# bisAUTH: A Blockchain-Inspired Secure Authentication Protocol for IoT Nodes

Cherif Diallo(✉)

Laboratoire Algèbre, Cryptographie, Codes et Applications (LACCA),
UFR Sciences Appliquées et de Technologies (UFR SAT),
Université Gaston Berger, 234 Saint-Louis, Senegal
cherif.diallo@ugb.edu.sn

**Abstract.** Some existing methods for authentication of IoT nodes are based on blockchain technology which has limitations that must be taken into account. In this paper, our contribution consists in proposing a new authentication protocol, named bisAUTH, for IoT objects inspired on certain characteristics of the blockchain technology. Our main objective is to propose a mechanism allowing neighboring IoT nodes of a network to authenticate in a decentralized and secure way. Evaluation of our proposed protocol against several criteria, its resistance to various attacks and its comparison with recent protocols, show that it brings significant improvements compared to the existing ones.

**Keywords:** IoT · Blockchain · Authentication · Lightweight security

## 1 Introduction

The advent of connected objects is one of the major innovations of the Internet. It is a technology that brings together a set of physical objects integrated with sensors, software and other technologies with the aim of collecting information and allowing internet communication between devices, thus offering a multitude of services. The IoT is therefore developing rapidly. However, it presents many challenges that arise from its inherent characteristics. It presents a heterogeneous, uncontrolled environment made up of vulnerable objects with limited resources in terms of computing capacity and storage space. Security-related issues therefore represent major challenges that require effective solutions for the development and deployment of certain IoT applications. In this context, authentication is an essential aspect to guarantee good security in IoT environments, where objects process and exchange data without human intervention. Therefore, it is essential to develop secured authentication mechanisms to avoid possible threats. Many works have proposed several types of object authentication mechanisms suitable for the IoT environment. Most of these works are based on centralized authentication whereby all devices must contact a single entity. When this one is compromised, this creates a major inconvenience. Such issue can be overcome by using blockchain technology which allows the trusted


J. Lin and Q. Tang (Eds.): AC3 2022, LNICST 448, pp. 108–119, 2022.
https://doi.org/10.1007/978-3-031-17081-2_7

third party to be replaced with a transparent and untampered block of records available through a distributed form, so that trust is moved from a single entity to decentralized nodes. Thus, the object of our work is therefore to propose an efficient and secure mutual authentication protocol for IoT objects inspired on the blockchain technology to guarantee a high level of resistance to attacks.

## 2 Related Works

For most of IoT use cases, securing authentication between nodes is critical. We give here a brief review of some works that offer blockchain-based authentication methods in IoT environments. These schemes have their own specificity such that using a one-way hash chain for authentication [1], creating virtual trust bubbles [2], proposing a decentralized web authentication system [3], integrating the constraints of WSN sensor networks [4] or using Fog Nodes to allow devices to be relieved of some heavy lifting [5]. In terms of performance, some previous works try to reduce computational load, power consumption and latency [1,2,4]. Then [5] tries to avoid congestion. But these methods have their shortcomings. Thus [3] suffers from a slow user account creation process. [3] and [4] do not offer mutual authentication. Moreover [4] does not guarantee integrity and is not compatible with heterogeneous environments; [6] requires lot of messages sending and verification, which can quickly flood all the communication mediums; [7] and [8] need lot of memory, and lead to greater energy consumption, but [8] uses timestamping which allows resistance against replay attack. Finally, the Table 1 shows an assessment of these protocols in relation to their resistance to attacks. As we can see, each of them is vulnerable to at least two or more attacks. Considering all these weaknesses, we propose, in the following, a protocol with its lightweight version to address the shortcomings of existing solutions.

