Security and Trust in IoT Data Streams using Tangle Distributed Ledger and Node-RED Technology

Chia Yu Lee
School of Electronic Engineering and Computer Science
Queen Mary University of London
London, United Kingdom
c.lee@hss19.qmul.ac.uk

Kamyar Mehran
School of Electronic Engineering and Computer Science
Queen Mary University of London
London, United Kingdom
k.mehran@qmul.ac.uk

***Abstract*— Data streams are an important feature of IoT sensor networks. Data provides information regarding the state of the system being monitored. For example, in a power system, IoT-enabled smart meters provide real-time knowledge regarding the state of power distribution, which can improve the operation of the power system in every aspect. This research project is about feasibility analysis and implementation of Tangle distributed ledger and Node-RED technologies for securing the communication and bringing trust to the collaborating parties. The implementation of the Tangle distributed ledger will be based on the available open-source software developed for IOTA cryptocurrency. In the end, it is expected to have a prototype of data transaction platform that provides the requirements of security and trust in IoT data streams, which can be used for applications such as power system and smart sensor monitoring.**

Keywords—IoT secure, Tangle distributed ledger, Node-RED, IOTA, secure data transactions.

I. Introduction

The booming technology has rapidly changed the way people live. Viewing from the point of our daily life, we are surrounded by massive Information and Communication Technology (ICT) devices (Scheepers & Middleton, 2013), such as PCs, mobile phones, and smart wearable devices. In more specifically, we are now stepping into a more intelligent and convenient world. Comparing to 30 years ago, people can now easily make a video call through their smartphone instead of making a costly long-distance call. However, this trend is not limited to the connection between people, but in many different aspects. For instance, the way in data streams transaction of the Internet of Things (IoT). Using small devices and sensors to transmit the data is not a "News" to most people. Instead, people are more concerned about the weakest part of the IoT data transactions, which is the "Security".

Therefore, an innovative technology called "IOTA" was introduced by David Sønstebø, Sergey Ivancheglo, Dominik Schiener, and Dr. Serguei Popov to the world in 2015 (Satoshi Watch, 2017). IOTA is a revolutionary data transaction structure specially designed for solving the low computational resource problem that faces by most of the IoT-based equipment. By using the open-source distributed ledger technology, IOTA allows connected devices to transfer data between each other without paying any extra fee (IOTA, 2020). Different from the prestigious technology "Blockchain", which is used by Bitcoin, IOTA adopts a technology called "Tangle", which is constructed based on Directed Acyclic Graph (DAG) instead of Block structure, and there are two advantages in using DAG. First, as figure 1 shows, the DAG structure can provide parallelized validation, which means that when Tangle grows with more transactions, IOTA will become faster and more secure. For another benefit is that, in Blockchain, parties need to compete against each other in order to earn the rewards and add the next block, and these competitors are also known as "Miners". However, in DAG, every new transaction will be auto-connected and validated by two previous transactions. This can make sure that every validation will be operated simultaneously, and it can also create a decentralized exchange network.

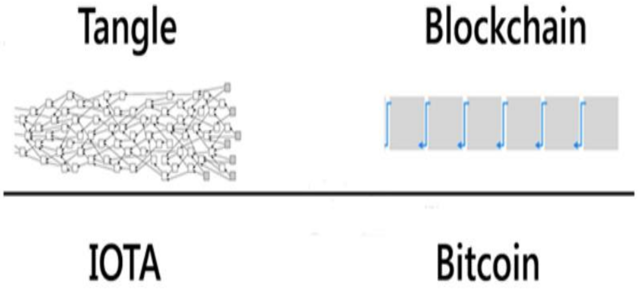


Figure 1. Data transaction structure of Tangle and Blockchain (IOTA Support, 2020).

On the other hand, another technology called "Node-RED" will also be adopted in this project. Node-RED is a flow-based programming tool, originally developed by IBM's Emerging Technology Services team and now a part of the JS Foundation (Node-RED, 2020). By using a web-based interface, Node-RED can easily transfer physical devices connected structure into a visualized workflow. It not only provides a simplified user interface but also supports various types of programming language. The concept of Node-RED is to create a friendly communication environment for both developers and users, and its function which supports the implementation of IoT devices has also made it the best choice in constructing a data transaction platform. In this research project, we aim to deliver a platform designing to realize the security and trust in IoT data streams transaction. For the following sections of this report, we will draw the building process and difficulties in detail. Elaborate requirements from different angles in designing a feasible platform architecture based on technical solutions.

II. Project overview

A. *Secure data transaction platform*

The initial idea for the secure data transaction platform is to create a platform that can be used to transmit real-time data or messages generated by IoT-enabled smart meters or small sensors. This platform should be constructed to meet three basic requirements, receiving data, transmitting data, and displaying data. In a traditional electronic engineering area, getting data from devices and showing them on the monitor is not a big problem. However, combining with Tangle distributed ledger technology will escalate this technique to the next level, and also difficult to realize. Fortunately, the embedded functions provided by Node-RED perfectly fit-in and solve this dilemma.

Overall, the platform structure will be based on the combination of Tangle distributed ledger and Node-RED technology. Before everything is started, a private Tangle network should be constructed on the localhost. As someone might critics that the original intention of using IOTA is to exchange cryptocurrency through the Internet like Bitcoin. But the truth is that Tangle distributed ledger technique can be used not only in transmitting digital currency but also in delivering more reliable data streams. Another reason for creating a local private Tangle network is because we want this prototype to be operated locally without interfering the real-world transactions. Therefore, creating a private Tangle network on the localhost becomes a mandatory prerequisite. After setting up the private Tangle, some trial-and-error need to be done by using open-source code provided by IOTA, in order to make sure the completion of installation. Subsequently, in Node-RED, a testing flow need to be created including with "inject node" which is used to send testing data, "mqtt node" which is used to simulate as physical sensors, and "debug node" which is used to display message payload. MQTT is the abbreviation of Message Queue Telemetry Transport. In this project, we adopt Eclipse Mosquitto to construct the "mqtt server" which will be directly linked to the "mqtt node". Afterward, we upgrade the testing flow by adding some IOTA related nodes such as "mamPublish node" to push data onto the private Tangle network, and "mamFetch node" to pull data from the private Tangle network. By integrating these techniques and nodes, the prototype of a dynamic dashboard platform was generated. This platform can not only receive simulation data but also real-time data. Besides, it can provide a secure and trusting environment for data streams transactions.

