Remote Authentication of IoT devices based upon Fog Computing

Manabhanjan Pradhan[1] and Sujata Mohanty[1]

National Institute of Technology, Rourkela, Odisha, India
mannpradhan@protonmail.com

Abstract. This paper presents a fog based system model for IoT architecture for remote authentication. It ensures that the communicating devices are authentic and reject any malicious requests, securing the communication held between devices. In the proposed architecture, only genuine users can access IoT devices through the nearest fog node. Every time a device associated with a user sends a request, it is verified by the blockchain-enabled fog node; thereby, mutual authentication is achieved. Also, the proposed work is compared with some competent schemes and found to achieve security features, such as data integrity, availability, and mutual authentication.

Keywords: Fog computing, IoT, Blockchain, Mutual authentication

1 Introduction

The interconnection between different devices through a network is popularly known as the Internet of Things (IoT) [11]. Kevin Ashton first proposed the IoT technology in 1999, which has become a promising research area nowadays. For instance, smart bulb, smart TV, and smart assistance, such as Alexa, have become a part of our daily life. The IoT devices work independently with dedicated sensors, known as, nodes which gather information from the environment [1]. With a predefined set of rules, they execute the instructions. In general, IoT devices consist of three parts, namely, tags, sensors, and RFIDs. As many IoT devices are connected and communicated with each other, security is the utmost concern. A breach in security may damage the functionalities and also resulting personal data theft [8]. To mitigate this, these devices can be associated with mutual authentication for each communication, and thereby, any adversary could not get access to the central architecture.

Blockchain was first introduced by Satoshi Nakamoto, who proposed a decentralized technique for digital cryptocurrency named Bitcoin [9]. He proposed a technique that overcomes the trust issues involving the third party. Here, two parties can communicate and share their resources, whereas the decisions are made by the nodes involved in the network. Blockchain can be used for access management, authentication, and many more real-life applications. It provides

basic security services, such as authentication, data integrity, data confidentiality, and non-repudiation. A blockchain consists of a database, miners, and network of nodes. After 2009, many blockchains have been implemented. Every blockchain has a different mechanism for mining a new node. While Bitcoin needs Proof of Work(PoW), Ethereum needs Proof of Stake (PoS), and Hyperledger needs Practical Byzantine Fault Tolerance (PBFT) approach.

Authentication in Blockchain technology is a promising field, and much research is going on in this area [2][3][5]. Hammi et al. proposed an authentication mechanism based on blockchain with a focus on various security issues[4]. Khalid et al. developed a blockchain-based authentication mechanism for IoT [10]. But later, it was found to be susceptible to malicious requests. Nguyen et al. proposed a prototype implementation in a real data sharing scenario for secure Electronic Health Records (EHRs) [6]. M. Tahir et al. proposed an authentication framework using a probabilistic model [12].

We present a fog based system model for remote authentication. It ensures mutual authentication between the communicating devices and rejects any requests which are found to be malicious. In the proposed model, only the authentic users can access IoT devices through the nearest fog node. Every time a device associated with a user sends a request, it is duly verified by the blockchain-enabled fog node. The proposed scheme can withstand passive and active attacks, such as spoofing attack, man-in-the-middle attack, and fog node impersonation attack [7]. Also, the proposed work is compared with some competent schemes and found to achieve security features, such as data integrity, availability, and mutual authentication. The proposed work is of low computational cost along with less communication overhead as compared to existing literature.

The rest of the paper is organized as follows. Section 2 demonstrates the proposed system model for remote authentication. Section 3 discuss and analyzes the experiment with results. Finally, the paper concludes in Section 4.

2 Proposed System Model

According to the Open Web Application Security Project (OWASP), one of the major vulnerabilities in IoT communication is weak authentication and inefficient mobile interfaces. We proposed a blockchain-based model to overcome the problem stated earlier that provides a secure environment for communication between IoT devices in a network. We devised a fog based system model for IoT architecture. The model consists of interconnected fog nodes which are controlled by a fog node controller. The layout is shown in Figure 1. Each fog node is associated with users and IoT devices, as shown in Figure 2. The users act as an interface between the fog node and IoT devices. If one IoT device wants to communicate with another device, it must be authenticated by the user who has access to that IoT device. The fog nodes and users are blockchain-enabled.