**Table 1.** Resistance to attacks of some protocols

| Types of attacks | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] |
|---|---|---|---|---|---|---|---|---|
| DoS/DDoS attack | Yes | No | Yes | Yes | No | No | Yes | No |
| Sybil attack | No | Yes | No | No | Yes | No | No | Yes |
| Impersonation attack | No | Yes | Yes | No | Yes | Yes | No | No |
| Man in the middle | Yes | No | No | Yes | No | Yes | Yes | No |
| Replay attack | Yes | Yes | Yes | No | Yes | Yes | No | Yes |
| Insertion of malicious nodes | Yes | Yes | Yes | No | No | Yes | No | Yes |
| Brute force attack | Yes | Yes | Yes | Yes | Yes | Yes | No | No |

## 3 bisAUTH: A New Blockchain-Inspired Secure Authentication Protocol for IoT Nodes

As in the blockchain, our approach guarantees trust between the different nodes of the network in a consensual way. It is decentralized, and is characterized by a

mechanism for distributing blocks of secrets to the different nodes. This allows them to authenticate new objects wishing to join the network. The solution also relies on the use of asymmetric cryptography. In addition, it has two versions, one of which offers a lightweight authentication function with a very high level of security.

### 3.1 Main Components of the bisAUTH Protocol

*Server.* The protocol is mainly characterized by a server (Fig. 1), hosted in the cloud, which contains in particular a set of blocks of secrets containing information that is only accessible by authorized objects. This server does not intervene in the authentication process, but it is used in the initialization phase of the objects before the deployment of the network. It contains other information such as the fingerprints or hashes of the blocks of secrets, those of the nodes, but also other useful information for an administrator such as logs and timestamping system. In addition, this server will have to store the public keys of the nodes to which it will assign blocks. It has its own public key which allows it to distribute blocks of secrets to different nodes. In the initialization phase, the server will encrypt the secret message contained in the block with a secret key which will be distributed to the nodes.

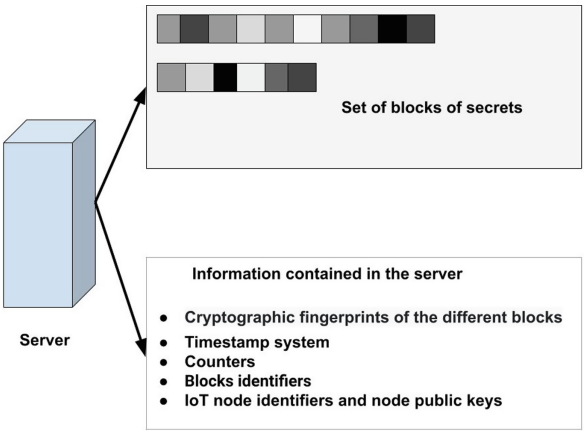


**Fig. 1.** Server for managing and distributing blocks of secrets.

*Secret Blocks.* Here, the secret blocks form an unlinked list of blocks independent of each other. Each secret block has its own fingerprint or hash through the use of a hash function. This ensures the integrity of a message and its authenticity. In our protocol, the use of hash function to generate a hash will allow nodes that have blocks in common to be able to authenticate in two different ways:

– In the lightweight version, the nodes will rely solely on the fingerprints of their blocks to be able to validate an authentication.
– In the full version, in addition to the fingerprints of their blocks, the nodes will also verify the secret message with the secret key distributed during the initialization phase in order to validate an authentication.

### 3.2 Main Phases of the bisAUTH Protocol

**The bisAUTH Initialization Phase.** Our approach begins with an initialization phase of the different IoT nodes. This phase is common to both lightweight and full versions of the protocol. The different blocks of secrets created in the server are distributed to the nodes, following this process:

1. A node requests the server public key.
2. The server sends its public key to the node.
3. Then, the node sends an encrypted message with the public key of the server containing its identifier and its public key.
4. The server calculates and assigns a list of secret blocks to the node.

