International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-8 Issue-10, August 2019

Integration of Blockchain Technology in IoT for High Level Security

Siva Naga Lakshmi Pavani Kallam, BVNR Siva Kumar

Abstract: ***IoT (Internet of Things) made headway from Machine to Machine communication without human intrusion for number of machines to connect with the aid of network. There is esteem; by 2020 there will be 26 times more connected things than people. Hence, the concern of security rises along with the high installments. The BlockChain Technology takes place of all central entities, which is peer to peer communication with the distributed network. In this paper, two Arduino boards as nodes and a Raspberry Pi as server are to be configured to connect to the Wi-Fi using ESP8266(node mc). To make data transmission from the two nodes to server, integration of temperature and humidity sensor in one node and RFID (Radio Frequency Identification) reader in other node is to be done. Data should be in the form of blocks and integration of data is in the form of a chain, forming it a Blockchain. All the blocks are linked in the chain manner of which the current hash of the previous block must match with the previous hash of the next block. Then only the blocks of data are secured. While receiving data every time from nodes to server, the previous hash is to be checked such that the arrival of the information is being verified to know if it's really genuine. If the cryptographic hash does not match then data manipulation is happened. So, in this paper, we will see, along with how practically the security is highly offered by the blockchain technology and how can we easily identify if the data has been tampered along the way it reaches to us. Henceforth, we will found a way of application to secure our IoT data without any regrets in this paper.***

Index Terms**:** ***Blockchain Technology, Cryptographic hash, IoT, peer to peer communication***

I. INTRODUCTION

IoT is the concept which has the potential of not only to impact how we live but also how we work. IoT allows for virtually endless opportunities and connections to take place but also opens lot doors to big challenges. Security is a big issue that major times bought up. With the billions of devices connected together, what makes sure that the information stays secure? How can we make sure that the information we traded for sugar is not salt or something else

With the centralized access control systems, users are able to access all their work programs and assets through a single set of login credentials. A hacker only has to breach one set of credentials to access everything that verified users can access. Also at every time data is being transmitted it is a long journey as include more delays because of its centralized authority. An alternative to the centralized access management system is to have identities distributed across many different systems. In the distributed access systems, management of individual systems must known. Henceforth, new ways of approaching the problem are needed.

In this paper, we present an integration of blockchain technology which is having the advantages of data transparency and auditability, distributive information, decentralized consensus, secure and decentralised network in IOT of which the major challenge is security.

II. OVERVIEW OF BLOCKCHAIN TECHNOLOGY

A. *BLOCKCHAIN*

The popular and successful application of Blockchain came into picture in the year of 2009 by Satoshi Nakamoto in the name of Bitcoin which is the cryptocurrency or electronic cash implemented using peer–to-peer communication. Not only the blockchain transfer and store money, but it can also replace all processes as the charges for transaction is small fee. Blockchains will change the way stock exchanges work, loans are bundled, and insurances contracted. Later, using the same structure, other forms of cryptocurrencies evolved. Using this technology, along with the cryptocurrencies, various different applications like smart contracts, the sharing economy, crowd funding, supply chain auditing, prediction markets, neighborhood microgrids, AML, KYC, Data management have entered the scene.

The predominant thing in the digital transfer of money is to be safe and secure so that there should not be any double spending problem. Not only about the double spending problem but the transaction should be secured and transparent in such a way that it should be accepted by everyone which means immutable. All these type of requirements are satisfied in the blockchain technology.

B. *BLOCKCHAIN TECHNOLOGY*

The blockchain is a system which evident list of transactions with the timestamp, data, previous hash and nonce made in bitcoin or another cryptocurrency.

Information held on a blockchain exists as a shared-and continually reconciled-database. This is a way of using the network that has obvious benefits.

Manuscript published on 30 August 2019.
*****Correspondence Author(s)
Siva Naga Lakshmi Pavani Kallam, Electronics and Communication Engineering, Lakireddy Bali Reddy College of Engineering, Mylavaram, INDIA.
BVNR Siva Kumar, Associate Professor, Electronics and Communication Engineering, Lakireddy Bali Reddy College of Engineering, Mylavaram, INDIA.

