Blockchain at the Edge: Performance of Resource-Constrained IoT Networks

Sudip Misra, *Senior Member, IEEE,* Anandarup Mukherjee, *Student Member, IEEE,* Arijit Roy, *Student Member, IEEE,* Nishant Saurabh, Yogachandran Rahulamathavan, and Muttukrishnan Rajarajan, *Senior Member, IEEE*

Abstract—The proliferation of IoT in various technological realms has resulted in the massive spurt of unsecured data. The use of complex security mechanisms for securing these data is highly restricted owing to the low-power and low-resource nature of most of the IoT devices, especially at the Edge. In this work, we propose to use blockchains for extending security to such IoT implementations. We deploy a Ethereum blockchain consisting of both regular and constrained devices connecting to the blockchain through wired and wireless heterogeneous networks. We additionally implement a secure and encrypted networked clock mechanism to synchronize the non-real-time IoT Edge nodes within the blockchain. Further, we experimentally study the feasibility of such a deployment and the bottlenecks associated with it by running necessary cryptographic operations for blockchains in IoT devices. We study the effects of network latency, increase in constrained blockchain nodes, data size, Ether, and blockchain node mobility during transaction and mining of data within our deployed blockchain. This study serves as a guideline for designing secured solutions for IoT implementations under various operating conditions such as those encountered for static IoT nodes and mobile IoT devices.

Index Terms—Internet of Things, blockchain, Edge nodes, Ethereum, Constrained-networks

1 Introduction

A majority of the present-day IoT solutions are plagued by limitations such as constrained energy, limited computing capabilities, high-degrees of mobility, and many others. In terms of security and privacy of IoT devices, vulnerabilities such as weak or hardcoded passwords, insecure network segments, poorly protected interfaces, unsecured data access mechanisms, insecure data transfer mechanisms, and other challenges make a majority of IoT devices prone to easy manipulation and disruption. Furthermore, the massive deployments of IoT devices often make it impossible for a network administrator to pin-point malicious or compromised devices amongst the deployed devices.

The nature of the devices in IoT, especially at the Edge, is vastly heterogeneous. As the IoT devices at the Edge primarily focus on ensuring low-power connectivity and basic computation, a significant chunk of these Edge devices does not possess sufficient processing power or resources to host conventional network security mechanisms. Typically, IoT Gateways are popularly associated

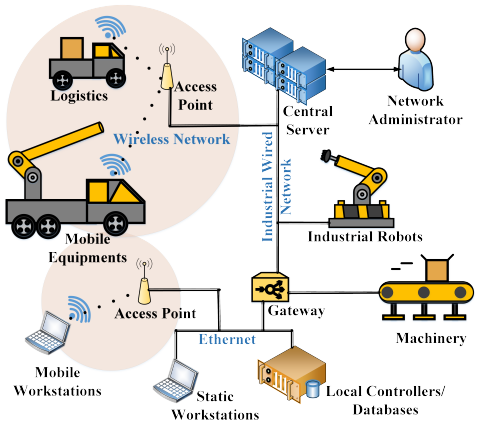


Fig. 1: An outline of a typical IoT-based Industrial ecosystem

S. Misra and A. Mukherjee are with the Department of Computer Science and Engineering at Indian Institute of Technology Kharagpur, India
A. Roy is with the Advanced technology Development Center at Indian Institute of Technology Kharagpur, India
N. Saurabh is with the Department of Electronics and Communication Engineering at National Institute of Technology Patna, India
Y. Rahulamathavan is with the Institute for Digital Technologies, Loughborough University London, UK
M. Rajarajan is with the Information Security Group, School of Engineering and Mathematical Sciences, City University London, London, UK

with providing security to the IoT devices/nodes under its operational purview. The current state-of-the-art IoT infrastructure relies on a centralized Gateway to process and aggregate data from IoT devices [1]. The centralized Gateway plays a vital role in ensuring the security of the sensed data.

The Edge devices may be static or mobile, and they not only sense and transmit data, but also perform actuation based on the data received from other IoT devices. This trend clearly shows that the majorly adopted centralized approach is not scalable and will soon become a bottleneck, which necessitates distributed technologies to replace the role of the Gateway. A centralized approach often leaves the IoT nodes under the domain of a Gateway, quite open to security breaches such as unauthorized access to data directly from the Edge devices. Rather than focusing on traditional security solutions, which rely majorly on remotely hosted security mechanisms such as at Cloud or centralized Gateways, the requirements of IoT-based systems necessitate distributed solutions [2]. These distributed solutions primarily focus on the IoT devices at the Edge [3] or even utilize hardware-based security [4].

Towards this objective, we analyze the performance and feasibility of using blockchains – a promising distributed security paradigm for ensuring data security for IoT-based systems [5]. Architecting a blockchain-based solution for IoT systems at the Edge requires addressing the following challenges:

- More the number of Edge devices in the IoT ecosystem that is part of the blockchain, more is the work-load of each of these blockchain nodes. The generally constrained nature of the network associated with the IoT systems/devices further makes it challenging for the devices to partake in network-based blockchain operations reliably.
- Blockchains require real-time synchronization between its nodes. Most of the constrained IoT Edge devices do not have an internal clock for time synchronization, making it necessary to come up with solutions to address this lacuna.
- The resource-constrained nature of most of the Edge IoT devices require mechanisms to handle processing-heavy blockchain-based operations.

Why Ethereum? – In this work, we deploy a Ethereum blockchain consisting of IoT Edge devices as its nodes and experimentally verify the performance of this approach. As Ethereum is an open and public blockchain, in addition to being highly customizable, we have considered it to test our implementation. Similar to Ethereum, solutions such as IOTA can be a viable choice for implementing our solution. However, unlike Ethereum, IOTA's central coordinator is a close-source project, which does not allow us to customize smart contracts and add features. Other blockchain frameworks, such as Hyperledger's Sawtooth and Fabric are permissioned blockchains, which restricts the free interaction of the IoT nodes with the blockchain system, in turn, making their configuration complex, especially for dense deployments and rapidly changing configurations of the nodes. The proposed work establishes the feasibility of using day-to-day IoT devices as blockchain nodes and can be used as nodes for different blockchain frameworks. Our implementation additionally allows for the integration of security features of Attribute-based Encryption (ABE) [6] and other encryption algorithms on top of the proposed blockchain via smart contracts. These encryptions can be reliably used to ensure the correctness of the time string and the time synchronization.