The following assumptions are considered for the proposed blockchain-based IoT Architecture.

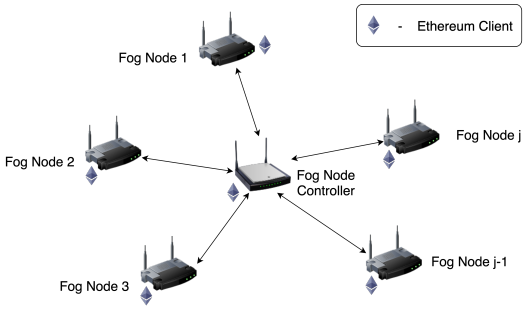

Fig. 1. Proposed System Model (a)

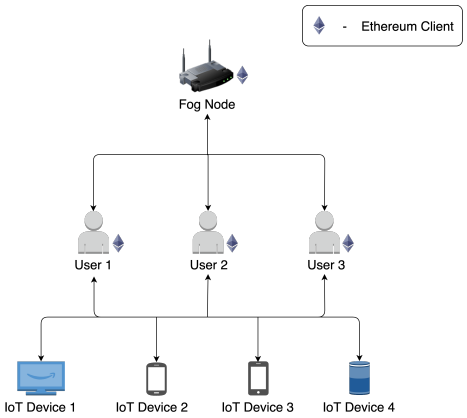

Fig. 2. Proposed System Model (b)

– Every device has a unique identity (ID), which is a combination of the device's MAC address and private key.
– The fog node controller is trusted.

The model is designed in such a way that each and every device in the network except the fog node controller must register with the unique ID. After registering the device, whenever a device wants to communicate with another device, authentication is confirmed before the communication. The proposed model is divided into three phases, namely, initialization phase, authentication phase, and communication phase. Table 1 shows the notations used in the proposed modelß. Details of each phase are described below.

2.1 Initialization Phase

The initialization process consists of the registration of the fog node, user, and IoT device. In fog node registration, each fog node F registers itself with the fog node controller C,which is associated with a unique identity. The fog node

Table 1. Notations used in Transactions

S No.	Notations	Meaning
1	C	Fog Node Controller
2	F	Fog Node
3	U	User
4	D	IoT device
5	X_iId	Unique ID of ith instance of X Node
6	X_iPu	Public Key of ith instance of X Node
7	X_iPr	Private Key of ith instance of X Node

can be registered only when the fog node controller is in the active state. F will send its ID to C encrypted with C's public key. It will be considered as the first transaction named as ϕ_a. This process is described in equation 1.

$$Step\ 1: \quad F \rightarrow C: \quad \phi_{\mathrm{a}} = Enc\left[C_{\mathrm{i}}Pu\left(F_{\mathrm{j}}Id\right)\right] \tag{1}$$

Next, C will decrypt ϕ_a with its private key to obtain F's ID.

$$Step\ 2: \quad F_jId \leftarrow Dec\left[C_iPr\left(\phi_a\right)\right] \tag{2}$$

A certificate $Cert_a$ will be generated by C and sent to F along with F's ID, encrypted with F's public key and this will be considered as next transaction ϕ_b.

$$Step\ 3: \quad C \rightarrow F: \quad \phi_b = Enc\left[F_iPu\left(F_jId, Cert_a\right)\right] \tag{3}$$

F will decrypt ϕ_b using its private key and will obtain ID and certificate. F will verify ϕ_b by comparing that ID with its own ID, if both IDs are same then it will store the certificate $Cert_a$ and fog node will be registered. The fog node registration phase is depicted in Algorithm 1.

$$Step\ 4: \quad F_jId, Cert_a \leftarrow Dec\left[F_iPr\left(\phi_b\right)\right] \tag{4}$$

Algorithm 1: Fog Node Registration

```
begin
if (fog_node_controller.active() = true) then
    if (FNID_Exists(fog_node.id, block_chain) = true) then
        return error()
    end
    else
        FNID_Register(fog_node.id, fog_node_controller.id, block_chain)
        return success()
    end
else
    return error()
end
```