The blockchain database isn't stored in any single location, meaning the records it keeps are truly public and easily verifiable. No centralized version of this information exists for a hacker to corrupt. Hosted by millions of computers simultaneously, its data is accessible to anyone on the internet. The chain of blocks in the blockchain is formed by containing the previous hash in the current block. The next block should only be added to the chain if the miners verify the current block contains the previous hash which is generated by using the transaction data, time stamp, nonce and target. The first block in a blockchain is called as *genesis block*. The *genesis block* is almost always hardcoded into the software. The special case in the chain of blocks is *genesis block* that does not cite to previous block. In every blockchain, there is only one way to the *genesis block*. There are special nodes called miners to solve the computational puzzles to create and confirm blocks in the chain. The transactions before they added to a block are sustained in the area which is called as mempool. The miners will pick up these transactions in the batch wise and add them to the new block which is ready to do mining. Proof-Of-Work is the original consensus algorithm in a Blockchain network. With POW, miners compete with each other to get them rewarded if they complete the puzzle in the first place. If someone tries to hack or falsify a block, they need to hack the entire chain which is practically impossible. The new entries of blocks in the chain must be confirmed by the thousands and even millions of miners. Most probably, for every transaction to be confirmed, it requires minimum of eight blocks of miners acceptance of legal correctness of transaction is needed. The three paramount reasons which gain blockchain more popularity are:

a. DECENTRALIZATION

In the decentralized network, the information is not stored by one single entity. In fact, everyone in the network owns the information. In this system, we can directly interact with our peers without the need of third party. Even though there are decentralized entities before bitcoin, lots of points to be considered to make it successful including computational power which needs more to handle processing steps. And even scalability is also the quest of handling the nodes in the centralized network which is very much limited compared to the decentralized network because of all applications and processing power are housed in a single server. Coming to the blockchain technology, the difficulty of block makes the miners to take 10 minutes to add the new block. Hence, it takes huge computational power. So, we well known aware of a disadvantage in the decentralization that if the access is distributed each and every node will be masters. Hence, there will be some differences among them while the network needs to make critical decision. But, in the blockchain technology, if one miner got rewarded for solving the puzzle and add a new block to the chain series, other miners will check for confirmation other than creating differences among the nodes.

Hence, after adding a new block in the chain, there are confirmation blocks which show us the Proof-Of-Work. So, it is easier to secure the data in the IOT case by using trustless blockchain technology.

b. TRANSPERENCY

The magnitude of privacy that a blockchain can provide is the one of most winning aspects. In the blockchain technology the data which we transmitted is hidden via complex cryptography. The real information we are sharing is pseudo anonymous. Blockchain is one system created to untangle the probe of how to trust the network when all users are anonymous. Hence blockchain gives large businesses a platform to act with genuine integrity towards their community and customers. However, blockchain has the potential to add transparency not only to the financial aspect of business. Blockchain is the technology which provides this intensity of privacy which has never done in former.

Not to limit the spectrum of revolutionary benefits provided by the blockchain technology which is the model that does not require trust to safely interact and transact, we are integrating it in the IOT to make the statement of privacy dead false.

c. IMMUTABILITY

Blockchain's public ledger is not a book with numbered pages. There is no sequence of pages to verify the ledger out of which we are expecting to be in order. So, practically it is possible that tearing a page in middle and replacing it with the other can be a major threat to trust this technology. Hence, it is not a type of sequence of blocks but a series of blocks of which we can run checks on it get the same hashes by using the same signature of user's transaction. Hashing is the main requisite and vital function in the blockchain technology. The most appealing property of hashing algorithm is reversible property. We can perceive the same hash output if we recall the function with the same input.

The blockchain consists of hash pointer which points to its previous block, hence creating the chain. A hash pointer is similar to a pointer, but instead of just containing the address of the previous block it also contains the hash of the data inside the previous block. This one small tweak is what makes blockchains so amazingly reliable and trailblazing.