Provision for unified network time synchronization– To secure the data generated and exchanged between IoT devices in a distributed manner, we propose the use of low-power IoT Edge nodes (refer Table 1) as the blockchain nodes. These nodes are not only capable of continuing their regular sensing and actuation tasks, but also perform necessary blockchain functions such as verification, mining, and transactions. However, as most of the Edge devices do not have an internal clock, they have no provision to automatically synchronize their time to the network. To alleviate this problem, we additonally propose a centralized time server ensures synchronization of system/network time across the various resource-constrained Edge devices. The conjunction of a centralized time server and a decentralized blockchain approach, makes this work a hybrid one – not completely centralized, nor completely decentralized. Our implementation allows for the integration of additional security features and other encryption algorithms to the time string from the central server to the IoT nodes, and on top of the data transmitted to the blockchain from the distributed IoT nodes. These encryptions can be reliably used to ensure the correctness of the time string as well as preserving the privacy of the data being transmitted to the blockchain.

1.1 An IoT-based Industrial Ecosystem Application Scenario

We envision a real-life use case of an IoT-based industrial ecosystem for motivating the applicability of this work. Fig. 1 shows the significant physical and infrastructural components of an industrial complex. We choose an industrial complex primarily because of the massive density of deployed Edge devices, and the constrained nature of the network arising due to the high density of these devices and challenging areas of implementation – prone to interference and noise from the environment. Constraints to network and device capabilities are automatically induced in such ecosystems due to the presence of dedicated automation and control systems working with new as well as legacy infrastructures. It is common to see both wireless and many variants of wired connections for communication in industrial ecosystems. In continuation, the heterogeneity in devices in terms of their mobility, processing abilities, and energy consumption also makes it a challenging environment for implementing secure IoT systems.

The amounts of data generated and flowing through the network in an IoT-enabled industrial ecosystem are quite massive. The use of blockchain introduces the features of transparency and traceability to the IoT data generated within the industrial ecosystem. Both constrained IoT Edge nodes, as well as regular computing stations, can be incorporated within this setting. In our experimental evaluation, we fashion the blockchain nodes as such that they consist of both regular computing platforms such as PCs as well as constrained IoT Edge nodes consisting of Raspberry Pi boards (refer to Table 1). Here, we deploy a small four-node blockchain testbed. The preliminary, yet crucial trends and metrics obtained from this small-scale implementation is indicative of the overall behavior of our approach. From a security point of view, more nodes on the blockchain will increase the security and reduce the computational load from other nodes by sharing blockchain operations. Therefore, increasing the nodes will only enhance the security, trust and reliability. The data privacy of the system can be further ensured using techniques such as private transactions and ABE [6]. Private transactions have encrypted data within the blockchain transaction. Specific attributes depending on implementation scenarios can be used to develop smart contracts to ensure customized data privacy. For example, in the considered scenario, device locations or monitoring equipment details can be good attributes for defining group access policies for IoT monitoring implementation in industrial scenarios.

1.2 Contributions

The nature of the data plays a decisive role in evaluating the requirements of security and privacy to be used at the IoT devices. However, the integrity of data is an irrefutable need for all IoT data types and needs, which is ensured by the private blockchain. In this work, we make the following distinct contributions:

- We incorporate the heterogeneity of IoT devices by including both small nodes – constrained, with fewer resources and processing power – and large nodes – nodes with abundant resources and processing power. We incorporate network heterogeneity in our implementation by making use of both fixed Ethernet-based network connections as well as including WiFi-based connections.
- We also propose a centralized network time synchronization in conjunction with the decentralized blockchain. The proposed time synchronization allows for setting and coordinating time on the resource-constrained Edge devices, which do not have an internal clock.
- We evaluate the various interactions of constrained IoT devices with blockchain networks, even when they have heterogeneity in their connection and/or are mobile.

TABLE 1: Blockchain node specifications for our implementation

Features	Node-1	Node-2	Node-3	Node-4
Device	Raspberry Pi3-B+	Raspberry Pi3-B+	Dell Power Edge T410 server	Raspberry Pi3-B+
Processor	Quad Core 64 bit ARM cortex at 1.2 GHz	Quad Core 64 bit ARM cortex at 1.2 GHz	16x4 Core 64 bit at 2.67 GHz	Quad Core 64 bit ARM cortex at 1.2 GHz
RAM	1 GB	1 GB	32 GB	1 GB
Network connection	Ethernet	Ethernet	Ethernet	WiFi

1.3 Related Work

There have been several efforts in the recent past to integrate blockchain with IoT networks. Works such as the one by Lao *et al.* outline the challenges

associated with integrating and redesigning typically resource-intensive blockchain mechanisms – architecture, consensus, and traffic – with the inherently resource-constrained IoT devices [7], whereas Wu *et al.* provide a thorough analysis of issues and tentative solutions for implementing IoT-based blockchains by dividing their evaluation under four architectural layers – data, network, consensus, and application. Approaches such as hybrid public-private blockchains [8], inclusion of additional message verification devices with blockchains [9], incorporating smart contracts [10], [11], implementing ABE with blockchain transactions to ensure privacy of data [6], and many others [12] provide promision solutions and indicate the feasibility of using blockchains for IoT networks.

Dorri *et al.* demonstrate an energy-efficient use of blockchains in IoT systems by using distributed trust algorithm instead of Proof-of-Work (PoW) [13]. Similarly, utilizing Delegated Proof-of-Stake (DPoS) instead of PoW for enhanced privacy of data, Proof of Authority (PoA) [14], blockchain hosted at IoT gateways for dense deployments [15], blockchains for fog/edge devices [16], and other modifications to blockchain mechanisms [10] are some of the works ensuring reliable integration of blockchains for IoT, and that too with additional features. IoT blockchains have been successfully proposed for use in diverse, but complex application areas, such as smart cities [17], healthcare [11], crowd-sourcing [18], and others.