The secret blocks are assign to the objects in such a way that to guarantee a certain rate of similarity between objects (Fig. 2). Indeed, by considering any two objects, the probability that these two objects have blocks in common must be greater than or equal to the fixed rate of similarity ($Prob_{SimBlocks}$). Each node will be assigned several blocks. The nodes do not necessarily have the same number of blocks. The node will first send a request to the server to ask for its public key. This node will then send a registration message containing its identifier and its public key encrypted with the public key of the server. Then the server will send the secret blocks to the nodes through an encrypted message with the public key of the node. When a node receive its list of secret blocks, it will also store the fingerprints of each block. At each allocation of secret blocks, the nodes will generate a private/public key pair to encrypt and decrypt messages. The nodes are very limited in terms of storage, we use elliptic curves public keys. The Fig. 3 shows the flowchart of the initialization phase.

**The bisAUTH Authentication Phase.** Our approach aims to guarantee that IoT legitimate nodes will be able to authenticate each other, and not allow malicious nodes to enter the network. Firstly, a node which would like to be authenticated, has to request the public keys of its neighbors who will authenticate it later. Then, the node encrypts with these public keys a message for its authentication request. This message will contain its identifier, the identifiers of secret blocks with their fingerprints. After this step, the different neighboring nodes will then decrypt the received message with their own private keys and calculate the hash function to check if the message has not been modified by an

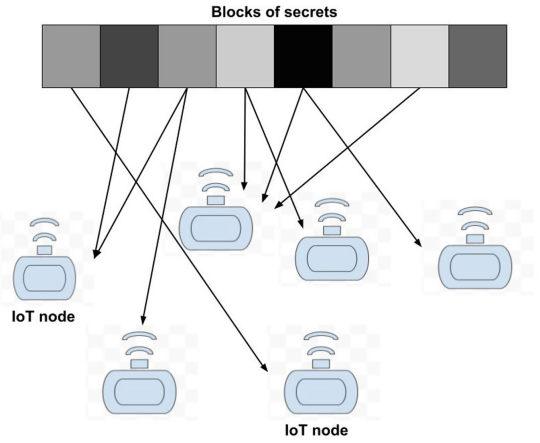


**Fig. 2.** Allocation of secret blocks to different IoT nodes.

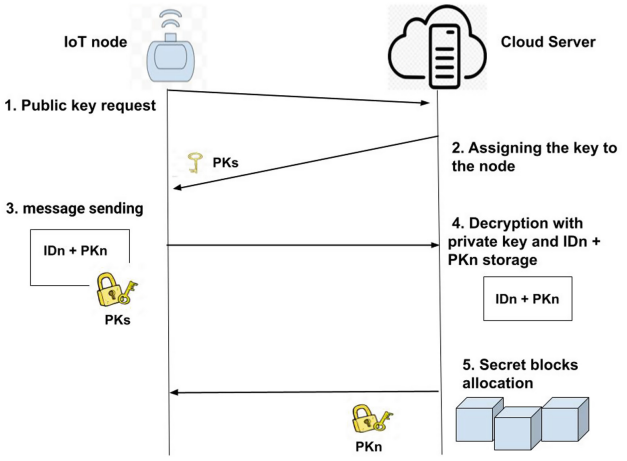


**Fig. 3.** Flowchart of the bisAUTH protocol initialization phase.

attacker node. If this information is correct, neighboring nodes will then see if they have one or more block fingerprints corresponding to the fingerprints they hold in their keyrings. Therefore, if a neighboring node recognizes one or more fingerprints, it will send a notification with the public key of the node to indicate to the node that it recognizes its block. In the lightweight version of the protocol, the authentication process stops at this step when the percentage of neighbors

considered for consensus is reached. While, in its full version, the authentication process continues with other steps. Once a neighboring node has recognized and verified the information (hash and secret block), it has therefore been able to authenticate the node. Then, it sends a confirmation message encrypted with the public key of the requesting node (Fig. 4). The confirmation message contains its identifier as well as the fingerprint of the block, its public key and a hash (Fig. 4). The authentication phase will be complete only when the percentage of neighbors considered for consensus is reached, i.e. a certain number of nodes possessing blocks in common with the requesting node. In the event that the neighboring nodes fail to recognize the secret blocks, they will relay the request to their neighbors which in turn check whether they recognize one or more of these blocks. This process is repeated until authentication succeeds, or until all neighbors are reached. The figure (Fig. 5) shows the flowchart of the of the lightweight version of the protocol, whereas the figure (Fig. 6) gives the flowchart of the full version. Finally, the Algorithm 1 shows the overall process of the authentication phase for any requesting node.