In the next section we will see the hardware requirements and how the integration of blockchain is implemented.

III. HARDWARE IMPLEMENTATION OF INTEGRATION OF BLOCKCHAIN TECHNOLOGY IN IOT

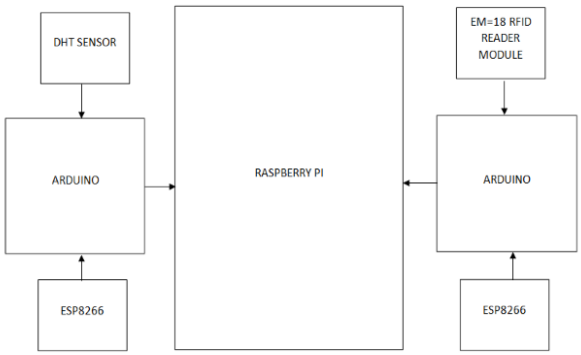

Fig.1. Block Diagram

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

At first, to implement blockchain technology, we need data to communicate form nodes to server. Let the two arduino boards be two nodes and a raspberry pi as the server. To get data from these nodes, we integrated temperature and humidity sensor in one arduino board and RFID (Radio Frequency Identification) reader in another one. Whenever the sensor senses the data regarding temperature and humidity, the data will be collected at the server. The data which we are receiving at the server side is to be identified whether it is the actual sensed data from those two arduino boards or it gets tampered in the mid-way. Let me first describe the arduino boards integrated with the Temperature and humidity sensor as node-1 and RFID reader as node-2.

The power supply is generated to the nodes by taking AC230v and connects this cable to step down centre tapped transformer. So the centre value is 0 and both sides of those three wires are same and here it is 12-0-12v with 750ma. The output and input of this transformer are in AC form. So, to convert it to DC (Direct Current), a full wave rectifier is designed by using two diodes and a capacitor. The difference between RMS and Peak is square root of 2. Here while converting 12V AC (Alternating Current) to DC, it will multiples with the square root of 2 and hence results in approximately DC power supply 17V (16.9V). But, our sensors, NodeMCU and RFID reader will operate in 5v. Hence, we connected a regulator 7805. The centre pin of the regulator is connected to ground and the first pin of the regulator is connected to the capacitor positive as the input and the other pin is at the output which is 5V. And using this power supply, wherever the VCC is required connections are made accordingly to 5V and wherever the circuit has to be grounded, make it to 0V. And the capacitors are labelled at both the input and output of the regulator. Hence the power supply is designed. The power supply design is same to both the nodes. We will start with node-1 description.

Apart from the power supply, the sensor used in the node-1 is DHT sensor (Digital Temperature and Humidity Sensor). The DHT sensors are made of two parts, a capacitive humidity sensor and a thermistor. There is also a very basic chip of ADC (Analog to Digital Converter) inside that does some analog to digital conversion and spits out a digital signal with the temperature and humidity. The DHT sensor body size is 15.5mm*12mm*5.5mm and operates with 5V power. The sampling rate is no more than 1Hz i.e. once every second. There are 3 pins with 0.1" spacing. The digital signal obtained from the analog to digital converter is fairly easy to read using any microcontroller. So, here we used Node MCU ESP8622 which is the microcontroller having integrated support for Wi-Fi network. NodeMCU is the perfect apt for this project because we have to connect our nodes and server to the same Wi-Fi module for IoT account and this microcontroller has wireless fidelity inbuilt. This NodeMCU operates at 5V which is Breadboard-Friendly. The pins of the DHT sensor are connected in such a way that one is to VCC supply and another to Ground. The centre pin which digitally transmits the data of temperature and humidity is connected to the Node MCU. We can literally connect the centre pin of DHT sensor to any GPIO (General Purpose Input/output) pin of the NodeMCU. The respective connected pin is programmed accordingly in the code for reception of data. Hence the node-1 is designed to get continuous messages to the server whenever it sensed the geographic temperature and humidity.