Extending the use of blockchains for Industrial IoT (IIoT) applications is even more complex as industrial IoT deployments are marred by the challenges of dense device deployments, heterogeneity (of devices, data, and protocols), increased interference/ signal distortions, and the need for real-timeliness of data and decisions. Systematic surveys, such as those by Choo *et al.* [19] and Mistry *et al.* [20] povide furtehr insights to the challenges and the upcoming solutions for the use of IoT blockchains in industrial scenarios. Approaches such as BASA [21], ELIB [22], LightChain [23], Tornado [24], and others show promising results for the use of modified IoT blockchains in industrial scenarios. Besides accomodating the base functionality of blockchains under constrained operating conditions, these approaches additionally offer the benefits of improved identity-based access management, lightweight concensus mechanisms, distributed and enhanced throughput management, reduced latencies, optimized resource efficiencies, certificateless cryptography, and many more.

However, most of the works consider fairly powerful computing devices at the edge of the IoT network, which are not necessarily non-real-time. Also, a majority of the works do not address the issues of mobility, network and device heterogeneity, and the need for device synchronization with the network time (especially for non-real-time devices such as Raspberry Pis). Through this work, we attempt to cover these gaps and provide a real-life evaluation of the implications of using blockchains in IoT networks.

2 System Model

In this work, we implement an Ethereum-based blockchain on heterogeneous IoT nodes, some of which connect to the blockchain over an Ethernet-based connection, whereas the others connect through a WiFi-based connection, forming a hybrid network connection as shown in Fig. 2. Further, adding to the device heterogeneity, the devices themselves have different specifications and processing capabilities, as outlined in Table 1.

IoT blockchain nodes have unique "ENODE" values and connect using these values. The "ENODE" value consists of a public key, an IPv4 address, and a port number. Simulating a real-life IoT implementation, we have incorporated heterogeneous IoT nodes, some with low processing power and reduced energy requirements (i.e., Raspberry Pi) and some with high processing power and more significant energy requirements (i.e., server, PC). The Raspberry Pi-based nodes connecting over WiFi are considered as mobile and treated as such during the performance evaluation of our setup. However, these IoT nodes in our blockchain are capable of independently handling their transactions as well as mining.

2.1 Incorporating blockchain for IoT

Fig. 2 outlines the representative network architecture of our implemented IoT blockchain. The network can be considered to consist of heterogeneous nodes (N1-N4). These nodes may consist of large static devices such as servers and PCs, or they may be small and portable consisting of Raspberry Pi boards. All these devices act as nodes in the blockchain. A switch connects an external backbone network to the internally formed network. The network connections from the switch may be either used for connecting physically to the IoT nodes through Ethernet or wirelessly through a

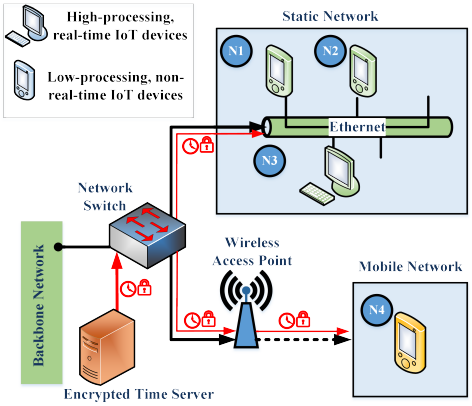


Fig. 2: The representative network architecture of our implemented IoT blockchain

wireless access point. An external (centralized) encrypted time server is also used to provide network time synchronization to the IoT devices, which are mostly non-real-time.

TABLE 2: Specifications of our private blockchain

Blockchain specifications	**Values**
Gas limit (in Hexadecimal)	0x47b760
Gas limit (in Decimal)	0x4700000
Difficulty	0x1
Consensus Engine used	Clique - Proof of Authority (PoA)
Time (in sec) each block takes	5 seconds
Number of accounts on each node	1
Accounts which are allowed to seal	Accounts of all the nodes
HomesteadBlock	1
EIP150 Block	2
EIP155 Block	3
EIP158 Block	3
Gasprice for node 1	3×10^{55} wei
Gasprice for nodes 2, 3, 4	3×10^{28} wei
Syncmode	full

We implement a private blockchain to account for the low-processing capabilities of the implemented IoT nodes, as well as keeping the data and transactions localized within an application area. Each of these nodes runs an Ethereum framework, the specifications of which are outlined in Table 2. Each of these nodes has an account associated with it over the Ethereum framework and uses a "CLIQUE- Proof of Authority (PoA)", instead of regular "ETHASH- Proof of Work (PoW)" to reduce mining times and reduce the average energy consumed by the nodes. The transaction of Ethers and data are performed based on the "ENODE" values of each node, which are subsequently mined by intended nodes. Post successful completion of a transaction, Ether balance is updated to a receiver node's account by the same amount it gets deducted from the sender's account. The Ether balance with the sender node was initially logged at $7.5^{19} wei$, the whole of which gets transferred to the receiver upon completion of a transaction. Unlike public blockchains, which deal with unknown and trustless systems, the private blockchains do not need an incentive-based mechanism to work.

We automate the process of an IoT node joining the blockchain, generating data, and performing transaction and mining operations. Algorithm 1 highlights this automation process. On power-up, each IoT node boots into a startup file containing the multi-threaded instructions and commands for time synchronization using the encrypted network-broadcasted time string and initialization of the node's Genesis file. Subsequently, each of the activated nodes checks for transaction data (to send or receive), which is then mined and submitted accordingly. Blockchain contracts can also be deployed similarly. Irrespective of a node's processing capabilities, the nodes are self-sufficient to carry out mining operations on their own. It is prudent to mention that in the absence of proper time synchronization, the connection between nodes is interrupted, resulting in association and disassociation with the blockchain. This drop in connection results in a significant increase in mining times at the affected nodes.

Algorithm 1 Node automation

```
while DEVICE POWER ON do
    Locate node blockchain automation file
    Initialize Genesis file
    Start mining
    if (Transaction Data == TRUE) then
        Submit the transaction and generate receipt
    end if
end while
```

2.2 Encrypted Time Synchronization

Operations such as mining rely heavily on the synchronization of time and its maintenance between the nodes of the blockchain. Our implementation requires the communication of an encrypted time string to a node joining the blockchain for the first time or every time it is powered on from a central time server. This provision has been kept mainly because of the absence of Real-Time Clocks (RTC) in the resource-constrained IoT devices. Every time these devices power-up, the internal clock

resets to the default value, which is unlike personal computers or machines with RTCs. Further, unless the sender and receiver nodes have a common system time, network security provisions prevent them from joining the network or communicating reliably, especially for blockchains.