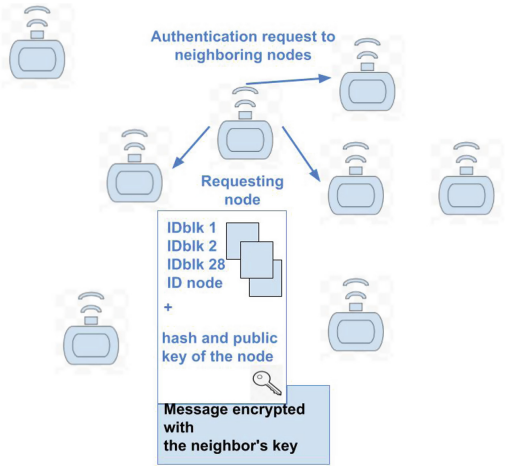


**Fig. 4.** Authentication request to neighbors.

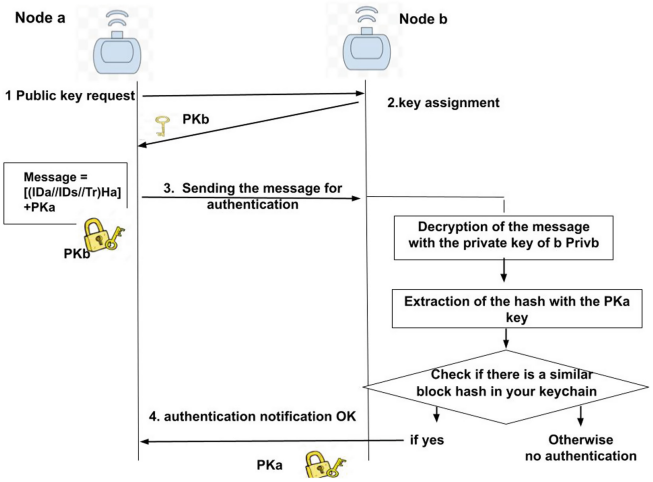


**Fig. 5.** Flowchart of the lightweight version of bisAUTH protocol.

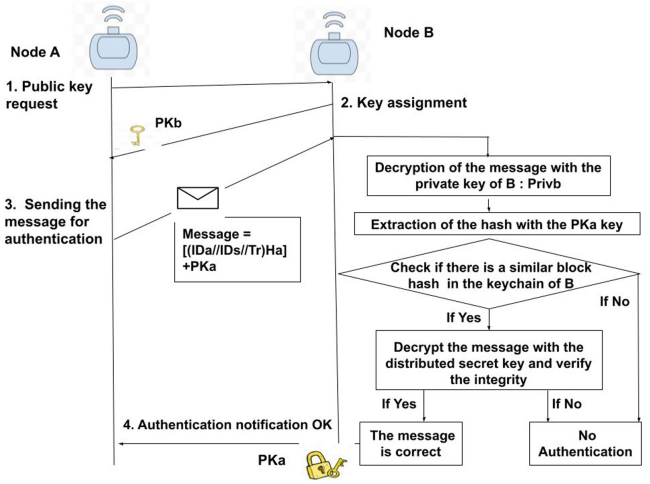


**Fig. 6.** Flowchart of the full version of bisAUTH protocol.

**Algorithm 1.** The bisAUTH Authentication phase for any requesting node

$Consensus \leftarrow Percent\_of\_neighbors\_for\_consensus$
$N \leftarrow Number\_of\_neighbors\_of\_the\_node$
$M \leftarrow \lceil \frac{N*Consensus}{100} \rceil$
$Sb \leftarrow Number\_of\_secret\_blocks\_of\_the\_node$
$Lsb \leftarrow Secret\_blocks\_of\_the\_node$ ▷ Secret blocks list of the requesting node
$Na \leftarrow 1$ ▷ Number of authentication attempts
$Auth \leftarrow False$
**while** $((Na \leq Sb)$ and $(Auth \neq True))$ **do**
  diffusion of the secret block $Lsb[Na]$
  $Cc \leftarrow 0$ ▷ Consensus counter
  **for** $i = 1$ to $N$ **do**
    neighbor[i] checks the received secret block
    **if** the received secret block is validated by neighbor[i] **then**
      $Cc \leftarrow Cc + 1$
    **end if**
  **end for**
  **if** $Cc \geq M$ **then**
    $Auth \leftarrow True$
  **end if**
  $Na \leftarrow Na + 1$
**end while**
**if** $Auth = True$ **then**
  Return successful authentication message
**else**
  $EndProc \leftarrow False$
  **while** $EndProc \neq True$ **do**
    $i \leftarrow 1$
    **while** $i \leq N$ **do**
