Proof-of-Authentication for Scalable Blockchain in Resource-Constrained Distributed Systems

Deepak Puthal*, Saraju P. Mohanty†, Priyadarsi Nanda*, Elias Kougianos‡, and Gautam Das§
* Faculty of Engineering and Information Technologies, University of Technology Sydney, Australia
Email: {Deepak.Puthal, Priyadarsi.Nanda}@uts.edu.au
† Department of Computer Science and Engineering, University of North Texas, USA
Email: {Saraju.Mohanty}@unt.edu
‡ Department of Engineering Technology, University of North Texas, USA
Email: {Elias.Kougianos}@unt.edu
§ Department of Computer Science and Engineering, The University of Texas at Arlington, USA
Email: gdas@uta.edu

***Abstract*—Resource -constrained distributed systems such as the Internet of Things (IoT), edge computing and fog computing are deployed for real-time monitoring and evaluation. Current security solutions are problematic when there is a centralized controlling entity. The blockchain provides decentralized security architectures using proof-of-work (PoW). Proof-of-work is an expensive process for IoT and edge computing due to the deployment of resource-constrained devices. This paper presents a novel consensus algorithm called Proof-of-Authentication (PoAh) to replace Proof-of-Work and introduce authentication in such environments to make the blockchain application-specific. This paper implemented the Proof-of-Authentication system to evaluate its sustainability and applicability for the IoT and edge computing. The evaluation process is conducted in both simulation and real-time testbeds to evaluate performance. Finally, the process of Proof-of-Authentication and its integration with blockchain in resource-constrained distributed systems is discussed. Our proposed PoAh, while running in limited computer resources (e.g. single-board computing devices like the Raspberry Pi) has a latency in the order of 3 secs.**

***Index Terms*—Blockchain, Consensus Algorithm, Proof-of-Work, Proof-of-Authentication, Internet of Things (IoT), Resource constrained distributed systems**

I. Introduction

The Internet of Things (IoT) is a collection of "Things" that can communicate and share information between different applications to achieve certain goals. The IoT follows different communications management than the TCP/IP stack, where things follow an Identity Management Protocol to become identified devices during data collection [1]. The IoT operates at the lower lever of data collection and processing as shown in Figure 1. The IoT combines the sensing layer and middleware for efficient data collection and transmission, while data is also transmitted to a higher layer (Edge Layer) for near real-time data processing [2]. The edge layer includes a large number of edge datacenters (EDC) and EDCs are distributed in nature without central dependencies. This edge deployment makes the IoT deployment sustainable by processing emergency data in the edge layer [3].

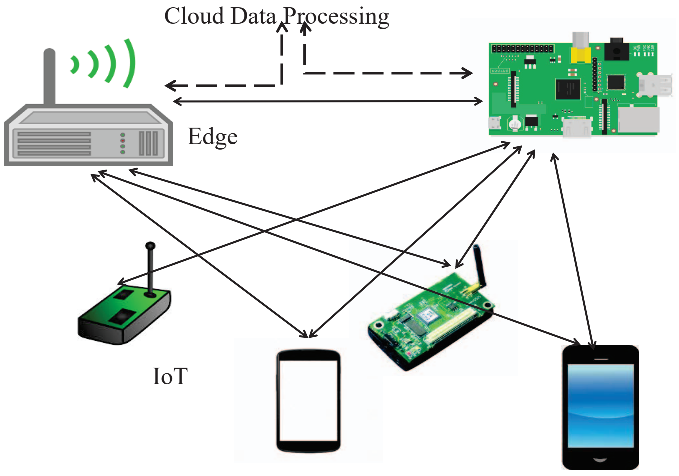


Figure 1. Multi-layer architecture of resource-constrained distributed systems.

The combination of IoT and EDCs forms a distributed environment with resource-constrained devices. Security becomes a key requirement for these mission critical environments. Several security solutions involving cryptography exist for IoT and similar applications [4][5]. Cryptographic solutions are broadly divided into two types: asymmetric cryptography and symmetric cryptography. Symmetric cryptography is approximately 1000 times faster than asymmetric cryptography, and therefore symmetric cryptography is widely accepted for IoT infrastructure [6]. The IoT is a large part of resource-constrained distributed systems, so it also adopts symmetric cryptosystem for encryption and decryption. In either case, there is always a central entity to initialize or process these security mechanisms, i.e. a Key Distribution Center (KDC) for symmetric cryptography and a Certificate Authority (CA) for asymmetric cryptography. As a result, a single-point failure may lead to a compromise of the whole system. There is always demand of decentralized security solutions for distributed systems, and the blockchain is presently a widely accepted decentralized security solution.