Networked time synchronization: Any external efforts to include a network-based time synchronization should be secure enough to ensure long-lasting and interference-free membership of the IoT nodes to the blockchain. In case the time server or any message generated from it is compromised or altered, the IoT nodes forming the blockchain will get dissociated, resulting in the breakdown of the blockchain. To avoid any such eventuality, we additionally implement the use of an encrypted time string from a centralized time server (refer Fig. 2), which can be read only by the member nodes of the implemented blockchain as outlined in Algorithm 2. Further, approaches such as ABE can be just as easily incroporated with this approach to further strengthen the reliability and security of the time string. Considering a typical IoT scenario, the inclusion of thousands, if not millions of devices in the proposed scheme would overwhelm even high-end servers. However, using approaches such as ABE with suitable group access policies, the same time string can be utilized by a group of IoT nodes, instead of individually customizing and encrypting each time-string for synchronization purposes [6]. This enhances the scalability of the time synchronization approach. Additionally, as the proposed time synchronization is centralized, the effects of anomalies associated with distributed systems, such as Byzantine failure, are absent.

Resilience of the proposed approach: The time server has a record of all the member nodes of the blockchain along with their "ENODE" values. The IP of each node corresponding to its "ENODE" value gets periodically updated at this time server. For our encryption, we adopt a *different node – different encryption* policy [6], which adds an additional level of security to our IoT blockchain. The synchronizing encrypted time string is customized according to each of the registered member nodes, which can only be decrypted by the target IoT node using its "ENODE" value as the private key. Any attempts to falsify or manipulate the IP address of the node or the ENODE address will result in a clash in the records at the server, alerting the network administrator of this attempt. As the server broadcasts the time strings over the blockchain network, all the nodes can see the encrypted message, but only the designated node with the proper "ENODE" value can decrypt it. The mapping of IP addresses and ENODE values also prevents the duplication of ENODE values by malicious nodes. Further, the encrypted time string meant for a node will be relayed multiple times, similar to a typical networking scenario, if the time server is not directly connected to the target node.

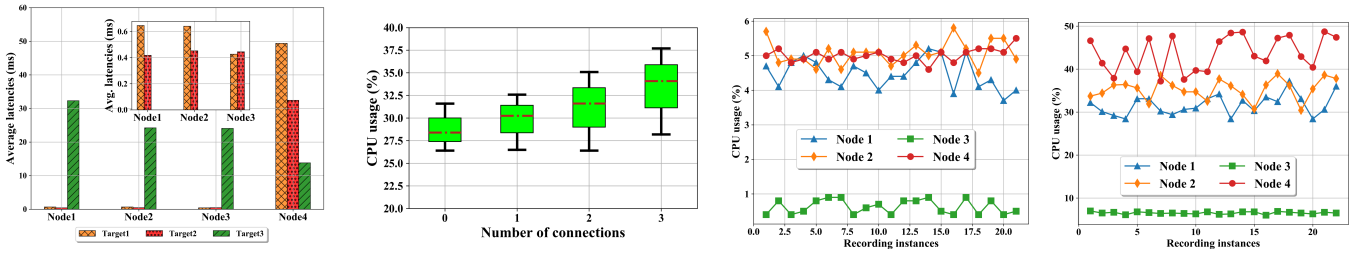
Algorithm 2 Time Synchronization

SERVER
$n \leftarrow$ number of nodes
$message \leftarrow$ time string
Replicate $ENODE$ and IP of all the nodes in the time server
for $i = 1$ to n **do**
  $IP[i] \leftarrow IP$ of i^{th} node
end for
for $i = 1$ to n **do**
  $key[i] \leftarrow ENODE$
  Encrypt time string using $ENODE \rightarrow ENODE(message)[i]$
  for $all\ i$ **do**
    send $IP[i] \leftarrow ENODE(message)[i]$
  end for
end for
RECEIVER (intended node)
$TIME \leftarrow$ original time of the node
During node startup **DO**
Copy $ENODE \rightarrow$ file.txt
Receive encrypted time T_{ep}
Auto decrypt using $ENODE$ of the node T_{dp}
Set $TIME = T_{dp}$

The case of compromised IoT nodes: Concerning a Man-in-the-Middle Attack for modifying the time, the encrypted time server (refer Fig. 2) is tasked with periodically updating the mapping of ENODE and IP addresses of the participants in the private blockchain. As the ENODE values are unique to each blockchain node, these ENODE values can be uniquely mapped to the nodes' IP addresses. Even if there is a change in the node's IP address, the periodic check by the time server ensures its update in the mapping repository. Once a node with the proper IP address receives the encrypted time string meant for it, only it can decode it using its unique ENODE value. The mapping of IP addresses and ENODE values also prevents the duplication of ENODE values by malicious nodes.

3 Performance Evaluation

In this work, we establish a private Ethereum blockchain with four nodes, following the architecture outlined in Fig. 2, the exact specifications



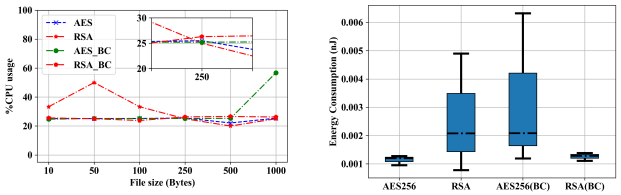
(a) Average network latencies across nodes (b) Average CPU usage vs. number of connecting nodes (c) CPU usage before mining (d) CPU usage during mining

Fig. 3: Network and node performance characteristics for our implemented IoT blockchain

of which are briefly outlined in Table 1. Two of these nodes (nodes 1 and 2 in Table 1) are non-real time, static systems with constrained processing power and energy, and which join the blockchain network through the Ethernet. The third node (node 3) has significant processing resources and does not have any energy constraints as it draws power directly from the grid. This node also takes part in the blockchain through a dedicated Ethernet-based connection and is deemed a static node. Finally, the last node is yet another non-real-time, resource, and energy-constrained node similar to nodes 1 and 2, but takes part in the blockchain through a wireless connection (WiFi) as it is mainly mobile. It is to be noted that both the Ethernet and WiFi-based networks are not established dedicatedly for this evaluation, but are part of a single institutional network over which a significant number of users communicate simultaneously at any time of the day.