Fig.2.Arduino Board with DTH sensor and NodeMCU

In the node-2 design, the same power supply design is used as described previously. Along the NodeMCU, here we use EM-18 RFID reader module. This is a module which reads the ID (Identity) information stored in RFID Tags. This ID information is unique for every Tag which cannot be copied. The reader module comes with an on-chip antenna and can be powered up with a 5V power supply. We power-up the module and connect the transmit pin of the module to receive pin of the Node MCU. I took three RFID tags having the ID's as 1E002CB326A7, 1E002C9CEA44, 1E0034CAC323. The operating voltage of EM-18 is +4.5V to +5.5V. It can operate in low power and the frequency which this module operating is 125Hz. The distance which we can show the RFID tags are up to 10cm from the module. Henceforth, using the RFID Tags, we can continuously send the messages to the server by showing tags to the EM-18 RFID reader module.

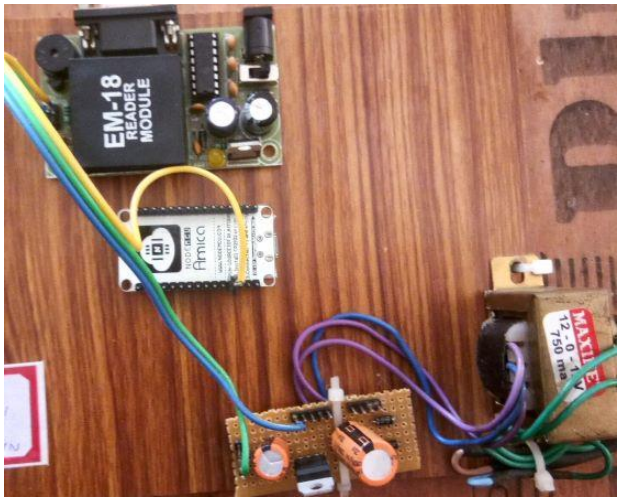

Fig.3.Arduino Board with RFID module and NodeMCU

In this way the two nodes are designed and connected to the same Wi-Fi module. Now, we have to set the server as Raspberry Pi using the same Wi-Fi (Wireless Fidelity) module. In this project Raspberry Pi 3 model B is used. The Raspberry Pi 3 Model B is having Broadcom BCM2837 SoC features a quad-core 64-bit ARM (Advance RISC Machine) Cortex A53 clocked at 1.2 GHz.

Hence, the reason makes Pi 3 runs one-second percentage faster than other previous versions, the graphic capabilities are provided by a VideoCore IV@ 400 MHz graphics processor and have 1GB of LPDDR2-900 (Low-Power Double-Data-Rate) SDRAM (Synchronous Dynamic Random-Access Memory), it's having 802.11 bgn Wireless LAN (Local Area Network) and Bluetooth 4.1 Classic. Ports such as HDMI (High-Definition Multimedia Interface), 3.5 mm analogue audio-video jack, 4*USB 2.0, 10/100 Base T Ethernet socket to quickly connect the Raspberry Pi to the internet, Camera Serial Interface (CSI), Display Serial Interface (DSI), Micro SD are integrated in this SOC. We do not require an external antenna as the radios connected to this chip antenna are soldered directly to the board. The antenna here in the Pi 3 model is more capable of collecting the Wireless LAN and Bluetooth signals even through walls.

Fig.4.Raspberry Pi 3 Model B

Using the user name and password, Raspberry Pi is set to connect to the same Wi-Fi module I have been using for both the nodes by scanning the IP address and hence connecting and programming it in the VNC (Virtual Network Computing) Viewer in our personal computer.

So, till now we designed our nodes and server and made them to connect to the same Wi-Fi module.

Before going to the software section, first we have to set variables in the server using a MQTT (Message Queuing Telemetry Transport) platform.