In a change in outlook, the blockchain disposes of the requirement for any central entity between various users executing data and financial transmissions by utilizing an immutable, simple and decentralized public ledger. This public ledger is distributed in nature, where individual network participants maintain this database to keep track of network transactions. A major issue with the original blockchain is the cost of proof-of-work (PoW). PoW cannot be directly implemented into resource-constrained distributed systems and this paper proposes a new consensus algorithm named Proof-of-Authentication (PoAh) to make the blockchain suitable for resource-constrained distributed systems.

The **novel contributions of the current paper** that advance blockchain technology are:

- Proof-of-Authentication (PoAh) is proposed as a new consensus algorithm for lightweight blockchain.
- The new consensus algorithm is validated for resource-constrained distributed systems.
- Finally, Proof-of-Authentication is evaluated in both simulation and testbed environments.

The rest of the paper is organized in the following manner. Section II presents background and motivation of this work. The proposed novel Proof-of-Authentication (PoAh) algorithm is discussed in Section III. Experimental results are presented in Section IV. Conclusions and future directions are summarized in Section V.

II. Preliminaries and Problem Motivation

In 2008, the possibility of the blockchain was first presented by a researcher who implemented the advanced digital cryptocurrency known as bitcoin, where the blockchain is an essential piece of its framework [7]. Various cryptocurrencies with cutting edge advances have appeared since then. For example, Ethereum which presents uses contracts [2]. The decentralized ledger is the key innovation of the blockchain, where users can view the transactions related to them anytime they want, however once approved and added to the blockchain, the transactions can neither be erased nor adjusted, which makes the blockchain irreversible and changeless. Every transaction is checked by the members by methods for pre-characterized validation and agreement instruments without confirmation or validation by any central entity. This reduce the expense as well as disposes of data corruption because of a single point failure, since ledger duplicates are synchronized over all the network participants [8].

The members of distributed networks can be any type of user, institution or organization sharing a duplicate of the ledger containing their legitimate transactions in a successive sequence. The ledger is comprised of a sequence of blocks as shown in Figure 2, connected together by their hashes in sequential order to maintain data integrity and timeliness. Each block consists of a set of transactions digitally signed by the proprietor and verified by the rest of the participants before being added to the block.

There are several consensus algorithms for the blockchain such as proof-of-work, proof-of-stake, proof-of-activity, and proof-of-relevance. This paper proposes a new algorithm named Proof of Authentication [9]. A classification of various consensus algorithms is shown in Figure 3.

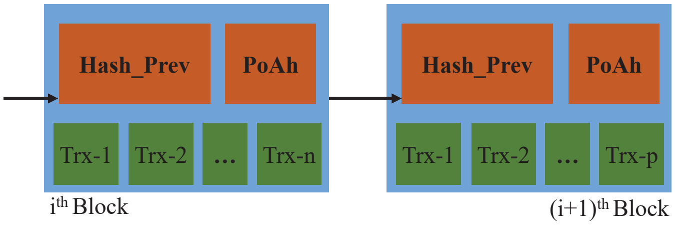


Figure 2. Block processing with transactions and hashing.

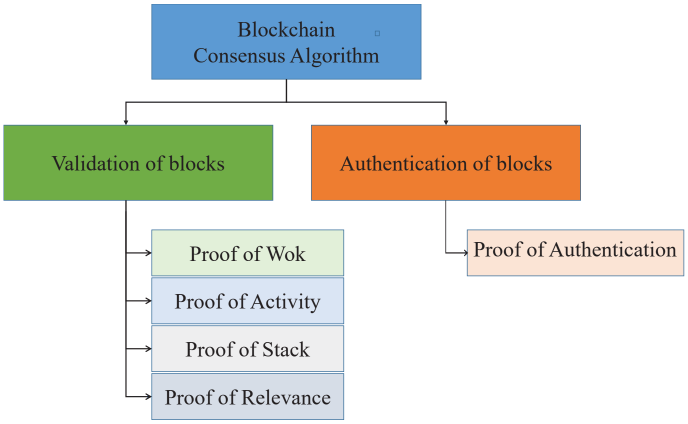


Figure 3. Taxonomy of blockchain consensus algorithms.