3.1 Effect of Encryption Algorithms

To evaluate the performance of our approach and ensure additional security and privacy measures, we encrypt the timestring from the centralized server to the IoT nodes, as well as the data from the nodes being forwarded on the blockchain using two algorithms – RSA and the 256-bit AES. We analyze the standalone effect of these algorithms on the CPU usage and energy consumption of the resource-constrained devices, as shown in Fig. 4. We have first used AES and RSA in a standalone mode to encrypt data on the IoT node. Thereafter, both of these encryption algorithms are used to encrypt data before it is mined in the blockchain – the IoT node simultaneously runs one of these algorithms along with blockchain operations, which are denoted as AES256(BC) and RSA(BC) in Fig. 4(a). From the same figure, we observe that for varying data sizes, the four algorithms have comparable CPU usage (neglecting the intermittent outlier behavior observed in some of the readings). We calculate the processing energy required for these security measures from the CPU utilization of each type of IoT device [25]. From Fig. 4(b), we observe that although the energy consumed for executing each of the four algorithms (AES, RSA, AES256(BC), and RSA(BC)) is significantly small, the RSA and AES256(BC) have a high variance for data sizes ranging from $10B$ to $1000B$.



(a) Percentage of CPU usage (b) Energy consumed (in nJ)

Fig. 4: Performance of various security measures on a resource-constrained IoT node

3.2 Performance of Encrypted Time Synchronization

As the proposed time synchronization essentially follows the same approach as evaluated in Section 3.1, the performance of proposed encrypted time synchronization concerning CPU utilization and node energy follows the same trend. However, it is to be noted that the data size of the encrypted time string lies between $30B$ - $50B$, the performance of which is reflected in the initial stages of the plots in Fig. 4. We observe an average latency of 0.4 ms to 0.7 ms on each of the nodes 1, 2, and 4 while they receive information/ connection requests from other nodes. However, node 3 connecting to the network through a WiFi-based connection encountered average latencies of around 13 ms to 50 ms when receiving messages from the other nodes. Similarly,

nodes 1, 2, and 4 observe average network latencies of up to 24ms to 33ms, when receiving messages from node 3.

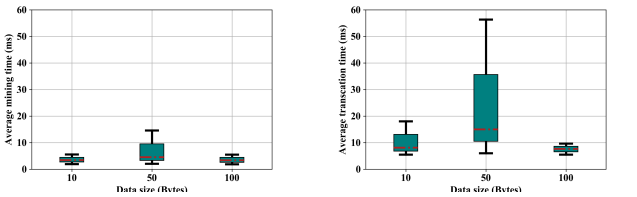
3.3 Effect of Network Latency

Considering the network architecture discussed previously, Fig. 3(a) shows the comparison between network latencies while sending *ping* packets from each node to every other node (designated as Targets 1 -3) in the network. We observe that for *ping* queries over the Ethernet-connected nodes, the response time is significantly lower than that of the one connected over WiFi. Additionally, we observe that the response time for *ping* from Node-3 (server) is relatively lower than the responses from the resource-constrained nodes (Nodes- 1 and 2), even when connected over the same Ethernet-based connection. These relatively higher latencies incurred due to the resource-constrained nodes (Nodes-1 and 2) is attributed to the time taken by them to process the packets. In continuation, the significantly higher latencies at Node-4 can be attributed both to its resource-constrained nature, requiring more time to process the packets, as well as its mobility, which causes it to have unstable network characteristics. These latencies are crucial in estimating the performance of our implemented IoT blockchain and act as the network performance baseline. In PoW blockchains, increased network latencies can lead to increased block convergence times and failure of six confirmations [26]. However, in this work, as we adopt a PoA concensus mechanism, which is much faster than PoW and PoS based mechanisms, the effect of network latency on the security of the blockchain is minimized. In PoA, only reputed validators can approve transactions on the blockchain, which is very useful for IoT-based scenarios.

3.4 Effect of Increase of Node on CPU Usage

Fig. 3(b) shows the average CPU usage (denoted in %) for a randomly selected constrained node in our blockchain network. We observe that as the number of network connections to that node increases, the nodes' average CPU usage goes up to maintain the connections to and from it. An important takeaway from this observation is that resource-constrained nodes support only a limited number of simultaneous network connections, which necessitates the use of distributed security solutions for reliable use of such nodes.

Further, Fig. 3(c) represents the CPU usage at each of the four implemented nodes before joining the blockchain, whereas Fig. 3(d) represents the CPU usage in the same nodes during mining in the blockchain. From Figs. 3(c) and 3(d), we observe that the three constrained nodes (Raspberry Pi) incur almost 5-8 times the CPU usage as compared to the regular node (server). Additionally, the mobile constrained node (connected to the WiFi), incurs further resource usage (CPU usage) as compared to the constrained nodes connected to the Ethernet.



(a) Average mining time (b) Average transaction time

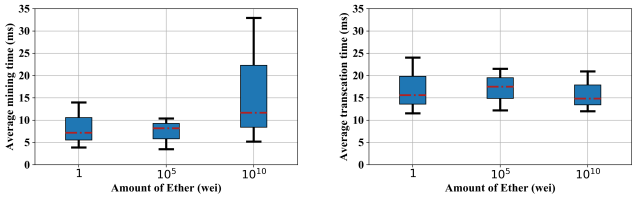
Fig. 5: Variation in blockchain performance with the variation in data size

We further observe that being part of the blockchain and performing its operations induces a massive increase in CPU usage of the devices by almost 10 times as compared to when the devices are operating on their own (refer Figs. 3(c) and 3(d)). For resource-constrained nodes, the percentage CPU usage is about 5 to 7 times more as compared to that for the node with ample storage and high processing power. As in our implemented blockchain, nodes 1, 2 and 4 are resource-constrained, we observe their average CPU usage to be around 31.686%, 35.323%, 43.704% respectively, while node 3 which is a server accounts for about 6.536% of CPU usage during blockchain mining operations.