In the User device registration, user U first registers itself to the fog node F. The fog node provides certificate to user, which is used to authenticate itself with other devices attached to it. U will send its ID to F encrypted with F's public key in transaction named as ϕ_c. This process is described in equation 5

$$Step\ 1: \quad U \rightarrow F: \quad \phi_{\mathrm{c}} = Enc\left[F_{\mathrm{k}} Pu\left(U_l Id\right)\right] \tag{5}$$

Next, F will decrypt ϕ_b with its private key to obtain U's ID.

$$Step\ 2: \quad U_l Id \leftarrow Dec\left[F_k Pr\left(\phi_c\right)\right] \tag{6}$$

A certificate $Cert_b$ will be generated by F and sent to U along with U's ID, encrypted with U's public key and will be considered as next transaction ϕ_d.

$$Step\ 3: \quad F \rightarrow U: \quad \phi_d = Enc\left[U_l Pu\left(U_l Id, Cert_b\right)\right] \tag{7}$$

U will decrypt ϕ_d using its private key and will get ID and certificate. U will verify ϕ_d by comparing that ID with its own ID, if both IDs are same then it will store the certificate $Cert_b$ and user will be registered.

$$Step\ 4: \quad U_l Id, Cert_b \leftarrow Dec\left[U_l Pr\left(\phi_d\right)\right] \tag{8}$$

The details of this process are shown in Algorithm 2.

Algorithm 2: User Device Registration

```
begin
if (FNID_Exists(fog_node.id, block_chain) = true) then
    if UserID_Exists(user_device.id, block_chain) = true then
        return error()
    else
        UserID_Register(user_device.id, fog_node.id, block_chain)
        return success()
    end
else
    return error()
end
```

In the IoT device registration, each IoT device sends request to its respective user. Then the user approves the request and sends to blockchain based fog node for registration. After successful registration of IoT device, fog node notifies the user that the device is registered. IoT device D will send its ID to User U encrypted with U's public key in transaction named as ϕ_e. This process is described in equation 9

$$Step\ 1: \quad D \rightarrow U: \quad \phi_e = Enc\left[U_x Pu\left(D_y Id\right)\right] \tag{9}$$

Next, U will decrypt ϕ_e with its private key to obtain D's ID.

$$Step\ 2: \quad D_yId \leftarrow Dec\left[U_xPr\left(\phi_e\right)\right] \tag{10}$$

U will forward D's ID to fog node F by encrypting it with F's public key.

$$Step\ 3: \quad U \rightarrow F: \quad \phi_f = Enc\left[F_zPu\left(D_yId\right)\right] \tag{11}$$

F will decrypt ϕ_e with its private key to obtain D's ID.

$$Step\ 4: \quad D_yId \leftarrow Dec\left[F_zPr\left(\phi_f\right)\right] \tag{12}$$

A certificate $Cert_c$ will be generated by F and will be sent to D along with D's ID, encrypted with D's public key and this will be considered as transaction ϕ_g. The same certificate $Cert_c$ is sent to user U encrypted with U's public key considered as transaction ϕ_h.

$$\begin{aligned} Step\ 5: \quad F \rightarrow D: \quad & \phi_g = Enc\left[D_yPu\left(D_yId, Cert_c\right)\right] \\ F \rightarrow U: \quad & \phi_h = Enc\left[U_xPu\left(D_yId, Cert_c\right)\right] \end{aligned} \tag{13}$$

D and U will decrypt respective ϕ_g and ϕ_h using their private key and will get ID and certificate. Both D and U will verify the D's ID and then only IoT device D will be registered. The details of this process are shown in Algorithm 3.

$$\begin{aligned} Step\ 6: \quad & D_yId, Cert_c \leftarrow Dec\left[D_yPr\left(\phi_g\right)\right] \\ & D_yId, Cert_c \leftarrow Dec\left[U_xPr\left(\phi_h\right)\right] \end{aligned} \tag{14}$$

Algorithm 3: IoT Device Registration

```
begin
if (FNID_Exists(fog_node.id, block_chain) = true) then
    if (UserID_Exists(user_device.id, block_chain) = true) then
        if (IoTID_Exists(iot_device.id, block_chain) = true) then
            return error()
        else
            IoTID_Register(iot_device.id, user_device.id, fog_node.id,
              block_chain)
            return success()
        end
    else
        return error()
    end
else
    return error()
end
```

2.2 Authentication Phase

In this phase, first the fog node authentication takes place, followed by the user device authentication, followed by the IoT device authentication.