III. Background research

A. *Studies between Tangle distributed ledger and IoT*

Talking about the technology of Tangle distributed ledger, although this idea was only introduced to the world a few years ago, many researchers have already dug into this area. In the paper of (Divya & Nagaveni , 2018), the authors pointed out that using technology as Tangle distributed ledger can be a novel replacement of conventional Blockchain technology. Blockchain technology has long-time being considered as the most secure technique in transmitting messages from one node to another due to the prestigious reputation of Bitcoin. Looking back on the Blockchain researches, it is not difficult to find out some advantages of it. For instance, in (Maroufi et al., 2019), Maroufi et al mentioned that Blockchain technology is a distributed ledger, which can provide a secure communication network between old blocks and the new one. By competing with other miners, the algorithm will automatically select the fastest miner who solves the complex mathematical problem, reward the miner with Bitcoins, and give the miner an opportunity in adding a new block with a permanent record on the Blockchain. Moreover, researchers such as (Conoscenti et al., 2016) also gave further details in the applications of Blockchain from 18 different use cases, in some of them specifically designed for IoT scenarios. Regarding the validation of Blockchain, it relies on the high difficulty in solving Proof-of-Work (PoW) to gain the trust of other miners. This process is usually time-consuming but once a miner solves the problem, it will become easy to verify the transaction as every block in the Blockchain was hashed based on the previous block. From these studies, we can understand the benefits of Blockchain, as it can be seen as a feasible solution within secure cryptocurrency transactions by using a decentralized consensus network structure. In (Reyna et al., 2018) (Atlam et al., 2018), challenges of using IoT-based devices such as issues in transmitting trustworthiness, data immutability, and data privacy have been brought to the surface. The introduction of Blockchain technology seems to be a feasible solution toward the transactions between IoT-based devices. However, Blockchain technology still has some limitations that are sometimes overlooked by researchers such as, competitive relationship between miners, time-consuming in redundancy copy during validation, and charging in transaction fees (Pervez et al., 2018). These drawbacks have force people to track for a better solution to replace Blockchain. Besides, Blockchain technology was initially designed and used with powerful computer support, and this means that it may not suitable in a resource-constrained-based IoT devices environment. Therefore, this scenario seems to be far from the idea of IoT reality. Luckily, in 2015, an intuitive cryptocurrency method called IOTA was published, and the came out of IOTA has also introduced the concept of Tangle distributed ledger. The concept of Tangle distributed ledger technology was built upon the Blockchain, which means that it not only contains the benefits of Blockchain but also improve the downsides of it. For instance, the transaction within Tangle does not require a transmitting fee, every transaction node will always be validated by two previous transaction nodes, and multiple transactions are allowed to be transmitted in one Tangle at the same time. Moreover, unlike Blockchain technology, which is mainly known as a cryptocurrency transaction technique, Tangle distributed ledger technology can transmit both cryptocurrency and other formatted of data streams, such as digital signals and text messages. IOTA is mainly designed specifically for the transaction of IoT devices, and this makes Tangle distributed ledger a better option for this research project instead of adopting Blockchain technology.

B. *Studies between Node-RED and IoT*

As we mentioned in the previous paragraph that Node-RED is a flow-based programming tool that was initially designed for the integration of IoT devices, and it also provides open-source API (Application Programming Interface) which allows developers to be able to create their customized functions with JavaScript code. Node-RED technology was first launched by several IBM engineers in 2013 and it quickly became a popular development tool due to the visualized manipulating interface and the connection with MQTT servers. Recent papers (Lekic & Gardasevic , 2018) (Blackstock & Lea, 2014) have shown the technique of integration or implementation between IoT and Node-RED. In (Lekic & Gardasevic , 2018), the authors used a Raspberry Pi, which contains temperature and humidity sensors, to transfer data into the Cloud server by using Node-RED as an intermediate connector. This concept is considerably similar to our research purpose, however, our project has moved the goal into another level. Apart from receiving data from sensors and transmitting it onto the Cloud, we also integrate Tangle distributed ledger technology with Node-RED to escalate the security and trust within data transactions. On the other hand, in (Blackstock & Lea, 2014), the researchers introduced a distributed Node-RED platform to solve the connection problem in Web of Thing (WoT), which is a large group of connected devices in a web-based concept. Meanwhile, some limitations of Node-RED have been pointed out in this paper. For instance, unlike the multi-receiving capability of a traditional processor, the node on Node-RED can only accept one input at a time. However, this problem can be solved by developing multiple node-connected structures in Node-RED flow. In addition, Node-RED technology can be also used in Human-Computer Interaction (HCI) area such as combining it with Amazon Alexa, which is a voice service device (Rajalakshmi & Shahnasser, 2017). As we know that Amazon Web Service (AWS) provides cloud computing service and support MQTT brokers to realize M2M (Machine to Machine) interaction. Therefore, using "mqtt node" as a broker in Node-RED flow allows researchers to push data, which received from physical sensors, onto AWS. Meanwhile, when we give commands to Alexa, it will connect to AWS, pull the specific data, and reveal them vocally. By using a similar concept, in this research project, we will also use "mqtt node" to catch data streams but replace the publish node as "mamPublish" and receive node as "mamFetch". Many other functions in integrating IoT devices and Node-RED have been revealed by researchers. However, in this research project, we will mainly focus on the security of data streams transaction from smart meters and sensors.

IV. Requirement Analysis

A. Functional requirements

Functional requirements mean that the most basic functions which should be provided by the system designer or developer. From the overall perspective of a secure data transaction platform, the functional requirements are as follows:

- Installation steps of the private Tangle network should be easy for any user who does not have specific IT (Information Technology) background.
- Every node in the Node-RED flow should be designed clearly and easily to understand.
- The integration process of Tangle distributed ledger and Node-RED needs to be provided.
- Example and testing code within the system structure need to be offered.
- Any further instruction should be announced with detailed guidelines.
- The platform should run smoothly with low latency in data transactions.

B. Non-functional requirements

Non-functional requirements mean that some hidden requirements which are required but will not be mentioned. These requirements are usually focused on the fields of reliability, performance, and security.

- *Reliability:* This platform should be able to receive data from any physical device, such as smart meters and sensors on Raspberry Pi, or non-physical devices, such as emulated signals and messages. Furthermore, this system needs to be designed suitable in any computer OS (Operation System), and the platform needs to be feasible on any webpage browser. All data transactions should be recorded for further queries.
- *Performance:* All data transactions on this platform should be required in real-time. Hence, data transfer time among sensors, nodes, and the platform should be less than a few seconds, hopefully, to be completed in 2 seconds. The design of this platform needs to consider in minimizing the pending time of users.
- *Security:* By using IoT-based devices like smart meters and sensors may expose the platform in a dangerous environment. Because highly dependent on the Internet network, data transactions are easily intercepted by attackers. Therefore, a secure method in transmitting data need to be provided, and in this research project, we adopted Tangle distributed ledger.