3.5 Effect of Data Size

Fig. 5 shows the effect of data size on the IoT blockchain operations of our implemented system from the perspective of the static nodes. Fig.5(a) shows the variation in mining time when the size of the data used for transacting over the blockchain is varied while the amount of Ethers transacted is kept fixed at 750 wei. The sender and receivers involved in the transaction are also kept fixed. We evaluate the performance of mining in our implemented IoT blockchain by using transaction data packets of size 10 bytes, 50 bytes, and 100 bytes. We observe the same variation in mining time for different data sizes over 30 repetitions of this exercise for each data size. Except for some random cases where

mining time may show an increased deviation from the norm (as can be seen for the 50 byte data packet in Fig. 5(a)), the mining time for all these data sizes remains reasonably consistent. We attribute these random unexpected values to unstable and congested network behavior and the induced latency thereof.



(a) Average mining time (b) Average transaction time

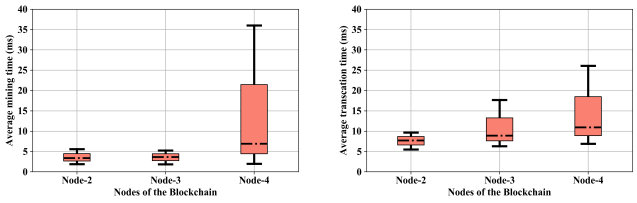
Fig. 6: Variation in blockchain performance with the amount of Ethers transferred

Similarly, Fig. 5(b) shows the variation in transaction time for the same repeat of the exercise outlined above. Similar to the observed behavior in mining time, the transaction operation also reports some unaccounted-for surge in transaction time, which we again attribute to fluctuating network conditions. As the plot in Fig. 5(b) shows the average behavior, considerable fluctuations in network conditions tend to disturb the norm, which for most of the cases, is reasonably consistent.

3.6 Effect of Ether

Fig. 6 shows the effect of Ethers on the IoT blockchain operations of our implemented system from the perspective of the static nodes transacting a data of 100 bytes over the blockchain. Fig. 6(a) shows the variation in mining time on varying the number of Ethers transacted while keeping the data size fixed to 10 bytes, between pre-determined senders and receivers. We transact 1 wei, 10^5 wei, and 10^{10} wei in our blockchain for over 30 times in each case. We observe that the variations in mining time remain almost the same for all cases except for some unexpected random fluctuations because of varying network conditions, which is evidenced from the apparently high error bar in the plots.

Similarly, Fig. 6(b) shows the variation in transaction time for the same exercise as described above. For each of the three cases, i.e., for 1 wei, 10^5wei, and 10^{10} wei, we observe almost the same type of variations as reported previously. We attribute this randomness in behavior to unstable network conditions. The randomness distorts the norm of



(a) Average mining time (b) Average transaction time

Fig. 7: Variation in blockchain operation times with respct to various nodes

the readings for all three cases, as is evident from the significantly larger error bar in the plots.

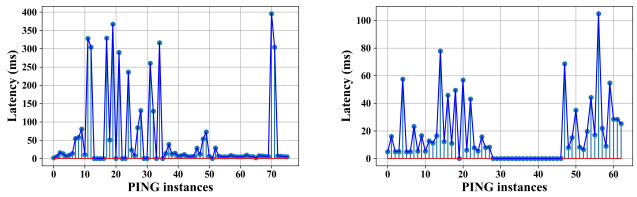
3.7 Effect of Node Characteristics

Fig. 7(a) shows the variation in mining time at node-1 with the change of receiver nodes while keeping the data size fixed at 100 bytes and the number of Ethers at 750 wei. The make of the nodes is described in Table 1. We observe that there is almost no difference in mining time when nodes 2 and 3 – connected to the blockchain over an Ethernet-based connection – act as receivers of the data. However, there is a significant rise in mining time when node-4, which connects to the blockchain over WiFi, is made the receiver of data from node-1. The error bar for the plot of mining time at node-4 indicates a massive fluctuation of values, indicating unstable network connection.

Similarly, Fig. 7(b) shows the variation in transaction time for transactions between node-1 and the other three nodes under the same operating conditions, as mentioned earlier. Here we observe that there is an increase in the average transaction times at node-1 when the transactions are performed between it and nodes 2-4. The increase in transaction time at nodes 3 and 4 are caused due to random variations in network latencies due to intermittent network connections, as evidenced by the relatively higher error bars in the plots for these two nodes.

3.8 Effect of Node Mobility

In contrast to the static node analysis until Section 3.7, in this Section, we evaluate the performance of the network as well as the implemented blockchain from the perspective of a mobile node. The mobile node under consideration is node-4, which connects to the blockchain through a WiFi-based connection, which gives it the ability to relocate without changing any physical configurations quickly. To estimate the network quality available to this node when



(a) Mobile node to a static node (b) Static node to a mobile node

Fig. 8: Network latency encountered during *ping* operation

it is mobile, we perform two network-based tests – 1) check the network response time when the mobile node queries an address over the network during mining operation, and 2) check the network response time when a static node queries the mobile node's address during mining operation.

Fig. 8(a) shows the network latencies witnessed by node-4 during the first test. Whereas, Fig. 8(b) shows the network latencies witnessed by a static node during the second test. It is to be noted that the static node connects to the network through a fixed Ethernet-based connection. We observe that there is a considerable variation in the recorded network latencies as node-4 moves through regions of weak and strong WiFi signal strengths. This mobility and fluctuations in signal strength further give rise to intermittent connectivity issues such as the unavailability of the network (as seen in Fig. 8(a) between instances from 33 to 46). The network stays unreachable until the mobile node enters into a zone of good signal strength. As a result of this behavior, there is an induced lag in mining times whenever mobile nodes connect to the blockchain over constrained networks.

Fig. 9(a) shows the variation in mining time for two different data sizes, i.e., 10 bytes and 100 bytes while transacting between a static and a mobile node in our implemented blockchain. The considerable variations in mining times, as evidenced by the error bars, are a result of unstable network connections when the mobile node traverses through zones of good and bad signal strengths. Considering equal network variations during the transference of the two data blocks, we observe that the norm for 100 bytes is higher than that for 10 bytes of data, indicating higher mining time incurred for more significant data sizes.