We are using UBIDOTS as our MQTT server to create the variables using our implemented hardware. UBIDOTS is the platform which turns the data sensed using our sensors into data analytics of Internet Of Things. So, MQTT server is used here to get our data from the two nodes as above mentioned. So, practically the data has been empowered to visualisation using UBIDOTS. As discussed in the following chapters, it will visualize the data we sensed using our nodes to our best interpretation. Henceforth, we utilized this platform to make our real-time application to be visualized impeccably. Five variables are to be created by signing in with our user id and password. The variables are created by using the hardware nodes which are designed. The variables are named *as temperature, humidity, RFID, tamper and status*. The variable named *temperature* will show us the DHT sensor sensed data which is from node-1 i.e. Temperature: 36. And the *humidity* variable shows us the sensed information by the DHT sensor from node-1 i.e. Humidity: 61. Whenever RFID tag has been shown to the EM-18 RFID module reader, that particular unique ID of the tag which is used among the three will be displayed in the *RFID* variable in the form of numbers labelled as one, two and three. It means that if the unique ID is 1E002C9CEA44, then it will show as 1 and if the unique ID is 1E002CB326A7, it will show as 2 in the created variable RFID. The *tamper* variable is to be *set* if we want to manipulate the data as per IoT command. *Tamper* variable has been used to make the blockchain identify if the data is manipulated. The *status* variable is used to let us know if the data is being tampered or not. If the data is identified by the blockchain i.e. if the current hash of the previous block does not matches with the previous block hash when recheck is done, then it is proved that the data is tampered or manipulated. Hence the *status* is shown as red which is the symbol of something miscellaneous happened and we have to alert. If the current hash of the previous block matches with the previous block hash, then the data is not manipulated or tampered. Hence in the variable *status*, we will see the *green* intimation which represents that the data is safe and secure. So, in this manner we can easily identify if the data we send is accurately secured or not.

IV. CODING

In this blockchain project, python language is used to implement blockchain technology. Here are the reasons why we choose python as an excellent language for a blockchain project.

i. It's advanced and easy to learn

Python has been around for a while now, and its position on the tech scene is growing stronger. Since it's supported by a large and passionate community of developers, Python has significantly evolved as a language and is now at an advanced stage, which guaranteed stability and reliability. Python has a gentle learning curve, making it easier for developers to master it within a reasonable time-frame, and even allows for less experienced developers to contribute to projects immediately.

ii. Simple and minimalistic

Simplicity and minimalism are at the core of Python's philosophy. Its simplicity derives from many different features-for example, in python white spaces signify code blocks, and developers don't need to worry about adding curly brackets or keywords. We can use python to code a blockchain without having to write a lot of code.

iii. It can be run compiled or uncompiled

Contrary to C++, python is a scripted language that doesn't compilation to become understandable to machines, which makes blockchain developers lives more comfortable. Imagine running an application and noticing a bug. If we use a compiled language, to fix it, we will have to stop the application, return to the source code, fix the bug, recompile the code, and restart our application. In python we won't have to recompile code which is massive advantage in building blockchains. Python offers the option of pre-compiling the code along with many other techniques that speed it up, giving developers working in blockchain a choice.

Retrieval Number: J88150881019/19©BEIESP
DOI: 10.35940/ijitee.J8815.0881019
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

iv. Python for Blockchain

Blockchain has specific requirements when it comes to code and language. When choosing a programming language for a blockchain project, we must make sure that the language is secure, performant, and scalable. We need an advanced and reliable language to make our blockcahin as safe as possible- and python is helping us with that.

Since anyone in the same IOT account can add to our blockchain, our network and code should be able to deal with a growing query list. Python has that covered well. Finally, python allows us to create a simple blockchain in less than 50-100 lines of code. First, we need to define what our block will look like; each block in the blockchain is stored with a timestamp and an index. Blockchain integrity is a key, so it should be ensured with a cryptographic hash of the block's index, timestamp, data and a hash of the previous block's hash (or the genesis block at the start). Python is strongly recommended for blockchain if have to address an Internet Of Things use case for the reasons which we have discussed. In python, we can easily perform many tasks with a single command. It makes the work of building blocks with the relevant information and linking them together a much easier one to do. Python is clean and has a huge collection of libraries already available, to name just a few of the reasons why we used python.