V. Methodology

A. Design

Detail of building steps and technical processes will be provided here. More information can be found at "https://docs.iota.org/docs/compass/0.1/how-to-guides/set-up-a-private-tangle". In this project, every installation step was made under Ubuntu 18.04 OS environment and using Python as a testing programming language.

1) Install Docker engine

Before setting up a private Tangle network on the localhost, a Docker engine must be installed mainly because IOTA network is designed in running under Docker environment. Docker is a set of lightweight Platform as a Service (PaaS) that uses OS-level virtualization to deliver software in packages. The first step is to set up the repository by using the command as follows:

```
$ sudo apt-get update
$ sudo apt-get install \
   apt-transport-https \
   ca-certificates \
   curl \
   gnupg-agent \
   software-properties-common
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg |
sudo apt-key add –
$ sudo apt-key fingerprint 0EBFCD88
$ sudo add-apt-repository \
```

Afterward, install Docker engine by using the command as follows:

```
$ sudo apt-get update
$ sudo apt-get install docker-ce docker-ce-cli containerd.io
```

After installing the Docker engine, you may verify it by running a "hello-world" image.

```
$ sudo docker run hello-world
```

2) Install Docker compose

The next step is to install Docker compose. Docker compose relies on Docker engine, so make sure to fully install Docker engine before entering this phase. Likewise, the installation process varies upon different OSs. The commands below should be only used under Linux OS environment:

```
$ sudo curl -L
"https://github.com/docker/compose/releases/download/1.26.0/
docker-compose-$(uname -s)-$(uname -m)" -o
/usr/local/bin/docker-compose
$ sudo chmod +x /usr/local/bin/docker-compose
$ sudo ln -s /usr/local/bin/docker-compose /usr/bin/docker-
compose
```

Test the installation by using the command as below:

```
$ docker-compose --version
```

If the installation was successful, you will see the feedback log as follows:

```
docker-compose version 1.26.0, build 1110ad01
```

3) Install Git

After Docker installation, the next procedure is to install Git. Git is a distributed version control software, and it is often used to share source code or text messages. As we mentioned in previous sections that Tangle distributed ledger is based on the open-resources provided by IOTA. Therefore, after installing Git, we can simply use the "git" command to clone the resource from a specific website instead of building by ourselves. The installation command is as follows:

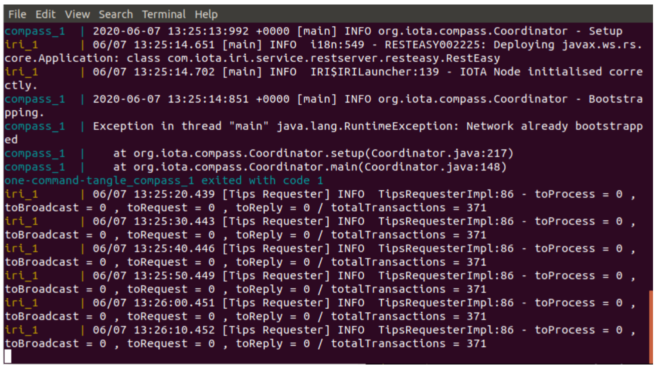
```
$ apt-get install git
$ add-apt-repository ppa:git-core/ppa
$ apt update; apt install git
```

4) Set up the private Tangle

With the prerequisite support of Docker and Git, we can now begin setting up a local private Tangle network. The first step is to clone the repository by using "git" command as shown below:

```
$ git clone https://github.com/iota-community/one-command-
tangle.git
```

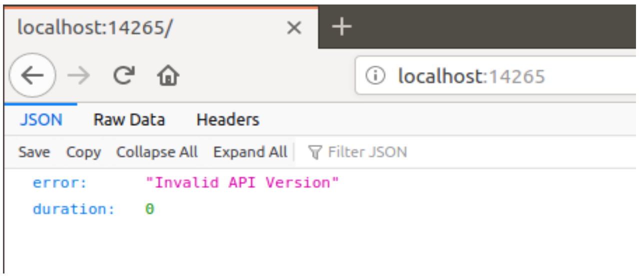
Next step, go to your private Tangle's directory and execute "docker-compose up" in the command shell. You should get a log similar to Figure 2.



```
compass_1  | 2020-06-07 13:25:13:992 +0000 [main] INFO org.iota.compass.Coordinator - Setup
iri_1      | 06/07 13:25:14.651 [main] INFO  i18n:549 - RESTEASY002225: Deploying javax.ws.rs.
core.Application: class com.iota.iri.service.restserver.resteasy.RestEasy
iri_1      | 06/07 13:25:14.702 [main] INFO  IRI$IRILauncher:139 - IOTA Node initialised corre
ctly.
compass_1  | 2020-06-07 13:25:14:851 +0000 [main] INFO org.iota.compass.Coordinator - Bootstra
pping.
compass_1  | Exception in thread "main" java.lang.RuntimeException: Network already bootstrapp
ed
compass_1  |    at org.iota.compass.Coordinator.setup(Coordinator.java:217)
compass_1  |    at org.iota.compass.Coordinator.main(Coordinator.java:148)
one-command-tangle_compass_1 exited with code 1
iri_1      | 06/07 13:25:20.439 [Tips Requester] INFO  TipsRequesterImpl:86 - toProcess = 0 ,
toBroadcast = 0 , toRequest = 0 , toReply = 0 / totalTransactions = 371
iri_1      | 06/07 13:25:30.443 [Tips Requester] INFO  TipsRequesterImpl:86 - toProcess = 0 ,
toBroadcast = 0 , toRequest = 0 , toReply = 0 / totalTransactions = 371
iri_1      | 06/07 13:25:40.446 [Tips Requester] INFO  TipsRequesterImpl:86 - toProcess = 0 ,
toBroadcast = 0 , toRequest = 0 , toReply = 0 / totalTransactions = 371
iri_1      | 06/07 13:25:50.449 [Tips Requester] INFO  TipsRequesterImpl:86 - toProcess = 0 ,
toBroadcast = 0 , toRequest = 0 , toReply = 0 / totalTransactions = 371
iri_1      | 06/07 13:26:00.451 [Tips Requester] INFO  TipsRequesterImpl:86 - toProcess = 0 ,
toBroadcast = 0 , toRequest = 0 , toReply = 0 / totalTransactions = 371
iri_1      | 06/07 13:26:10.452 [Tips Requester] INFO  TipsRequesterImpl:86 - toProcess = 0 ,
toBroadcast = 0 , toRequest = 0 , toReply = 0 / totalTransactions = 371
```

Figure 2. A successful connection log of private Tangle network in the console.