Similarly, Fig. 9(b) shows the variation in transaction time for the two selected data sizes, i.e., 10 bytes and 100 bytes while transacting between a static and a mobile node in the network. We again observe the average transaction time of 100 bytes data packet to be slightly higher than that for 10 bytes data packets, which is due to the increase in time required to transmit and process the data. The variations and increased values of the error bars signify intermittent network connectivity, resulting in higher transaction times for the mobile node.

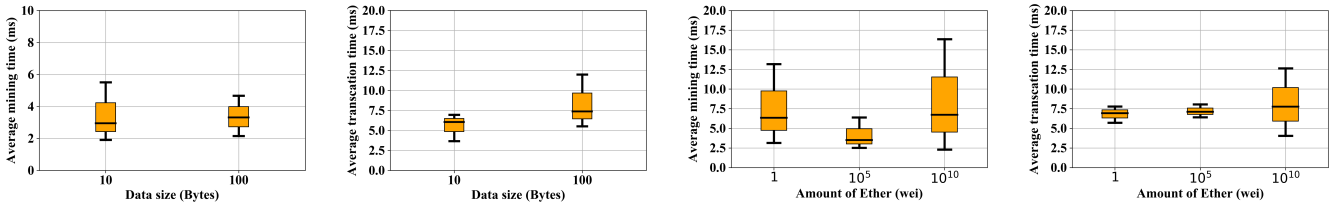
Further, Fig. 9(c) shows the variation in mining time for three different amounts of transacted Ether, viz. 1 wei, 10^5 wei, and 10^{10} wei while transacting from the static node to the mobile node in the blockchain. Here, we consider the bar for 10^5 wei to be the standard baseline as its error bars are relatively much lesser than that of the other two bars. The increased error bars for 1 and 10^{10} wei indicate an increase in network-based disturbance, which affects the mining operation, even for increased Gas prices.

Similarly, Fig. 9(d) shows the variation in transaction time for the three different amounts of transacted Ether, viz. 1 wei, 10^5 wei and 10^{10} wei, when they transfer from a static node to a mobile node of our blockchain. As compared to the mining time, the transaction time experiment witnesses relatively lesser network disturbances, as evidenced by the smaller error bars for 1 wei and 10^5 wei.

4 Conclusion

As a significant majority of IoT Edge devices and IoT networks are resource-constrained, the provision for incorporating reliable security measures is often not available for these devices. These restrictions have resulted in an abundant presence of unsecured data propagating through IoT networks and make the Edge devices susceptible to unauthorized access and tampering. In this work, we have proposed and analyzed the feasibility of incorporating heterogeneous IoT Edge devices as functional blockchain nodes to extend the feature of decentralized security to resource-constrained IoT deployments. We also implement an encrypted network-based time-synchronization mechanism to enable the non-real-time IoT Edge nodes to co-exist in the blockchain.

We conclude that the feasibility of utilizing a blockchain-based decentralized security at the IoT Edge devices itself is significantly high in terms of restricting data repudiation and enforcing trust in the constrained deployment, which were previously susceptible to manipulation. However, the underlying connectivity of the network and the minimum



(a) Average mining time vs data size (b) Average transaction time vs data size (c) Average mining time vs Ethers used (d) Average transaction time vs Ethers used

Fig. 9: Evaluation of parameters during transmission of data from static to mobile IoT nodes

processing capabilities of the blockchain nodes control the blockchain performance, which further restricts the nature of the sensing and actuation tasks that the Edge node can accommodate. In the future, we plan to design and develop methodologies to incorporate processing-intensive tasks such as computer vision with our implemented blockchain at the Edge, in addition to the evaluation of the large-scale behavior of the proposed solution as an extension of this work.

ACKNOWLEDGEMENT

This work is sponsored by the University Grants Commission (UGC)-UK India Education Research Initiative (UKIERI) Joint Research Programme (UKIERI-III) under project file No. 184-17/2017(IC).

REFERENCES

[1] V. Thangavelu, D. M. Divakaran, R. Sairam, S. S. Bhunia, and M. Gurusamy, "DEFT: A Distributed IoT Fingerprinting Technique," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 940–952, 2019.

[2] O. Novo, "Scalable Access Management in IoT using Blockchain: a Performance Evaluation," *IEEE Internet of Things Journal*, 2018.

[3] C. Xu, K. Wang, P. Li, S. Guo, J. Luo, B. Ye, and M. Guo, "Making big data open in edges: A resource-efficient blockchain-based approach," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 4, pp. 870–882, 2019.

[4] R. T. Tiburski, C. R. Moratelli, S. F. Johann, M. V. Neves, E. de Matos, L. A. Amaral, and F. Hessel, "Lightweight Security Architecture Based on Embedded Virtualization and Trust Mechanisms for IoT Edge Devices," *IEEE Communications Magazine*, vol. 57, no. 2, pp. 67–73, 2019.

[5] O. Novo, "Blockchain meets IoT: An architecture for scalable access management in IoT," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1184–1195, 2018.

[6] Y. Rahulamathavan, R. C. . Phan, M. Rajarajan, S. Misra, and A. Kondoz, "Privacy-preserving blockchain based IoT ecosystem using attribute-based encryption," in *2017 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, Dec 2017, pp. 1–6.

[7] L. Lao, Z. Li, S. Hou, B. Xiao, S. Guo, and Y. Yang, "A Survey of IoT Applications in Blockchain Systems: Architecture, Consensus, and Traffic Modeling," *ACM Computing Surveys (CSUR)*, vol. 53, no. 1, pp. 1–32, 2020.

[8] L. Wu, K. Meng, S. Xu, S. Li, M. Ding, and Y. Suo, "Democratic centralism: A hybrid blockchain architecture and its applications in energy Internet," in *IEEE International Conference on Energy Internet (ICEI)*. IEEE, 2017, pp. 176–181.

[9] L. Wu, X. Du, W. Wang, and B. Lin, "An out-of-band authentication scheme for internet of things using blockchain technology," in *2018 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2018, pp. 769–773.

[10] S. Huh, S. Cho, and S. Kim, "Managing IoT devices using blockchain platform," in *19th International Conference on Advanced Communication Technology (ICACT)*. IEEE, 2017, pp. 464–467.