III. The Proposed Proof of Authentication

Proof-of-Authentication is a new consensus algorithm proposed in the current paper to make blockchain lightweight and compatible for resource-constrained devices. This algorithm follows traditional communications, where there are only updates during block validation.

At the start of the process, individual precipitants/nodes in the network generate transactions (Trx) with the data or processes and combine them into a block. The transactions and block formation details are as shown in Figure 2. The nodes broadcast the blocks for further evaluation. Individual nodes are responsible for public and private key generation (PuK and PrK). This model uses the ElGamal method of encryption i.e. $y = g^x (modp)$, where y is the public key and x is the private key. The generator function g and prime number p are known to the public network. Before node broadcast, the source node uses its private key PrK, i.e. x, to sign the block and makes its public key PuK, i.e. y, available to everyone.

There must be trusted nodes within the network for block validation, where trusted nodes are deployed with the minimum trust value required for being a trusted node and other nodes have a trust value zero ‘0’. With each successful full authentication of a block, trusted nodes gain trust values.

Once the block is received by the trusted node, it is processed to evaluate its authenticity by getting the source

Algorithm 1: Procedure of the Proposed PoAh.

```
Inputs : All nodes in the network follow $SHA-256$
         Hash Individual node has Private $(PrK)$
         and Public key $(PuK)$
1  Nodes combine transactions to form blocks
2  $(Trx^{+}) \rightarrow$ blocks
3  Blocks sign with own private key
4  $(S_{PrK})$(block) $\rightarrow$ broadcast
5  Trusted node verifies signature with source public key
     $(V_{PuK})$(block) $\rightarrow$ MAC Checking
6  if $Authenticated$ then
7      $block||PoAH(D) \rightarrow$ broadcast
8      $H(block) \rightarrow$ Add blocks into chain
9  else
10     DROP the block
11 GOTO $(Step-1)$ for next block
```

node public key, i.e. y. Based on the asymmetric cryptography property, the signature can be validated only with the use of the public key. Based on the discrete log problem property, one cannot compute the value of x, when other values are known to them. After signature validation, the trusted node also checks the MAC value for a second round of evaluation. After successful authentication, the trusted nodes broadcast the block to the network with PoAh identification. Following this, individual nodes in the network find the PoAh information from the block to add in the chain. Individual nodes compute the hash of the block and keep it to link the next block and the previously computed hash value is stored in the current block to maintain the chain, as shown in Figure 2. The steps of the PoAh procedure are given in Algorithm 1.

Trusted nodes use PuK (y) to verify the authentication of the block, followed by MAC checking for a second round of validation. Signature verification and MAC calculation takes negligible time even with resource-constrained devices [10], [11]. The next section provides experimental results for scalability evaluation.

IV. Experimental Evaluations

The proposed Proof-of-Authentication (PoAh) has been evaluated by us in both simulation and testbed. The experimental details for reproducibility and experimental results presented as the quality of the proposed algorithm.

A. Simulation Evaluation

The Proof-of-Authentication consensus algorithm is simulated in the Python programming language to evaluate its performance. For this experiment, there is a total of five participants in the network with two miner/trusted nodes and the block size is fixed to 35 bytes, where multiple transactions are kept in one block. All participants of the network use asymmetric key cryptography for the encryption and finally sign the certificate with their own private key. The block format is extended with more details in the context of simulation with format ¡Source ID, “Signature”, MAC, Trx1, Trx2, . . . >. The details of the block structure and block validation are shown in Figure 4 and 5, respectively. Figure 6 shows a sample output of PoAh evaluation.