To test the connection of your private Tangle, you can open up a browser and type the URL as "http://localhost:14265" to interact with it. The port 14265 refers to the IRI (IOTA Reference Implementation) node which was already embedded within the private Tangle network. The feedback should be similar to Figure 3.



```
error:      "Invalid API Version"
duration:   0
```

Figure 3. Successful interaction with private Tangle on the browser.

So far, we have finished the most basic installation of private Tangle network. However, this structure only contains the Tangle distributed ledger function that requires for this project, not including functions for cryptocurrency transaction testing. The next process is to get specific information from the private Tangle by using open-source code provided by IOTA. Before testing, an IOTA wallet can be chosen to install. The main idea of setting up an IOTA wallet is to offer users a friendly interface to check their balance when they make transactions with each other. In this project, we set up an IOTA wallet mainly for source code testing. There are two forms of IOTA wallet, one is called "Light wallet", and another is called "Trinity". The difference between them is the connection toward the Internet. For "Light wallet", it can be tested in a local environment or with an HTTP connection. For "Trinity", you must expose your IRI node to the Internet through an HTTPS connection. Likewise, we will only use the local one, which is "Light wallet", to do the testing, and the latest version of "Light wallet" can be found at "https://github.com/iotaledger/wallet/releases". This resource is provided and maintained by IOTA engineers. After the installation, you can use a sample seed as "SEED999999999999999999999999999999999999999999 99999999999999999999999999999999999" to login. The seed is a unique password for users to prove the ownership of messages or cryptocurrency that they transmit through the Tangle network. The sample seed we used in this project was also provided by IOTA, and the reason for using this sample seed is that there are already some IOTA tokens, which can be used in transaction testing, within it. Therefore, we can verify our code without interfering with real-world transactions. A successful login image is shown in Figure 4.

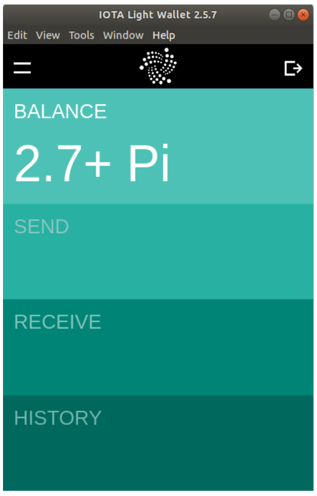


Figure 4. A successful login image of the IOTA Light Wallet.

Now we can use the open-source API to test our IRI node. IRI node is an open-source Java software specific run on nodes in the IOTA network, where users can make transactions among each other. In more precisely, you can consider every rectangle box within the Tangle structure that shows in Figure 1 as an individual IRI node. Each of these nodes can store valid transactions in a ledger and can also validate new transactions. Every time a user publishes a new transaction, a new IRI node will be appended to the Tangle network and this transaction will also be recorded in the ledger, waiting for being validated by two previous nodes. As an example, if we want to get the information of the node that we have created within the private Tangle, we can use code as follows:

```
import urllib2
import json

command = {"command": "getNodeInfo"}
stringified = json.dumps(command)
headers = {
    'content-type': 'application/json',
    'X-IOTA-API-Version': '1'
}
request = urllib2.Request(url="http://localhost:14265",
data=stringified, headers=headers)
returnData = urllib2.urlopen(request).read()
jsonData = json.loads(returnData)
print jsonData
```

In this project, we choose Python as our programming language. By executing the code above, you should get a feedback log similar to below:

```
{
 "appName": "IRI",
 "appVersion": "1.7.0-RELEASE",
 "jreAvailableProcessors": 8,
 "jreFreeMemory": 2115085674,
 "jreVersion": "1.8.0_191",
 "jreMaxMemory": 20997734400,
 "jreTotalMemory": 4860129502,
 "latestMilestone":
"CUOENIPTRCNECMVOXSWKOONGZJICAPH9FIG9F9K
YXF9VYXFUKTNDCCLLWRZNUHZIGLJZFWPOVCIZA9
999",
 "latestMilestoneIndex": 1050373,
 "latestSolidSubtangleMilestone":
"CUOENIPTRCNECMVOXSWKOONGZJICAPH9FIG9F9K
YXF9VYXFUKTNDCCLLWRZNUHZIGLJZFWPOVCIZA9
999",
 "latestSolidSubtangleMilestoneIndex": 1050373,
 "milestoneStartIndex": 1050101,
 "lastSnapshottedMilestoneIndex": 1039138,
  "neighbors": 7,
  "packetsQueueSize": 0,
  "time": 1554970558971,
  "tips": 9018,
  "transactionsToRequest": 0,
  "features": [
    "snapshotPruning",
    "dnsRefresher",
    "tipSolidification"
   ],
  "coordinatorAddress":
"EQSAUZXULTTYZCLNJNTXQTQHOMOFZERHTCGTX
OLTVAHKSA9OGAZDEKECURBRIXIJWNPFCQIOVFVV
XJVD9",
   "dbSizeInBytes": 144800000,
   "duration": 0
  }
```

5) Set up Node-RED

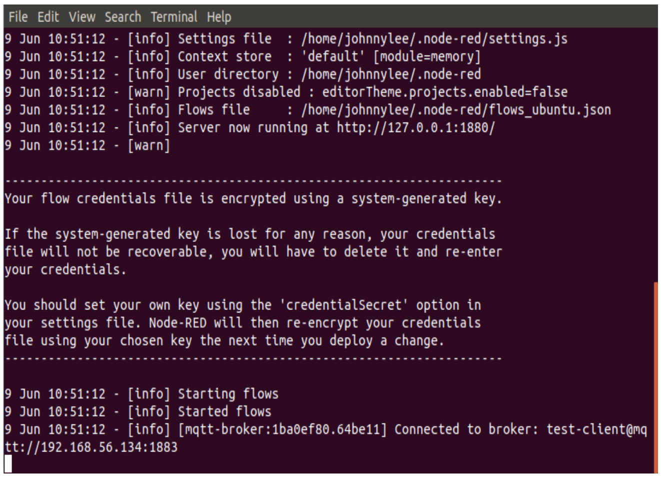
Getting started with Node-RED, one thing that needs to be confirmed is your installation environment. As mentioned above, we chose Ubuntu 18.04 OS as our installation environment in this project. Node-RED provides various options for installation such as running locally, installing on Raspberry Pi, or even deploying on the cloud. More information can be found at "https://nodered.org/docs/getting-started/". For a Linux environment, the first thing we recommend you to do is to make sure that if "npm" function is installed and workable by running the following command:

```
$ sudo apt install build-essential git
```

Subsequently, running the following command will download and run the script, which will remove the old version of Node-RED and Node.js and replace them with the latest version. If you want to review the contents of the script first, you can view it at "https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered". The Node-RED installation command is shown as below:

```
$ bash <(curl -sL https://raw.githubusercontent.com/node-
red/linux-installers/master/deb/update-nodejs-and-nodered)
```

After installation, you can use the "node-red" command to run Node-RED in a terminal, and press "Ctrl + C" to stop Node-RED by closing the terminal. A successful start-up image is shown in Figure 5.