In the authentication phase of fog node, the certificate received from the registration process as shown in Algorithm 1, is used to authenticate the fog node. Fog node F initiates the transaction ϕ_i to fog node controller C that is encypted using C's public key. It contains F's ID and the certificate $Cert_a$ recieved in equation 4.

$$Step\ 1: \quad F \rightarrow C: \quad \phi_i = Enc\left[C_i Pu\left(F_j Id, Cert_a\right)\right] \tag{15}$$

Then, ϕ_i is decrypted by C using its private key to obtain certificate $Cert_a$ and F's ID as shown in equation 16.

$$Step\ 2: \quad F_j Id, Cert_a \leftarrow Dec\left[C_i Pr\left(\phi_i\right)\right] \tag{16}$$

Now blockchain will verify by comparing the ID's and certificates from equation 4 with the ID's and certificates from equation 16, if both are same then authentication is valid else not. The details of this process are shown in Algorithm 4.

Algorithm 4: Fog Node Authentication

```
begin
if (fog_node_controller.active() = true) then
    if (FNID_Exists(fog_node.id, block_chain) = true) then
        if (FNID_Auth(fog_node.id, fog_node_controller.id, block_chain) = true)
         then
            return success()
else
    return error()
end
```

In the authentication phase of user, the certificate received from the registration process as shown in Algorithm 2, is used to authenticate the user device. Here user U will initiate the transaction ϕ_j to fog node F that is encypted using F's public key. It contains U's ID and the certificate $Cert_b$ recieved in equation 8.

$$Step\ 1: \quad U \rightarrow F: \quad \phi_j = Enc\left[F_k Pu\left(U_l Id, Cert_b\right)\right] \tag{17}$$

Then, ϕ_i is decrypted by F using its private key to obtain certificate $Cert_b$ and U's ID as shown in equation 18.

$$Step\ 2: \quad U_l Id, Cert_b \leftarrow Dec\left[F_k Pr\left(\phi_j\right)\right] \tag{18}$$

Now blockchain will verify by comparing the ID's and certificates from equation 8 with the ID's and certificates from equation 18, if both are same then authentication is valid else not. The details of this process are shown in Algorithm 5.

Algorithm 5: User Device Authentication

```
begin
if (FNID_Exists(fog_node.id, block_chain) = true) then
    if (UserID_Exists(user_device.id, block_chain) = true) then
        if (UserID_Auth(user_device.id, fog_node.id, block_chain) = true) then
            return success()
else
    return error()
end
```

In the authentication phase of IoT device, the certificate received from the registration process as shown in Algorithm 3, is used to authenticate the IoT. Here user D will initiate the transaction ϕ_k to user U that is encypted using U's public key. It contains D's ID and the certificate $Cert_c$ recieved in equation 15.

$$Step\ 1: \quad D \rightarrow U: \quad \phi_k = Enc\left[U_x Pu\left(D_y Id, Cert_c\right)\right] \tag{19}$$

Then, ϕ_k is decrypted by U using its private key to obtain certificate $Cert_c$ and D's ID as shown in equation 19.

$$Step\ 2: \quad D_y Id, Cert_c \leftarrow Dec\left[U_x Pr\left(\phi_k\right)\right] \tag{20}$$

Now blockchain will verify by comparing the ID's and certificates from equation 14 with the ID's and certificates from equation 20, if both are same then authentication is valid else not. The details of this process are shown in Algorithm 6.

Algorithm 6: IoT Device Authentication

```
begin
if (FNID_Exists(fog_node.id, block_chain) = true) then
    if (UserID_Exists(user_device.id, block_chain) = true then
        if (IoTDID_Exists(iot_device.id, block_chain) = true) then
            if (IoTDID_Auth(iot_device.id, fog_node.id, block_chain) = true)
              then
                return success()
else
    return error()
end
```

2.3 Communication Phase

In this phase, the communication rule for each device is established. If one IoT device initiates for communication, then the corresponding fog node checks if it is a legitimate device or not. Also, it verifies if the request is malicious or not. After that, it checks for the device to which it wants to communicate. If all the conditions are satisfied, then only a secure communication is established.