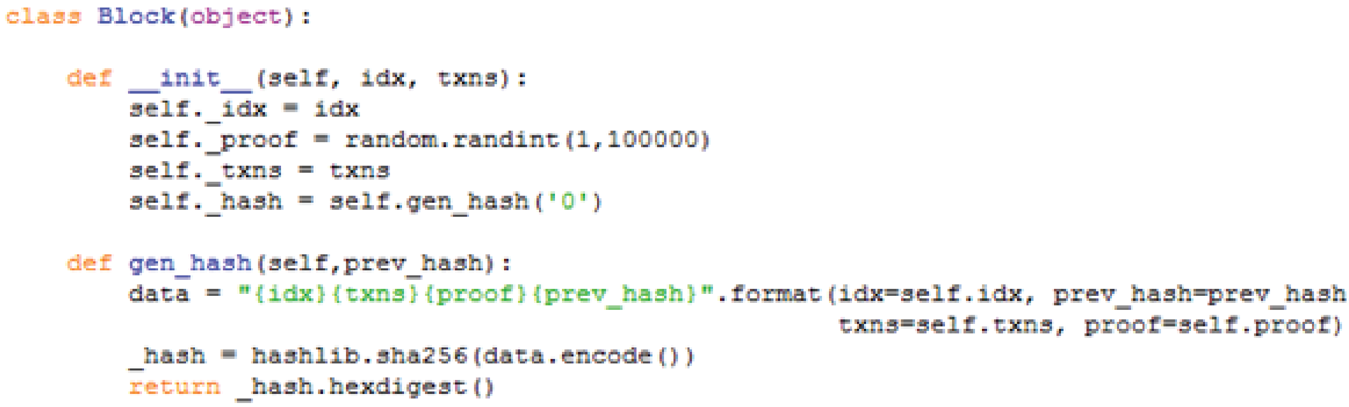
Table I
PoAh time for authentication of blocks.

PoAh Serial No.	Time taken for block validation in Seconds
1	3.3
2	3.4
3	2.8
4	4.02
5	2.9
6	3.8
7	3.4
8	3.42
9	3.38
10	3.23

Individual participants generate transactions and blocks, and sign them with their own private key before broadcasting. Trusted nodes validate the blocks with the source signature but using its public key. After successful authentication, trusted nodes again broadcast to all the participants in the network to keep a copy of the block into their ledger. The simulation results indicate an average time for the Proof-of-Authentication of 3.34 seconds. The average is computed from the PoAh time from 10 iterations. The results from those 10 iterations are listed in Table I, where the simulation follows the steps of Algorithm 1.

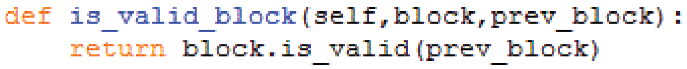
B. Testbed Evaluation

The proposed Proof-of-Authentication is implemented in a real-time testbed to evaluate its sustainability. The testbed implementation includes five Raspberry Pis, where three Pis are associated with multiple sensors (such as temperature, humidity and touch) and the other two work as the trusted nodes to the network. All Pis are configured with 1.2 GHz 64/32-bit quad-core ARM Cortex- A53 CPU, 1 GB LPDDR2 RAM at 900 MHz and Broadcom BCM2837 system-on-chip and use the SQLite database to maintain the index, public/private keys and cryptographic hashes. They are connected with each other through the Internet. An individual Pi collects data to form a block and then broadcasts it for authentication by the trusted Pis which maintain a database to find the public key of the source to verify the signature and broadcast the authenticated blocks after successful authentication. This implementation follows the step of Algorithm-1. The complete testbed setup is as shown in Figure 7. With the process of Proof-of-Authentication, the Pis take an average of 3.8 seconds while the total end-to-end time from block formation to becoming a part of blockchain is 4.4 seconds. We have concluded this number from 15 operations. The end-to-end time is subjective based on the medium of communications and computational power of the Pis. The testbed evaluation is mainly focusing on finding the sustainability of Proof-of-Authentication in resource-constrained distributed systems.