      neighbor[i] relays the authentication request using the $Lsb$ list
      **if** Successful authentication **then**
        $Auth \leftarrow True$
        $i \leftarrow N + 1$
        Return successful authentication message
        $EndProc \leftarrow True$
      **else**
        $i \leftarrow i + 1$
      **end if**
    **end while**
    **if** $Auth \neq True$ **then**
      Return failed authentication message
      $EndProc \leftarrow True$
    **end if**
  **end while**
**end if**

**The bisAUTH Maintenance Phase.** It corresponds to the management of the cryptography keys which have a lifetime each. The lifetime begins when the authentication phase begins and ends when the node has been successfully authenticated by a number of nodes (percent neighbors for consensus).

### 3.3 bisAUTH Protocol Assessment

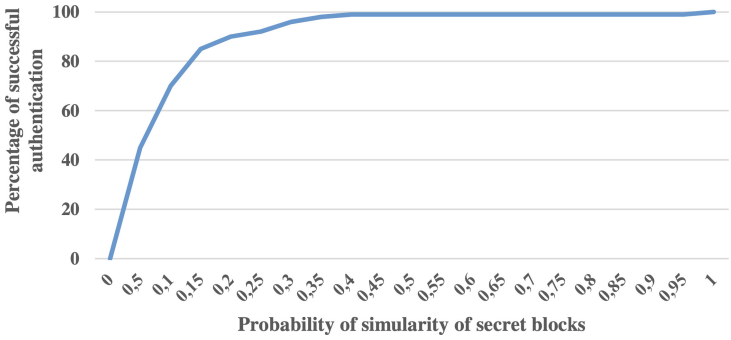


**Fig. 7.** Percentage of successful authentication of the node from its first request to neighbors according to the probability of similarity of secret blocks during the initialization phase. Here, the percent of neighbors for consensus is set to $\tau = 20\%$.

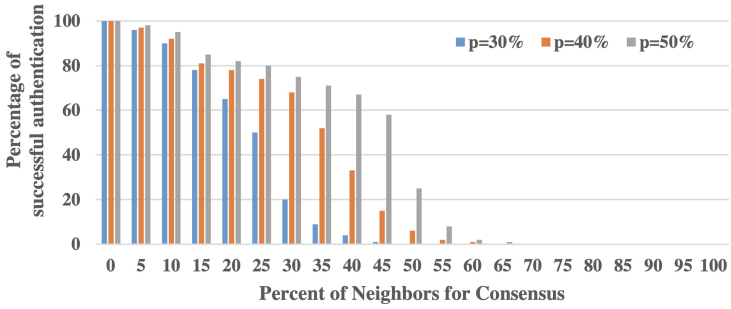


**Fig. 8.** Percentage of successful authentication of a node its immediate neighbors according to the percentage of neighbors for consensus when the probability of similarity of used secret blocks ($Prob_{SimBlocks}$) is p = 30%, p = 40% or p = 50%.