Coming to the written code in this project, first we have to scan the IP address of the IOT network of which both the nodes, UBIDOTS and the Raspberry Pi are connected to the same account. Then, we have to login to the Raspberry Pi using our user name and password using VNC Viewer. Then we declare our variables by opening terminal and giving commands like *create directory* and *sudo idle* to write the code in the administer mode, which was created using the MQTT server. Whenever a message arrives to the server, it will check the current hash in the previous block with the previous hash in the current block. If it does not matches, then data is manipulated messaged will be displayed in the screen. To prove it practically possible, we have to set the *tamper* variable to one so that last right index in the block is being changed to some other data other than original data based on the IOT command. So, when recheck happens it will be displayed as data is tampered.

V. RESULTS

Hence forth, based on the above discussion, creation of practical results is to be done to show how this idea becomes valid. So, we first connected all the devices we mentioned such as two nodes, one Raspberry Pi and UBIDOTS to the same network using a Wi-Fi module and run the code which is written in the python 2.7.9 Shell in Raspberry Pi to view the results. The MQTT works on publish-subscribe method to exchange data. The two nodes will be publishing the data when they continuously sense and send the messages regarding temperature, humidity and RFID unique number. Raspberry Pi will subscribe the data which will published by these nodes. The first block is an exception case as it is the special *genesis block*.

As we can see that in figure4, at first we will be reading the message from node-1 as it senses the information for every two seconds. Then the previous hash code for blockchain will be displayed. So, again the recheck has been done to verify whether the previous hash code displayed in this current block matches with the current hash code in the previous block. If both hash codes match, then data manipulation is not happened message will appear.

We can even evident ourselves from this how the blockchain technology is highly secured, that even before subscribing the data that published, the blockchain fore mostly checks the hashes and verifies that any tampering has been done or not. We can even write the programming in such a way that after the hashes verified; we can exit out of the loop without proceeding further to create blocks. But our theme in this project is to make the identification of each block has been tampered or not. So, the blocks are keep on adding even though we identify the manipulation is happened. So that we can get the status of the manipulation every time it does in the UBIDOTS *Status* Variable. We are making it possible to check and identify every block to know that it has been tampered or not.

After that, coming back to the result displayed in the screen, Raspberry Pi will subscribe the data published by node-1 i.e. Temperature and Humidity values. Then, current hash has been generated using the data appeared in this block. Again, by using this current hash as the previous hash code for the next block, verification of these two and again a new hash code has been generated as the current hash code in this block. In this figure we can even see that the message form node-2 also subscribed, as we show a RFID Card with the unique number as 1E002CB326A7. As we mentioned earlier, we set each RFID unique number to the integer value as one, two and three in the UBIDOTS platform. Whenever the Raspberry Pi subscribed the data, then the indications of temperature, humidity, status, RFID will be displayed in the dashboard in UBIDOTS. And even the log of data we interpret has also been stored so that we can check how many manipulations have been done entirely while we subscribing the data.

So, this is the evident of screen and intake of messages when *tamper* variable has not been *set*.

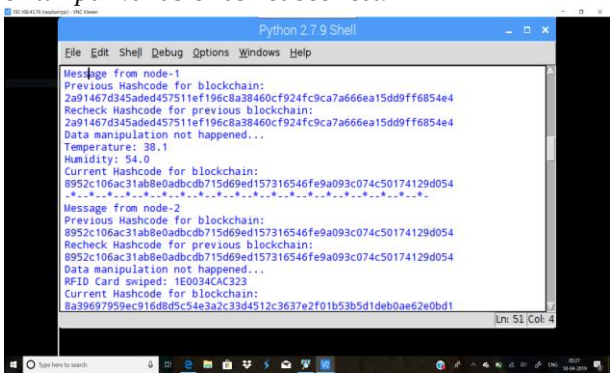

Fig.5.Identification of data which is not manipulated

The corresponding variables in the UBIDOTS with the *status* variable as green in colour represents that data is safe and hence not manipulated. We can check in *RFID* variable that last used card is 2 which is having the unique ID number as 1E0034CA323.