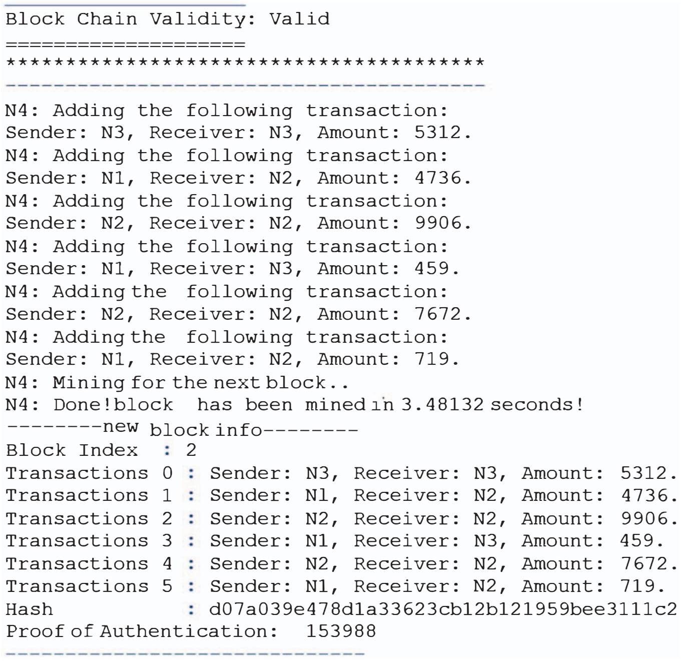
```
class Block(object):

    def __init__(self, idx, txns):
        self._idx = idx
        self._proof = random.randint(1,100000)
        self._txns = txns
        self._hash = self.gen_hash('0')

    def gen_hash(self,prev_hash):
        data = "{idx}{txns}{proof}{prev_hash}".format(idx=self.idx, prev_hash=prev_hash,
                                                      txns=self.txns, proof=self.proof)
        _hash = hashlib.sha256(data.encode())
        return _hash.hexdigest()
```

Figure 4. Block structure.



```
def is_valid_block(self,block,prev_block):
    return block.is_valid(prev_block)
```

Figure 5. Block validation.



```
Block Chain Validity: Valid
====================
****************************************
----------------------------------------
N4: Adding the following transaction:
Sender: N3, Receiver: N3, Amount: 5312.
N4: Adding the following transaction:
Sender: N1, Receiver: N2, Amount: 4736.
N4: Adding the following transaction:
Sender: N2, Receiver: N2, Amount: 9906.
N4: Adding the following transaction:
Sender: N1, Receiver: N3, Amount: 459.
N4: Adding the  following transaction:
Sender: N2, Receiver: N2, Amount: 7672.
N4: Adding the  following transaction:
Sender: N1, Receiver: N2, Amount: 719.
N4: Mining for the next block..
N4: Done!block  has been mined in 3.48132 seconds!
--------new block info--------
Block Index   : 2
Transactions 0 : Sender: N3, Receiver: N3, Amount: 5312.
Transactions 1 : Sender: N1, Receiver: N2, Amount: 4736.
Transactions 2 : Sender: N2, Receiver: N2, Amount: 9906.
Transactions 3 : Sender: N1, Receiver: N3, Amount: 459.
Transactions 4 : Sender: N2, Receiver: N2, Amount: 7672.
Transactions 5 : Sender: N1, Receiver: N2, Amount: 719.
Hash           : d07a039e478d1a33623cb12b121959bee3111c2
Proof of Authentication:  153988
------------------------------
```

Figure 6. Sample output of PoAh.

The testbed implementation process follows Algorithm 1 strictly to get the scalability of the network system.

It may be noted that Proof-of-Work (PoW) algorithm while running in high-performance computing resources has a latency in the order of 10 mins. Our proposed PoAh while running in limited computer resources has a latency in the order of 3 secs. Thus, our proposed PoAh is at least 200× faster than PoW which is used in traditional blockchain, and hence is highly scalable for large datasets which will run faster, will use minimal resources, and will have minimal energy consumption footprint.

From the above theoretical and experimental studies, we conclude that Proof-of-Authentication can replace Proof-of-Work by introducing an authentication mechanism into resource-constrained networks thus opening the door for scalable blockchain that can handle large amounts of data with limited resources with smaller latency [10], [11], [9].

V. Conclusions

This paper provides a new consensus algorithm named Proof-of-Authentication for lightweight blockchain. Lightweight blockchain is always a key factor for resource-constrained distributed systems to eliminate centralized dependencies. The proposed consensus algorithm is validated with analysis and experiments. The experiment is conducted in both simulation and testbed environments to evaluate its scalability. In the future work we plan to analyze the proposed PoAh algorithm with bigdata sets in larger systems. The security of the proposed algorithms for various benchmark attacks will be undertaken in future work.

References

[1] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, “Blockchain for IoT security and privacy: The case study of a smart home,” in *Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2017, pp. 618–623.

[2] D. Puthal, M. S. Obaidat, P. Nanda, M. Prasad, S. P. Mohanty, and A. Y. Zomaya, “Secure and sustainable load balancing of edge data centers in fog computing,” *IEEE Communications Magazine*, vol. 56, no. 5, pp. 60–65, 2018.