Algorithm 7: Device Communication

```
begin
if (FNID_Exists(fog_node.id, block_chain) = true) then
    if (IoTDID_Exists(iot_device.id, block_chain) = true) then
        if (IoT_Device_Authentication() = true) then
            if Request_Malicious() = false) then
                if (FNID_Exists(fog_node.id, block_chain) = true) then
                    if (IoTDID_Exists(iot_device.id, block_chain) = true) then
                        Secure Communication Established
else
    return error()
end
```

3 Experiment and Result Discussion

This section presents an overview of experiments that have been performed to validate the proposed model starting with experimental setup to the results obtained and the discussion on the obtained results. Fog node controller, Fog Node and User are connected to Blockchain-enabled IoT network. Blockchain simulation is achieved with the help of True Ganache framework and Meta-Mask connected with web3 interface. Communication between the nodes has been done using JsonRPC library.

3.1 Time & Power Consumption

In this section, we compared time and power consumption for registration of node in millisecond and to send data message in milliwatt with existing schemes. We have presented results that are based on 100 experiments. Comparative analysis of M.T. Hammi et al. [4] with our proposed model for time consumption is shown in Table 2 and for power consumption is shown in Table 3.

3.2 Security Analysis

In this section, the security analysis of the proposed architecture is done, which is illustrated as follows.

Table 2. Time comparison of different operations

Approach	Time needed			
	for registration of node (ms)		to send data message (ms)	
	Average	SD	Average	SD
Bubbles of Trust [4]	1.56	0.13	0.04	0.001
Proposed Model	1.44	0.11	0.04	0.001

Table 3. Power comparison of different operations

Approach	Power needed			
	for registration of node (mW)		to send data message (mW)	
	Average	SD	Average	SD
Bubbles of Trust [4]	9.76	2.04	3.35	0.87
Proposed Model	8.00	2.02	3.10	0.85

- **Data Integrity:** In the proposed architecture, all transactions are stored in blockchains. Each block in the blockchain consists of many parameters, including the previous block hash. If someone tries to change transaction data, they have to change each and every block present in the blockchain, which is very difficult as each client has a copy of the blockchain. So, data integrity is maintained.
- **Non-Repudiation :** After registration of any node, whenever a transaction happens, the sender will use its private key to encrypt the message and sends it to the receiver. Thereby, in every transaction, the sender's private key will be used. Hence, the sender cannot refute its initialization of the message as it can be verified by its public key.
- **Mutual Authentication:** Every time a device sends a request, it will be verified by the blockchain-enabled fog node and every time a fog node sends a response, it will be verified by the device. For example, in Algorithm 1, the fog node sends its ID as equation 1. The fog node controller verifies it by decrypting the transaction using its private key shown in equation 2. This ID is searched in the blockchain whether it already exists or not. After that, the fog node controller sends the certificate to the fog node. This transaction is verified by the fog node with its private key, as shown in equation 3. It is further verified by the public key of the fog node controller as shown in equation 4. Hence mutual authentication is achieved.
- **Availaibility:** Users are ethereum clients, which means they can access thier respective IoT devices from anywhere in real-time. Only the authentic and legitimate users can access IoT devices through the nearest fog node, thereby, are available to authentic users.
- **Spoofing Attack:** When the attacker in the network pretends to be some other entity, it is known as spoofing attack. In proposed model, every device has a unique ID i.e. C_iId, F_iId, U_iId and D_iId, which are stored in blockchains. Since every device's private key is unknown to every other de-

vice, it is very difficult to obtain the unique ID; hence the proposed model is resistant to Spoofing attack.

- **Man-In-The-Middle Attack (MIMA):** In the authentication phase, every device uses its own public and private key. As long as they do not share these keys with third parties, MIMA is not possible. The keys used in proposed model are C_iPu, C_iPr, F_iPu, F_iPr, U_iPu, U_iPr, D_iPu and D_iPr.
- **Denial of Service:** In the proposed model, the request is checked if it is malicious or not using $Request\_Malicious()$. If the same request with the same parameters is sent many times, the model will discard those requests to prevent a DOS attack. Hence all recourses are available to authentic participants of the system.
- **Message Replay Attack:** Since all the transactions are done by the devices with the help of their unique IDs, the same transaction cannot happen back to back, else the request will be considered malicious. The system will reject the request. Hence replay attack is prevented.
- **Fog Node Impersonation Attack:** It may happen that someone poses as a fog node to access the details of user devices and IoT devices. Every transaction related to the fog node involves F_iPu or F_iPr. The F_iPr is not available publicly; thereby, no one can decrypt the transaction except the original fog node. The public key of the fog node is available in the network. One can use F_iPu for the verification of the transaction or send some message to the fog node by first encrypting with their private key. Hence, it is not possible to impersonate a user or IoT device in the proposed architecture.