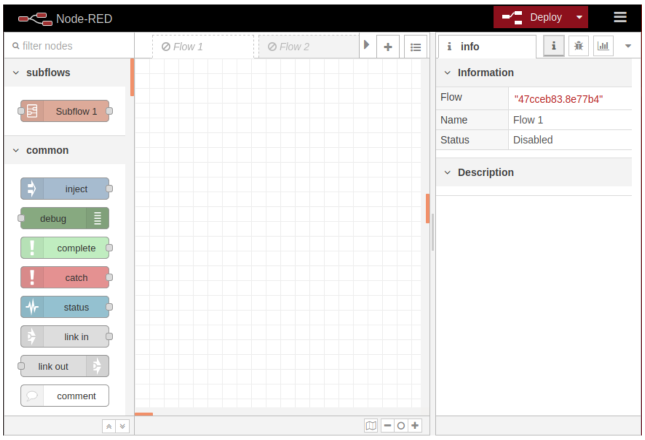


```
9 Jun 10:51:12 - [info] Settings file  : /home/johnnylee/.node-red/settings.js
9 Jun 10:51:12 - [info] Context store  : 'default' [module=memory]
9 Jun 10:51:12 - [info] User directory : /home/johnnylee/.node-red
9 Jun 10:51:12 - [warn] Projects disabled : editorTheme.projects.enabled=false
9 Jun 10:51:12 - [info] Flows file     : /home/johnnylee/.node-red/flows_ubuntu.json
9 Jun 10:51:12 - [info] Server now running at http://127.0.0.1:1880/
9 Jun 10:51:12 - [warn]

---------------------------------------------------------------------
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
---------------------------------------------------------------------

9 Jun 10:51:12 - [info] Starting flows
9 Jun 10:51:12 - [info] Started flows
9 Jun 10:51:12 - [info] [mqtt-broker:1ba0ef80.64be11] Connected to broker: test-client@mq
tt://192.168.56.134:1883
```

Figure 5. Successful start-up of Node-RED in a terminal.

6) Node-RED environment

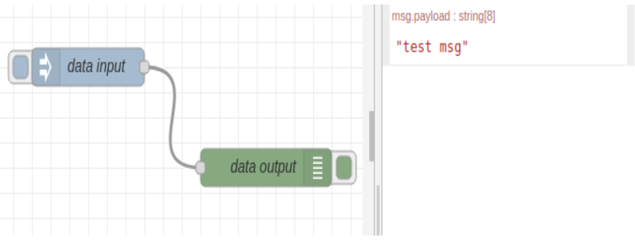
There are three basic components within the Node-RED environment: 1. Node panel (left part), 2. Flow panel (central part), and 3. Information panel (right part). Open up the browser and type in URL "http://localhost:1880" to start up the Node-RED user interface. A successful start-up image is shown in Figure 6.



Flow "47cceb83.8e77b4"
Name Flow 1
Status Disabled

Figure 6. Successful start-up of Node-RED in the browser.

Node-RED is a powerful and flexible development tool specially designed for IoT applications. Users can easily deploy their workflow by dragging specific nodes from the node panel to flow panel and connect different nodes by lines. Under this circumstance, users can design their workflow by combining nodes such as "input node", "output node", "mqtt node", or even using "function node" to customize their workflow. An example workflow is shown in Figure 7.



msg.payload : string[8]
"test msg"

Figure 7. A simple input-output workflow in Node-RED.

Many researchers have already published papers in collaborating on the usage of IoT applications and Node-RED such as temperature monitoring, acoustic censoring, and optical controlling. However, none of them focused on the security of data stream transactions. Therefore, a practical implementation process of a secure data transaction platform will be given in the next section.

B. Implementation

In this section, a vivid example of constructing a secure data transaction platform will be given. First, open up a terminal to start up the private Tangle like the one shown in Figure 2. Then open another terminal to initiate the Node-RED as the one shown in Figure 5. After executing the private Tangle and Node-RED, we can start to design our workflow by entering "http://localhost:1880" on the browser. IOTA is not a standard function of Node-RED, therefore, we need to import it onto the palette as shown in Figure 8, Figure 9, and Figure 10.

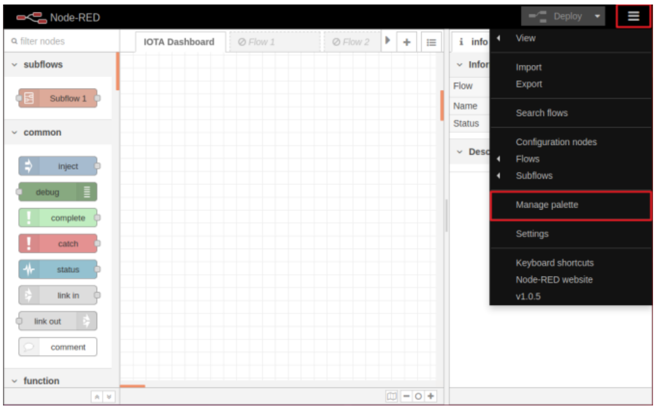
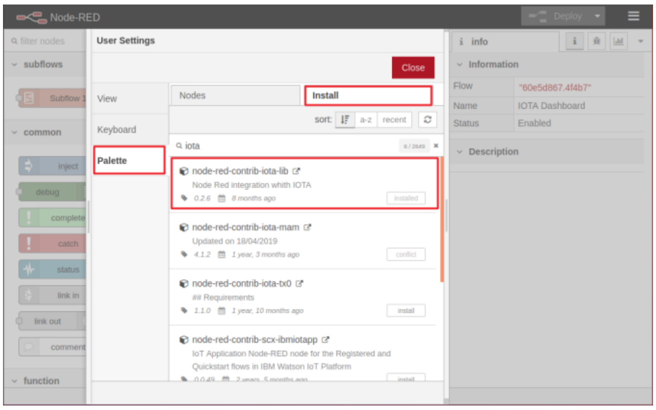


Figure 8. Click "Manage palette" in the drop-down list.



iota 6 / 2649

node-red-contrib-iota-lib
Node Red integration whith IOTA
0.2.6 8 months ago installed