[3] S. K. Mishra, D. Puthal, B. Sahoo, S. Sharma, Z. Xue, and A. Y. Zomaya, “Energy-Efficient Deployment of Edge Dataenters for Mobile Clouds in Sustainable IoT,” *IEEE Access*, 2018.

[4] D. Minoli, K. Sohraby, and J. Kouns, “Iot security (iotsec) considerations, requirements, and architectures,” in *Proceedings of the 14th IEEE Annual Consumer Communications & Networking Conference (CCNC), 2017*, 2017, pp. 1006–1007.

[5] J.-H. Lee and H. Kim, “Security and privacy challenges in the internet of things [security and privacy matters],” *IEEE Consumer Electronics Magazine*, vol. 6, no. 3, pp. 134–136, 2017.

[6] D. Puthal, S. Nepal, R. Ranjan, and J. Chen, “DLSeF: A dynamic key-length-based efficient real-time security verification model for big data stream,” *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 16, no. 2, p. 51, 2017.

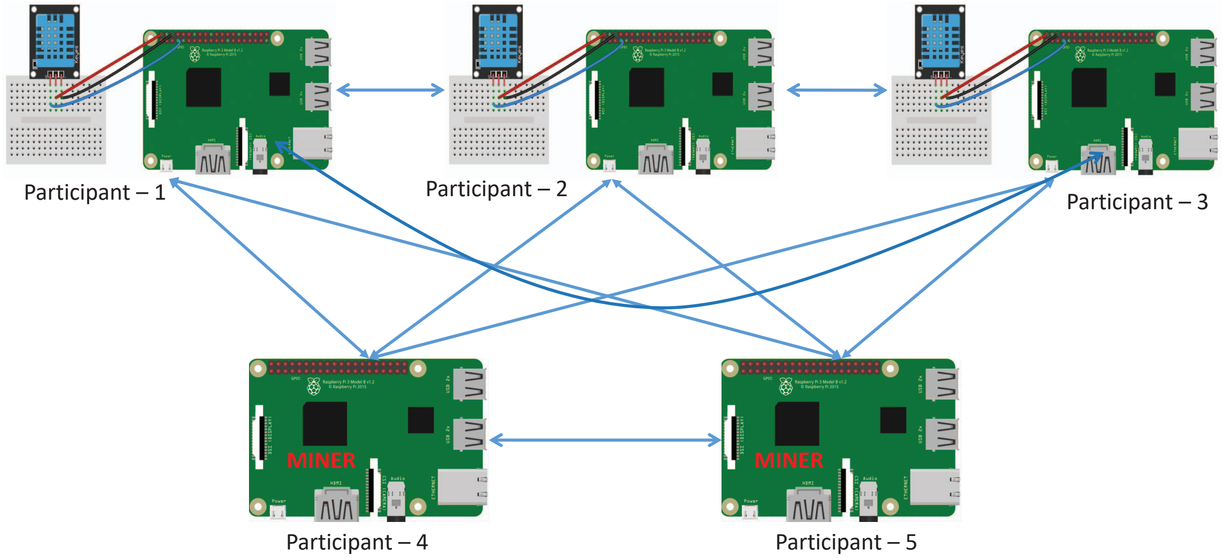


Figure 7. Block processing with transactions and hashing.

[7] D. Puthal, N. Malik, S. P. Mohanty, E. Kougianos, and C. Yang, “The blockchain as a decentralized security framework,” *IEEE Consumer Electronics Magazine*, vol. 7, no. 2, pp. 18–21, 2018.

[8] D. Puthal, N. Malik, S. P. Mohanty, E. Kougianos, and G. Das, “Everything You Wanted to Know About the Blockchain: Its Promise, Components, Processes, and Problems,” *IEEE Consumer Electronics Magazine*, vol. 7, no. 4, pp. 6–14, 2018.

[9] D. Puthal and S. P. Mohanty, “Proof of Authentication: IoT-friendly Blockchains,” *IEEE Potentials*, vol. 38, no. 1, 2019.

[10] N. Kshetri, “Can blockchain strengthen the internet of things?” *IT Professional*, vol. 19, no. 4, pp. 68–72, 2017.

[11] S. Alharbi, P. Rodriguez, R. Maharaja, P. Iyer, N. Bose, and Z. Ye, “Focus: A fog computing-based security system for the internet of things,” in *Proceedings of the 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, 2018, pp. 1–5.