**Simulation Results.** The Fig. 7 plots the percentage of success from the first authentication request. This indicator is important because if many authentication requests are relayed by neighbors, this will increase traffic in the network, which will be harmful in terms of performance. The result shows that the curve

converges quite quickly, which means that for our case, most of the requests are not relayed because the authentication is successfully performed by the immediate neighbors of the requesting node. As for the Fig. 8, it plots the percentage of success from the first authentication request as a function of the rate of similarity fixed for the secret blocks allocation process and the percentage of neighbors for the consensus. It is also an interesting result which shows that by increasing the initial similarity rate, we guarantee a good rate of authentication by the immediate neighbors. On the other hand, when the percentage of neighbors for consensus increases, the authentication conditions are toughened, and consequently, the success rate decreases. Which necessarily leads to the fact that more requests will be relayed by the neighbors who have not succeeded in authenticating the requesting node.

**Analysis of bisAUTH Against Authentication Criteria.** The bisAUTH protocol offers several features related to certain security criteria: privacy, integrity, mutual authentication, robustness and scalability. In terms of performance, the lightweight version guarantees a very good level of performance, while the full version could be a little wiped out when the average network density becomes too great. Moreover, bisAUTH could be considered as a Token-based authentication protocol since secret blocks can be considered tokens if embedded (Table 2). Even if it is not an authentication protocol based on physical hardware, secrets blocks could be burned in an inherently hardware-bound way. In addition, bisAUTH takes advantage of the performance offered by ECC cryptography which is considered as a natural modern successor of the RSA cryptosystem, because ECC uses smaller keys and signatures than RSA for the same level of security and provides very fast key generation, fast key agreement and fast signatures. Finally, [9] is resistant against many attacks, however it does not guarantee privacy whereas bisAUTH offers this important feature.

**Table 2.** Analysis of bisAUTH protocol against authentication criteria.

| Criteria | bisAUTH features |
|---|---|
| Authentication factor | Shared secrets |
| Authentication procedure | Two-way or mutual authentication |
| Authentication based on physical hardware | Not used here, but possible |
| Authentication architecture | Decentralized |
| Integrity, privacy | Yes |
| Low computational and memory load | Yes |
| Scalability and efficiency | Yes |
| Cryptography technique | Use of ECC-type asymmetric keys |
| Lightweight version | Yes |

**Table 3.** Analysis of bisAUTH in relation to its resistance to attacks

| Types of attacks | Initialization phase | Authentication phase |
|---|---|---|
| DoS/DDoS attack | No | Yes |
| Sybil attack | Yes | Yes |
| Impersonation attack | Yes | Yes |
| Man in the middle | Yes | Yes |
| Replay attack | Yes | Yes |
| Insertion of malicious nodes | Yes | Yes |
| Brute force attack | Yes | Yes |

**Analysis of bisAUTH in Relation to Its Resistance to Attacks.** The DoS/DDoS attack could occur during the initialization phase and the authentication phase. During the initialization phase, the server containing the secret blocks could be affected by this type of attack, preventing it from being able to carry out the collection of information from a node as well as the allocation process of the secret blocks to legitimate nodes. At the level of the authentication phase, this type of attack is not possible because requests from malicious nodes are ignored after a few unsuccessful authentication attempts. As for the sybil attack, an attacker cannot use fake identities to carry out this attack because each object, in our solution, has a unique identity that is intrinsically tied to its private/public key pair. Likewise, an attacker will be unable to impersonate a network node because he must have the node's private key in his possession. The Man in the middle attack cannot occur in our protocol because an attacker will not be able to put himself in the middle of a communication between two nodes and intercept messages coming from one of these nodes. He can generate his own private/public key pair but cannot hold the node's private key to decrypt the message sent to him. Indeed, this message will be encrypted with the public key of the recipient. So, the attacker must hold the private key of the recipient to read the message. For the same reason, an attacker will not be able to perform a replay attack because he must hold the node keys.