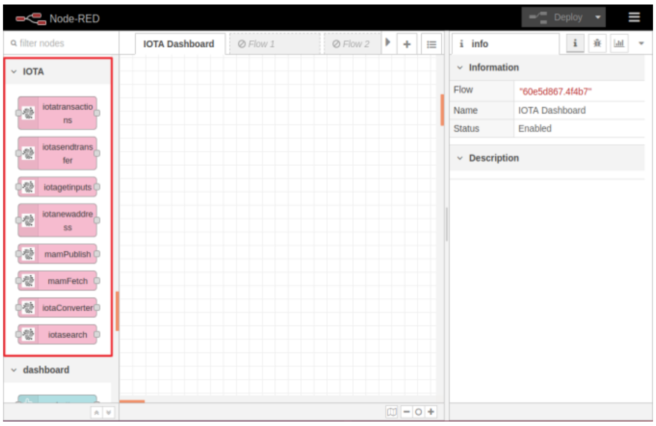
node-red-contrib-iota-mam
Updated on 18/04/2019
4.1.2 1 year, 3 months ago conflict

node-red-contrib-iota-tx0
Requirements
1.1.0 1 year, 10 months ago install

node-red-contrib-scx-ibmiotapp
IoT Application Node-RED node for the Registered and Quickstart flows in IBM Watson IoT Platform

Flow "60e5d867.4f4b7"
Name IOTA Dashboard
Status Enabled

Figure 9. Search "iota" then install "node-red-contrib-iota-lib".



Flow "60e5d867.4f4b7"
Name IOTA Dashboard
Status Enabled

Figure 10. IOTA related nodes have been installed onto the node panel.

We create two groups, which are the control group and experiment group, to demonstrate different ways of receiving and displaying data. For the control group, we imitate a scenario of receiving data in a traditional way that is "input sensor's data", "transmitting sensor's data", and "output sensor's data". As the red rectangle part shown in Figure 11, we deploy "sensor1" and "sensor2" as two virtual sensors to receive data, using "sensor broadcaster" to broadcast the signal, using "sensor receiver" to receive the signal, using "msg.payload" to display the message on Node-RED, and using "sensor output" to show the message on the platform. In "sensor broadcaster" and "sensor receiver", we adopt Eclipse Mosquitto as the MQTT broker to transmit the sensor's data.

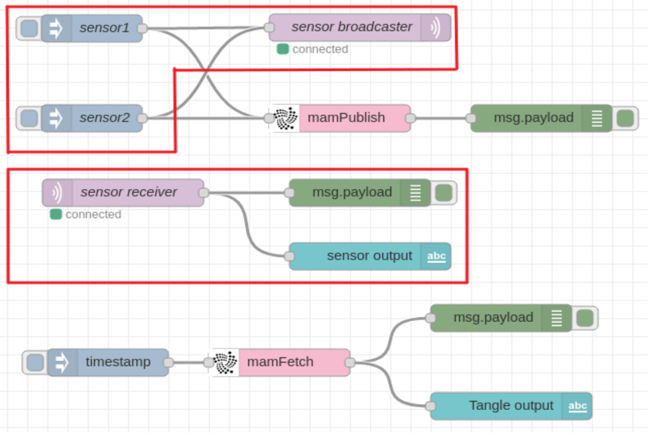


Figure 11. The control group (simulating traditional signal transaction).

On the other hand, for the experiment group, we publish data received from the sensor to Tangle distributed ledger by using "mamPublish" node, and request data from Tangle distributed ledger by using "mamFetch" node. In this scenario, as the red rectangle part shown in Figure 12, we deploy two "msg.payload" to examine the data by showing them on the Node-RED, one "timestamp" as a timer to automatically request data from Tangle distributed ledger periodically, and one "Tangle output" to display the message on the platform.

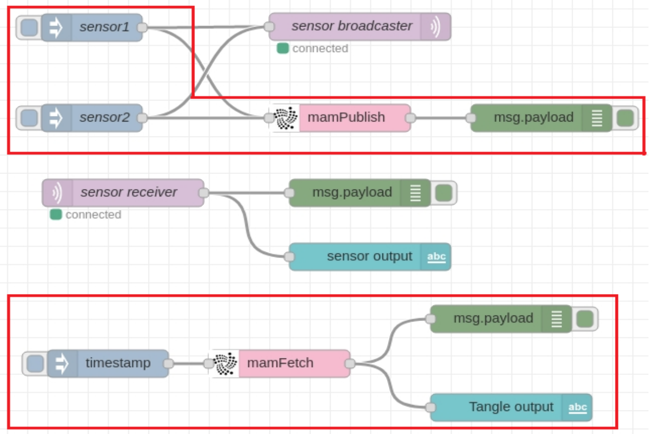
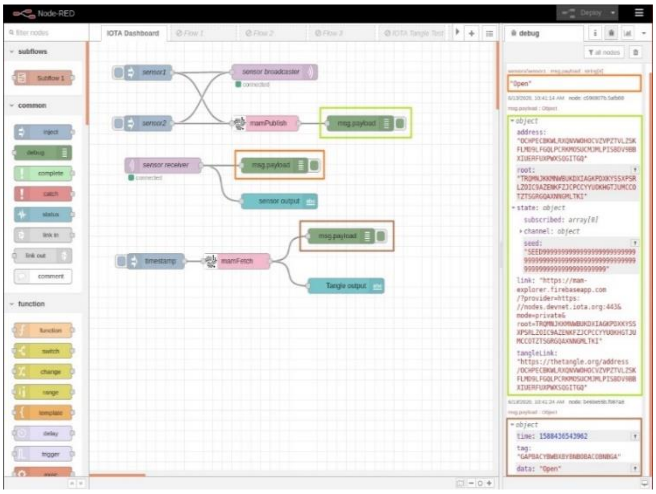


Figure 12. The experiment group (transmitting signal through IOTA Tangle).