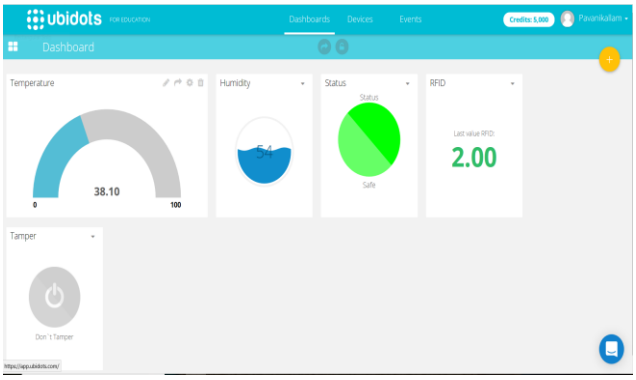

Fig.6.Variablesrepresented in UBIDOTS

If, *tamper* variable is *set* in the UBIDOTS server, the data has been set in such a way that the right index of the chain has been manipulated.

As the chain of blocks series is a never ending process when the data has been continuously publishing and subscribing, every message Raspberry Pi is getting is considered as the right index of the block and hence every block data has been manipulated as per the IOT Command. Here, an IOT Command is written to corrupt the data ourselves to practically prove whether data which is manipulated has been identified or not.

The figure below is the evident of identification of the data has been tampered.

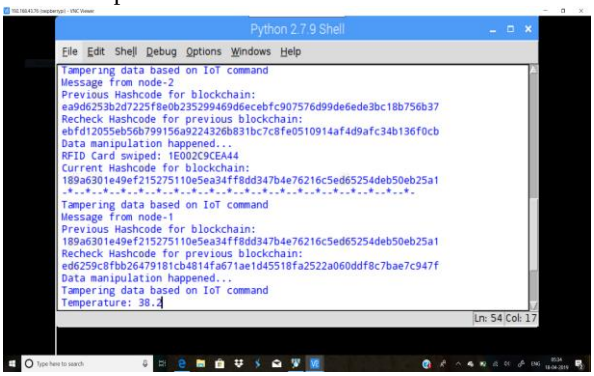

Fig.7.Identification of tampered data

The UBIDOTS representation of variables has also been shown in Fig7 which makes us known easily by seeing the *status* Variable that the data has been tampered because of *tamper* variable is on according to the IOT Command.

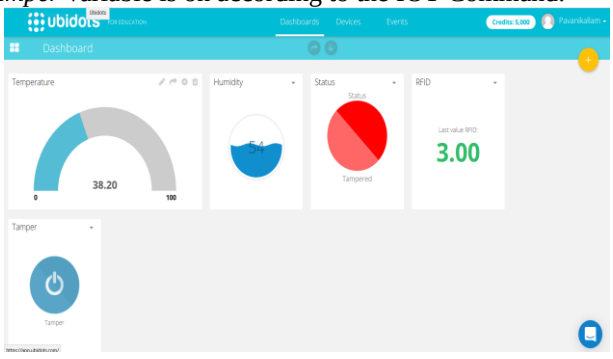

Fig.8.Variables representation in UBIDOTS when tampered

We can even check each variable's status at any time we need either in the graph form or directly integer value in table form in UBIDOTS.

Let's see in the RFID three Tags usage in the graphical form.

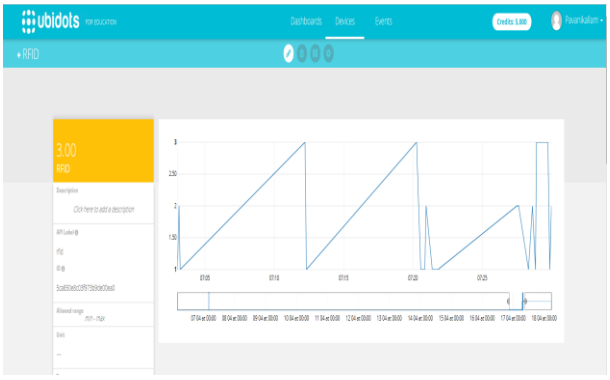

Fig.9.Graphical representation of RFID Tags usage in UBIDOTS

We can even check the *status* Variable to let us know how many times the variable shows red indication which enlightens us that the information is manipulated.