During the authentication phase, it is possible that malicious nodes try to be authenticated by the legitimate nodes of the network (Table 3). These malicious nodes which have not been initialized by the server will attempt to generate fingerprints in order to be authenticated and these fingerprints must match the fingerprints contained in the keyring of the legitimate nodes. A malicious node could generate a good number of fingerprints before being finally authenticated in the network. The probability that a node could be authenticated must depend on a certain number of parameters such as: the average number of secret blocks that exist in the vicinity of the node, the size of the hash ($n$), the average density of the network but also the percent of neighbors for consensus. The probability that a malicious node finds a legitimate node fingerprint is equal to:

$$P = \frac{Nb\_blocks}{2^n} * Average\_density * Percent\_of\_Neighbors\_for\_Concensus$$

This indicates that the probability that a malicious node integrates the network is very low. So, we can conclude that the protocol offers a very high level of security against this type of threat. To reinforce its resistance, one could use larger fingerprints ($n$). As for the brute force attack, it could happen when a malicious node manages to get all possible fingerprints. Therefore it would send a lot of requests to achieve this goal. The solution to avoid this attack is that, after a few unsuccessful attempts, neighboring nodes will blacklist this malicious node and simply ignore its future authentication requests.

## 4 Conclusion

In [9], the protocol did not guarantee confidentiality. In this paper, we have proposed a new authentication mechanism for IoT objects inspired by blockchain technology. We presented our new protocol and its lightweight version which ensure privacy. Finally, we assess it against several security challenges and attacks. Clearly, this protocol meets several criteria that an authentication method must guarantee to properly secure IoT nodes. Moreover, it brings important improvements, and it is better than the existing ones on various aspects that we have evaluated it.

## References

1. Panda, S.S., Jena, D., Mohanta, B.K., Ramasubbareddy, S., Daneshmand, M., Gandomi, A.H.: Authentication and key management in distributed IoT using blockchain technology. IEEE Internet Things J. **8**(16), 12947–12954 (2021)
2. Hammi, M.T., Hammi, B., Bellot, P., Serhrouchni, A.: Bubbles of trust: a decentralized blockchain-based authentication system for IoT. Comput. Secur. J. **78**, 126–142 (2018)
3. Mohanta, B.K., Sahoo, A., Patel, S., Panda, S.S., Jena, D., Gountia, D.: DecAuth: decentralized authentication scheme for IoT device using Ethereum blockchain. In: Proceedings of 2019 IEEE Region 10 Conference (TENCON 2019) (2019)
4. Moinet, A., Darties, B., Baril, J.-L.: Blockchain based trust & authentication for decentralized sensor networks. IEEE Secur. Priv. Issue Blockchain (2017)
5. Abdalah, A.N., Mohamed, A., Hefny, H.A.: Proposed authentication protocol for IoT using blockchain and fog nodes. Int. J. Adv. Comput. Sci. Appl. **11**(4), 710–716 (2020)
6. Aman, M.N., Chua, K.C., Sikdar, B.: A lightweight mutual authentication protocol for IoT systems. In: GLOBECOM 2017, pp. 1–6. IEEE (2017)
7. Li, D., Peng, W., Deng, W., Gai, F.: A blockchain-based authentication and security mechanism for IoT. In: 2018 27th International Conference on Computer Communication and Networks (ICCCN), pp. 1–6. IEEE (2018)
8. Alizai, Z.A., Tareen, N.F., Jadoon, I.: Improved IoT device authentication scheme using device capability and digital signatures. In: 2018 International Conference on Applied and Engineering Mathematics (ICAEM), pp. 1–5. IEEE (2018)
9. Diedhiou, O.N., Diallo, C.: An IoT mutual authentication scheme based on PUF and blockchain. In: Proceeding of the IEEE International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, USA, pp. 1034–1040 (2020)