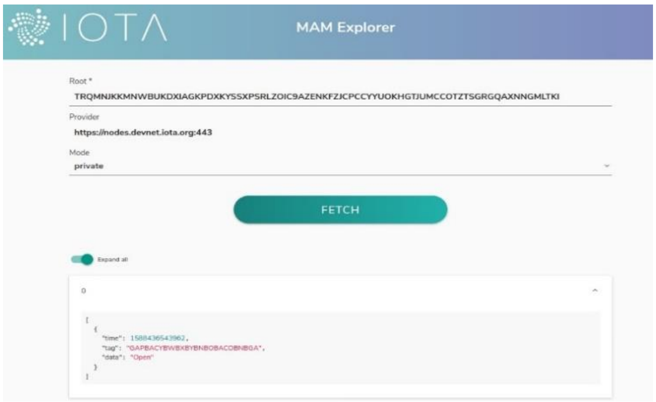
In this research project, we imitate sensor1 and sensor2 as a power switch to send "open" and "close" separately. If we click sensor1 to trigger the transaction, the data will be sent to the "sensor broadcaster" and "mamPublish", and we should get the result transmitted by the "mqtt broker" and "Tangle distributed ledger". As Figure 13 shows, after triggering the sensor to transmit the data, the output from MQTT responds first because it was deployed locally (the orange rectangle). Then we catch the feedback from Tangle distributed ledger because the data needs to be transmitted through the Internet (the light-green rectangle). This feedback is slower than the MQTT broker and it usually takes a few seconds for responding. The feedback from "mamPublish" is information such as the seed that we used to authenticate the transaction and the address where the transaction took place. Finally, the timer was triggered and the feedback we get from "mamFetch" is exactly the one sent by the sensor (the brown rectangle). This means that our transaction has successfully traveled through the Tangle network.



```
"Open"
msg.payload : Object
▾object
  address:
  "OCHPECBKWLRXQNVWOHOCVZVPZTVLZSKFLMD9LFGQLPCRKMOSUCMJMLPISBDV9BBXIUERFUXPWKSOGITGQ"
  root:
  "TRQMNJKKMNWBUKDXIAGKPDXKYSSXPSRLZOIC9AZENKFZJCPCCYYUOKHGTJUMCCOTZTSGRGQAXNNGMLTKI"
  ▾state: object
    subscribed: array[0]
    ▸channel: object
    seed:
    "SEED9999999999999999999999999999999999999999999999999999999999999999999999999999"
  link: "https://mam-explorer.firebaseapp.com/?provider=https://nodes.devnet.iota.org:443&mode=private&root=TRQMNJKKMNWBUKDXIAGKPDXKYSSXPSRLZOIC9AZENKFZJCPCCYYUOKHGTJUMCCOTZTSGRGQAXNNGMLTKI"
  tangleLink:
  "https://thetangle.org/address/OCHPECBKWLRXQNVWOHOCVZVPZTVLZSKFLMD9LFGQLPCRKMOSUCMJMLPISBDV9BBXIUERFUXPWKSOGITGQ"
msg.payload : Object
▾object
  time: 1588436543962
  tag:
  "GAPBACYBWBXBYBNBOBACOBNBGA"
  data: "Open"
```

Figure 13. Node-RED layout after triggering the sensor.

If we go to the IOTA webpage and search for the transaction which we have made, we can find the history data that was recorded within the Tangle distributed ledger. As Figure 14 shows, every transaction will be assigned with a unique timestamp as a certification.



Root *
TRQMNJKKMNWBUKDXIAGKPDXKYSSXPSRLZOIC9AZENKFZJCPCCYYUOKHGTJUMCCOTZTSGRGQAXNNGMLTKI

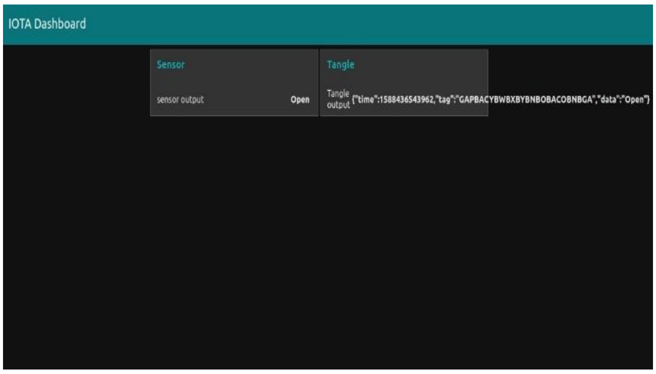
Provider
https://nodes.devnet.iota.org:443

Mode
private

```
0
[
  {
    "time": 1588436543962,
    "tag": "GAPBACYBWBXBYBNBOBACOBNBGA",
    "data": "Open"
  }
]
```

Figure 14. Transaction record on the IOTA website.

Finally, as Figure 15 shows, the secure data transaction platform can be accessed by typing "http://localhost:1880/ui" on the browser. This action requires the activation of Node-RED, otherwise it will not be started. In this research project, we only create two labels to display the message that we got from the MQTT broker and Tangle distributed ledger. However, this is only a simple example of integrating IOTA Tangle and Node-RED technology. In the future, this technique can be improved and extended for various purposes in different fields.



sensor output Open

Tangle output {"time":1588436543962,"tag":"GAPBACYBWBXBYBNBOBACOBNBGA","data":"Open"}

Figure 15. Demonstration of a secure data transaction platform.

C. Analysis

It is very important to evaluate the system before launch because an underestimated design might cause a potential impact in the future. Therefore, after designing, an analysis has to be made in order to make sure that this platform is secure enough in supporting data stream transactions. As this is only a research project, some insufficiencies may exist in the platform structure. However, the adoption of Tangle distributed ledger has significantly enhanced the security of data stream transactions mainly because IOTA technology was constructed upon the MAM protocol. MAM (Masked Authenticated Messaging) is a second layer data communication protocol that fulfills the data stream integrity in the IOTA consensus. MAM adds the functionality in emitting and access encrypted data streams of Tangle distributed ledger regardless of the size or cost of devices (Handy, 2017). Regarding the property of MAM, it successfully fulfills the requirement in the trust and privacy of data transactions. In the platform we designed, users can publish or request data at any time as long as they provide relatively channel seed or address. The bundle of MAM is based on the structure of Merkle Tree, which is also known as Hash Tree. The idea of Merkle Tree was first introduced by Ralph Charles Merkle and is commonly used in documentation and P2P systems (Merkle, 1988). During the transactions, various types of information are allowed to be transmitted, including but not limited to digital currency or text message. In the "mamPublish" node, a channel seed with 81 trytes needs to be given. When the data pass-through "mamPublish", the original data will be separated into indexes, and the seed will be used as a private key to do the hash. It can completely secure the integrity and trust of the original information by distributing data into small pieces, hashing, and integrating them as a Merkle root. Through the hash function of Merkle Tree, every single modification of the original data will generate a different hash value, and make the old hash value invalid. Therefore, we can guarantee the correctness and integrity of the original data. On the other hand, once the data has been published to Tangle distributed ledger, it will be assigned with a specific address (root). Then users can enter this address into "mamFetch" node to request specific information from the Tangle.