Date	Value
2019-04-18 05:33:43 -04:00	3.00
2019-04-18 05:33:34 -04:00	1.00
2019-04-18 05:25:35 -04:00	2.00
2019-04-18 05:25:18 -04:00	2.00
2019-04-17 07:29:58 -04:00	2.00
2019-04-17 07:29:51 -04:00	1.00
2019-04-17 07:29:43 -04:00	3.00
2019-04-17 07:28:54 -04:00	3.00

Fig.10.RFID Tags usage in Tabular Form

The same case belongs to all the Variables we created in the UBIDOTS Platform.

Hence, this section proves that the data we are receiving is secured or not in the way it reaches to us.

VI. CONCLUSION

If we identify that the information we are seeing in the screen is not exactly it is and is being manipulated along the way it reaches, we would not even perceive and there will be absolutely nothing to worry. This idea makes this project successful. The integration of Blockchain Technology in IOT for high level of security is in such a way that it is easy to verify our data and complex to establish the blocks in the chain form. Henceforth, it is very secure to exchange without even third party help. And python language helps us to make the code in even less than 120 lines to implement this complex structure in a well organized manner.

VII. FUTURE SCOPE

This paper is a proof to identify the tampered data in IoT using blockchain technology. This scope can be extended in such a way that the data of blocks cannot even be integrated if the block is manipulated. Also, there is a scope for multiple dimensions to be executed using blockchain technology in IoT for which the major constraints of security is a perfect match for the *consensus* in this technology.

REFERENCES

1 X. Sun and N. Ansari, "Dynamic resource caching in the Ioapplication layer for smart cities," IEEE Internet Things J., to be published.
2 [Online]. Available: http://bitcoin.org/bitcoin.pdf
3 [Online]. Available: https://link.springer.com/content/pdf/10.1007/978-0-387-35568-9_18.pdf
4 [Online]. Available: https://eprint.iacr.org/2013/881.pdf
5 The Autonomous Decentralized Peer-to-Peer Telemetry (ADEPT) System, IBM, Armonk, NY, USA, 2015.
6 M. Conoscenti, A. Vetrà, and J. C. D. Martin, "Blockchain for the Internet of Things: A systematic literature review," in Proc. 13th Int.
7 A. Ouaddah, H. Mousannif, A. A. Elkalam, and A. A. Ouahman,"Access control in the Internet of Things: Big challenges and new opportunities," Comput. Netw., vol. 112, pp. 237–262, Jan. 2017, doi: 10.1016/j.comnet.2016.11.007.
8 A. Ouaddah, A. A. Elkalam, and A. A. Ouahman, "Fairaccess: A new blockchain-based access control framework for the Internet of Things," Security Commun. Netw., vol. 9, no. 18, pp. 5943–5964, 2016.
9 [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X13000241
10 L. Atzori, A. Iera, and G. Morabito, "The Internet of Things:survey,"Comput. Netw., vol. 54, no. 15, pp. 2787–2805, Oct. 2010
11 O. Arias, J. Wurm, K. Hoang, and Y. Jin, "Privacy and security in Internet of Things and wearable devices," IEEE Trans. Multi Scale Comput. Syst., vol. 1, no. 2, pp. 99–109, Apr./Jun. 2015.

AUTHORS PROFILE

Siva Naga Lakshmi Pavani Kallam born on 10th August 1996 and is graduated from NRI Institute of Technology Affiliated to JNTUK in the stream of ECE. And pursuing Master's in the stream of VLSI &Embedded Systems in Lakireddy Bali Reddy College of Engineering, Mylavaram, INDIA. Area of Interest is Embedded Systems and VLSI.

BVNR Siva Kumar born on 26th March 1965. Post Graduated from JNTUEC, KAKINADAA in the year of 2004 and presently pursuing PhD in Amity School of Engg. &Tech. And working as Associate Professor in Lakireddy Bali Reddy College of Engineering, Mylavaram, INDIA. Area of Research is Medical Robotics.