongodb. Before storing, a unique id for the data is generated which is also stored with the data. Data stored here is in the form of the blockchain. So, chain validation also happens here.
- Once data is stored, Tendermint syncs the chain with available valid nodes on the network. The syncing process has more than just a simple sync. For detailed sync and consensus process, please refer to the Tendermint "Tendermint: Byzantine fault tolerance in the age of blockchains." mentioned in the references section.

If summarize, data is changed:

- On IoT/angular node: addition of token
- On Node JS server: elimination of token
- On BC Engine: addition of public keys and another component

This was a lifespan of a data in our proposed solution. In upcoming section, we will see the performance of our proposed solution by measuring the read and write speed of the data stored in BigchainDB.

VI. DATA READ AND WRITE MATRICES

In this section, we will see the performance of our proposed solution. To measure the performance of our solution, we used dataset of 1000 JSON formatted records which contained three values. Out of those three values, two values contained random 256 characters string. The configuration of each angular/IoT nodes are Ubuntu 18.04 with 2GB of RAM and 25GB of hard disk space. The configuration of each BigchainDB nodes are Ubuntu 18.04 with 4GB of RAM and 80GB of hard disk space. The Table III shows the angular/IoT node from where the data is sent, BCDB node where data is received, time taken to write the data, and time taken to read the data. Read and Write time shown in Table III were measured in milliseconds and converted in seconds. Also, it is important to that, this is a average time of three attempts.

TABLE III

Read and Write Time

Angular/IoT Node	**BCDB Node**	**Write Time**	**Read Time**
Node 1 (Newark, NJ)	BCDB #1 (Atlanta, GA)	196.062 seconds	8.341 seconds
Node 2 (Dallas, TX)	BCDB #2 (Atlanta, GA)	192.023 seconds	7.253 seconds
Node 3 (Fremont, NJ)	BCDB #1 (Atlanta, GA)	215.328 seconds	17.082 seconds
Node 4 (Fremont, NJ)	BCDB #2 (Atlanta, GA)	238.006 seconds	17.064 seconds

For this experiment, to measure read time, we wrote a loop in JavaScript which fetches each JSON records from the dataset and sends it to the database and waits till success response arrives. To measure write time, we wrote a loop in JavaScript which reads all the wrote data one at time. It is obvious that this numbers also depends on other factors like hard disk type, network speed, etc. But it gives overall view of how BigchainDB performance.

The distance of the angular/IoT node location from BCDB node is shown in the Table III above. Both, BCDB nodes, are located in Atlanta, GA data center. The closest angular/IoT node is Dallas, TX which took least amount of time to read and write than others. The other closest one is Newark, NJ. And furthest one is Fremont, CA which took longer time than any other nodes. The future solution would be here to make bigger BCDB network and let angular/IoT nodes dynamically select the BCDB nodes which is closest to it. This will reduce the performance issue caused due to distance. Up next, we will finish this paper by looking at what we conclude and future works.

VII. CONCLUSION AND FUTURE WORK

In this paper, we presented a secure provenance-aware system which possess properties like:

- Quality and trustworthiness of the data
- Security of the data
- Traceability of generated data
- Ownership of the data

To achieve such quality, our basic idea was to create provenance chain from which we moved our research focus

on blockchain. Blockchain has all the properties that provenance-aware system requires except, ownership of the data. All the transaction happens in blockchain, are anonymous. This made us investigate a hybridized database called BigchainDB. This hybrid database has properties of blockchain as well as of typical database. Using such technologies, we developed a system which is provenance-aware.

The research performed during this work revealed further work related to proposed solution. We would like to grow the BCDB network and make IoT devices dynamically choose the closet BCDB node for transaction.

REFERENCES

[1] Polyzos, George C., and Nikos Fotiou. "Blockchain-assisted information distribution for the internet of things." Information Reuse and Integration (IRI), 2017 IEEE International Conference on. IEEE, 2017.

[2] Panarello, Alfonso, et al. "Blockchain and IoT Integration: A Systematic Survey." Sensors 18.8 (2018): 2575.

[3] Outchakoucht, Aissam, ES-Samaali Hamza, and Jean Philippe Leory. "Dynamic access control policy based on blockchain and machine learning for the internet of things." International Journal of Advanced Computer Science and Applications (IJACSA) 8.7 (2017): 417-424.

[4] Lombardi, Federico, et al. "A blockchain-based infrastructure for reliable and cost-effective IoT-aided smart grids." (2018).

[5] Baracaldo, Nathalie, et al. "Securing data provenance in internet of things (IoT) systems." International Conference on Service-Oriented Computing. Springer, Cham, 2016.

[6] McConaghy, Trent, et al. "BigchainDB: a scalable blockchain database." white paper, BigChainDB (2016).

[7] Pon, Bruce. "Blockchain will usher in the era of decentralised computing." LSE Business Review (2016).

[8] Kshetri, Nir. "Can blockchain strengthen the internet of things?." IT Professional 19.4 (2017): 68-72.

[9] Hong, Namsu, et al. "A Study on a JWT-Based User Authentication and API Assessment Scheme Using IMEI in a Smart Home Environment." Sustainability 9.7 (2017): 1099.

[10] Dorri, Ali, Salil S. Kanhere, and Raja Jurdak. "Towards an optimized blockchain for IoT." Proceedings of the Second International Conference on Internet-of-Things Design and Implementation. ACM, 2017.

[11] T.K, Adarsh & Rethnaraj, Jebakumar. (2018). Security & privacy in IoT Data Provenance. International Journal of Engineering and Technology. 10. 843-847. 10.21817/ijet/2018/v10i3/181003085.

[12] Bradbury, Danny. "The problem with Bitcoin." Computer Fraud & Security 2013.11 (2013): 5-8.

[13] Kwon, Jae. "Tendermint: Consensus without mining." Draft v. 0.6, fall (2014).

[14] Buchman, Ethan. Tendermint: Byzantine fault tolerance in the age of blockchains. Diss. 2016.