Making the transaction more secure, MAM also provides three different types of channel modes, which are "Public", "Private", and "Restricted", to support data transmission. In public mode, everyone can inquire about the data as long as he or she knows the address. It works similarly to broadcasting, and it can also be used as public announcements. In private mode, the data can only be accessed by the publisher. It works similarly as an encrypted message, and this can stop other people from knowing the data content. For restricted mode, it is similar to private mode but with an additional secret key. The publisher can decide the secret key content when he or she published data onto the Tangle distributed ledger, and only the people who carry with the correct address and secret key are allowed to fetch and decrypt the message. This function has enhanced the strength and reliability of data stream transactions made through the Tangle distributed ledger.

VI. Future work and conclusion

In this research project, we have proposed a feasible solution in securely transmitting data streams through the integration technology of Tangle distributed ledger and Node-RED. Throughout the latest papers, IOTA technology has been recommended by many researchers due to its high suitability with IoT devices. In (Atlam et al., 2018), IOTA technology was mentioned as a future research direction in replacing Blockchain technology. Besides, the author of (Alexander, 2018) also mentioned that the appearance of IOTA in late 2015 was mainly focusing on tackling the limitation of Blockchain technique in such as transaction latency and network congestion problems. But why is it important in reducing the data transaction time? Imagine that in the near future, almost every device is closely connected through the Internet. For example, people can control all electronic furniture in their homes, all public facilities on the road can be communicated with each other, and all vehicles that run on the street are autopilot. Actually, this “Future” is now in progress and has already been under testing in some cities. A research report was given by Cisco in 2011, (Evans, 2011), has estimated that the usage of IoT devices in 2015 will be 25 billion around the globe, and by the time of 2020, this number will be double and reach 50 billion. This number predicted by Cisco was calculated from the connected device used by each individual. However, this estimate did not take into account rapid advances in the Internet or device technology which means that the actual number of connected IoT devices could be more than 50 billion currently, in 2020. Without a doubt, the adoption of Tangle distributed ledger technique can smoothly replace Blockchain in an IoT environment. In this research project, we have created a prototype of a secure data transaction platform by using the technique of IOTA Tangle distributed ledger and Node-RED. This platform can be easily constructed by people who do not have much knowledge in the IT area and also suitable for those who do not have an engineering background. However, IOTA is still a relatively innovative technology. The resources that can be found online are remaining limited and lots of software applications are also under construction. Therefore, this research project aims to set an example for others to follow. More research and further investigation are still needed to be made to improve the implementation of secure data stream transactions within an IoT environment.

Acknowledgement

This work was directed by Dr. Kamyar Mehran and supported by Ph.D. candidate Mr. Seyed Amir Alavi, both from the School of Electronic Engineering and Computer Science, Queen Mary University of London. The author would like to thank them for their constructive comment and advice during the whole process of this project. This research project would not have been produced without their honor of great help.

References

Alexander, R. (2018). ‘IOTA - Introduction to the Tangle Technology: Everything you need to know about the revolutionary blockchain alternative’, *Independently published*.

Atlam, H.F., Alenezi, A., Alassafi, M.O., & Wills, G.B. (2018). Blockchain with Internet of Things: benefits, challenges, and future directions. *International Journal of Intelligent Systems and Applications*, 10(6), pp. 40-48. doi:10.5815/ijisa.2018.06.05

Blackstock, M., & Lea, R. (2014) ‘Toward a Distributed Data Flow Platform for the Web of Things (Distributed Node-RED)’, *WoT '14: Proceedings of the 5th International Workshop on Web of Things*, pp. 34-39. https://doi.org/10.1145/2684432.2684439

Conoscenti, M., Vetro, A., & Martin, J.C.D. (2016). ‘Blockchain for the internet of things: A systematic literature review’, *2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA), Agadir*, pp. 1-6, doi: 10.1109/AICCSA.2016.7945805

Divya, M., & Nagaveni, B.B. (2018). 'IOTA-Next Generation Block chain', *International Journal of Engineering and Computer Science*, 7(4), pp. 23823-23826

Evans, D. (2011). 'The Internet of Things How the Next Evolution of the Internet Is Changing Everything', *Cisco Internet Business Solutions Group (IBSG)*.

Handy, P. (2017). *Introducing Masked Authenticated Messaging*. Available at: https://blog.iota.org/introducing-masked-authenticated-messaging-e55c1822d50e (Access: 30 July 2020)

IOTA. (2020). *IOTA - overview*. Available at: https://docs.iota.org/docs/getting-started/0.1/introduction/overview (Access: 12 July 2020)

IOTA Support. (2020). *An introduction to IOTA*. Available at: https://iotasupport.com/whatisiota.shtml (Access: 1 August 2020)

Lekic, M., & Gardasevic, G. (2018). 'IoT sensor integration to Node-RED platform', *17th International Symposium INFOTEH-JAHORINA (INFOTEH)*, pp. 1-5. doi: 10.1109/INFOTEH.2018.8345544

Maroufi, M., Abdolee R., & Tazekand B.M. (2019) 'On the Convergence of Blockchain and Internet of Things (IoT) Technologies*', Journal of Strategic Innovation and Sustainability, 14(1). https://doi.org/10.33423/jsis.v14i1.990*

Merkle, R.C. (1988). A Digital Signature Based on a Conventional Encryption Function. *In: Pomerance C. (eds) Advances in Cryptology — CRYPTO '87. CRYPTO 1987. Lecture Notes in Computer Science, vol 293. Springer, Berlin, Heidelberg*. https://doi.org/10.1007/3-540-48184-2_32

Node-RED. (2020). *About Node-RED*. Available at: https://nodered.org/about/ (Access: 15 July 2020)

Pervez H., Muneeb M., Irfan M.U., & Haq I.U. (2018). 'A Comparative Analysis of DAG-Based Blockchain Architectures', *International Conference on Open Source Systems and Technologies (ICOSST), Lahore, Pakistan*, pp. 27-34. doi: 10.1109/ICOSST.2018.8632193

Rajalakshmi, A., & Shahnasser, H. (2017). 'Internet of Things using Node-Red and alexa', *17th International Symposium on Communications and Information Technologies (ISCIT),* pp. 1-4. doi: 10.1109/ISCIT.2017.8261194

Reyna, A., Martin, C., Chen, J., Soler, E., & Diaz, M. (2018). 'On blockchain and its integration with IoT. Challenges and opportunities', *Future Generation Computer Systems*, 88, pp. 173-190

Satoshi Watch. (2017). *IOTA -The Winner Takes it All*. Available at: https://satoshiwatch.com/coins/iota/in-depth/iota-the-winner-takes/ (Access: 10 July 2020)

Scheepers, R., & Middleton, C. (2013). 'Personal ICT Ensembles and Ubiquitous Information Systems Environments: Key Issues and Research Implications', *Communications of the Association for Information Systems*, 33, https://doi.org/10.17705/1CAIS.03322