3.3 Performance Evaluation

In this section, we compared the proposed architecture with some of the existing schemes. The performance evaluation is depicted in Table 4. M. T. Hammi [4] and U. Khalid [10] models do not check if the request is being malicious or not, which makes their model vulnerable to intruder attack as well as social engineering. Our proposed model overcomes these shortcomings.

Table 4. Performance Evaluation with Existing Models

Author(s) & **Year(s)**	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**
M.T. Hammi *et al.* 2018 [4]	Yes	Yes	Yes	Yes	Yes	No	No	Yes	No
D. C. Nguyen*et al.* 2019 [6]	Yes	Yes	No	Yes	No	No	No	No	No
U. Khalid*et al.* 2020 [10]	Yes	Yes	Yes	No	Yes	No	No	Yes	No
M. Tahir*et al.* 2020 [12]	Yes	Yes	Yes	Yes	Yes	No	No	Yes	No
Proposed Model	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Note: S1- Data Integrity, S2- Non-Repudiation, S3-Mutual Authentication, S4- Availaibility, S5- Spoofing Attack, S6- Man-In-The-Middle Attack (MIMA), S7- Denial of Service, S8- Message Replay Attack, S9- Fog Node Impersonation Attack.

4 Conclusion

IoT is a promising area of research and IoT Devices have become a part of day to day life. Combining blockchain with IoT gives a lot of promising work to make IoT more secure. In proposed model IoT devices takes less time and consumes less power for communicating as compared to existing scheme which can be used in real life scenarios. The proposed scheme can withstand passive and active attacks, such as spoofing attack, man-in-the-middle attack, and fog node impersonation attack. In order to make the proposed model light weight, the future work would focus on certificate less authentication mechanism.

References

1. Atzori, L., Iera, A., Morabito, G.: The internet of things: A survey. Computer Networks **54**(15) (2010) 2787–2805
2. Bahga, A., Madisetti, V.K.: Blockchain platform for industrial internet of things. Journal of Software Engineering and Applications **9**(10) (2016) 533–546
3. Dorri, A., Kanhere, S.S., Jurdak, R., Gauravaram, P.: Blockchain for iot security and privacy: The case study of a smart home. In: 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), IEEE (2017) 618–623
4. Hammi, M.T., Hammi, B., Bellot, P., Serhrouchni, A.: Bubbles of trust: A decentralized blockchain-based authentication system for iot. Computers & Security **78** (2018) 126–142
5. Kamran, M., Khan, H.U., Nisar, W., Farooq, M., Rehman, S.U.: Blockchain and internet of things: A bibliometric study. Computers & Electrical Engineering **81** (2020) 1–12
6. Khalid, U., Asim, M., Baker, T., Hung, P.C., Tariq, M.A., Rafferty, L.: A decentralized lightweight blockchain-based authentication mechanism for iot systems. Cluster Computing **23**(3) (2020) 1–21
7. Khan, M.A., Salah, K.: Iot security: Review, blockchain solutions, and open challenges. Future Generation Computer Systems **82** (2018) 395–411
8. Lin, J., Yu, W., Zhang, N., Yang, X., Zhang, H., Zhao, W.: A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications. IEEE Internet of Things Journal **4**(5) (2017) 1125–1142
9. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system. Technical report (2019)
10. Nguyen, D.C., Pathirana, P.N., Ding, M., Seneviratne, A.: Blockchain for secure ehrs sharing of mobile cloud based e-health systems. IEEE access **7** (2019) 66792–66806
11. Ray, P.: A survey on internet of things architectures. Journal of King Saud University - Computer and Information Sciences **30**(3) (2018) 291–319
12. Tahir, M., Sardaraz, M., Muhammad, S., Saud Khan, M.: A lightweight authentication and authorization framework for blockchain-enabled iot network in health-informatics. Sustainability **12**(17) (2020) 1–23