[11] X. Yue, H. Wang, D. Jin, M. Li, and W. Jiang, "Healthcare data gateways: found healthcare intelligence on blockchain with novel privacy risk control," *Journal of medical systems*, vol. 40, no. 10, p. 218, 2016.

[12] Q. Xia, E. B. Sifah, K. O. Asamoah, J. Gao, X. Du, and M. Guizani, "MeDShare: Trust-less medical data sharing among cloud service providers via blockchain," *IEEE Access*, vol. 5, pp. 14 757–14 767, 2017.

[13] A. Dorri, S. S. Kanhere, and R. Jurdak, "Towards an optimized blockchain for IoT," in *Proceedings of the Second International Conference on Internet-of-Things Design and Implementation*. ACM, 2017, pp. 173–178.

[14] S. D. Angelis, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, and V. Sassone, "PBFT vs proof-of-authority: applying the CAP theorem to permissioned blockchain," in *Italian Conference on Cyber Security*, January 2018.

[15] S.-C. Cha, J.-F. Chen, C. Su, and K.-H. Yeh, "A Blockchain Connected Gateway for BLE-Based Devices in the Internet of Things," *IEEE Access*, vol. 6, pp. 24 639–24 649, 2018.

[16] K. Lei, M. Du, J. Huang, and T. Jin, "Groupchain: Towards a Scalable Public Blockchain in Fog Computing of IoT Services Computing," *IEEE Transactions on Services Computing*, vol. 13, no. 2, pp. 252–262, 2020.

[17] J. Sun, J. Yan, and K. Z. Zhang, "Blockchain-based sharing services: What blockchain technology can contribute to smart cities," *Financial Innovation*, vol. 2, no. 1, p. 26, 2016.

[18] M. Li, J. Weng, A. Yang, W. Lu, Y. Zhang, L. Hou, J.-N. Liu, Y. Xiang, and R. Deng, "CrowdBC: A blockchain-based decentralized framework for crowdsourcing," *IEEE Transactions on Parallel and Distributed Systems*, 2018.

[19] K.-K. R. Choo, Z. Yan, and W. Meng, "Blockchain in Industrial IoT Applications: Security and Privacy Advances, Challenges and Opportunities," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4119–4121, 2020.

[20] I. Mistry, S. Tanwar, S. Tyagi, and N. Kumar, "Blockchain for 5G-enabled IoT for industrial automation: A systematic

review, solutions, and challenges," *Mechanical Systems and Signal Processing*, vol. 135, p. 106382, 2020.
[21] M. Shen, H. Liu, L. Zhu, K. Xu, H. Yu, X. Du, and M. Guizani, "Blockchain-Assisted Secure Device Authentication for Cross-Domain Industrial IoT," *IEEE Journal on Selected Areas in Communications*, 2020.
[22] S. N. Mohanty, K. Ramya, S. S. Rani, D. Gupta, K. Shankar, S. Lakshmanaprabu, and A. Khanna, "An efficient Lightweight integrated Blockchain (ELIB) model for IoT security and privacy," *Future Generation Computer Systems*, vol. 102, pp. 1027–1037, 2020.
[23] Y. Liu, K. Wang, Y. Lin, and W. Xu, "LightChain: A Lightweight Blockchain System for Industrial Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3571–3581, 2019.
[24] Y. Liu, K. Wang, K. Qian, M. Du, and S. Guo, "Tornado: Enabling Blockchain in Heterogeneous Internet of Things through A Space-Structured Approach," *IEEE Internet of Things Journal*, 2019.
[25] T. X. Tran and D. Pompili, "Joint task offloading and resource allocation for multi-server mobile-edge computing networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 856–868, 2018.
[26] L. Wan, D. Eyers, and H. Zhang, "Evaluating the Impact of Network Latency on the Safety of Blockchain Transactions," in *2019 IEEE International Conference on Blockchain (Blockchain)*. IEEE, 2019, pp. 194–201.

Sudip Misra (M'09–SM'11) He is a Professor with the Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur, India. Prior to this, he was associated with Cornell University (USA), Yale University (USA), Nortel Networks (Canada), and the Government of Ontario (Canada). He possesses several years of experience working in academia, government, and private sectors in research, teaching, consulting, project management, architecture, software design, and product engineering roles. His current research interests include wireless ad hoc and sensor networks, Internet of Things (IoT), computer networks, learning systems, and algorithm design for emerging communication networks.

Anandarup Mukherjee He is currently a Senior Research Fellow and Ph.D. Scholar in Engineering at the Department of Computer Science and Engineering at the Indian Institute of Technology, Kharagpur. He finished his M.Tech and B.Tech from West Bengal University of Technology in the years 2012 and 2010, respectively. His research interests include, but are not limited to IoT, networked robots, unmanned aerial vehicle swarms, and enabling deep learning for these platforms for controls and communications.

Arijit Roy He is a Ph.D. scholar at the Indian Institute of Technology, Kharagpur, India. Prior to that, he received an MS (by research) degree and B.Tech degree in Information Technology from the Indian Institute of Technology Kharagpur in 2015 and the West Bengal University of Technology in 2010, respectively. His research works are published in different reputed SCI journals (including IEEE/ACM Transactions) and in many reputed conferences.

Nishant Saurabh He has completed his B.Tech degree in Electronics and Communication Engineering from National Institute of Technology, Patna, Bihar, India in June 2019. His research interest includes a broad range of areas such as Internet of Things, Microprocessors and Microcontrollers, blockchain, VLSI, and others, with the main focus on integrating the blockchain with other technologies and creating a decentralized and secure platform in other domains.

Yogachandran Rahulamathavan He is a lecturer and a program director for MSc Cyber Security and Big Data program at Loughborough University's London Campus in the UK. His research interest is on developing novel security protocols to advance machine learning techniques to solve complex privacy issues in emerging applications e.g., patient's healthcare data sharing, biometric authentication systems, identity management in cloud, etc. Currently, he is coordinating UK-India project (worth of £200k) between Loughborough University London, IIT Kharagpur and City, University of London.

Muttukrishnan Rajarajan He received his BEng and PhD degrees from City University London in 1994 and 1999 respectively. From 1999 he worked at City University London as a Research Fellow. In August 2000 he moved to Logica as a Telecommunication Consultant. After a few years in the industry Raj is now a Professor of Security Engineering. He is also the Programme Director for the Engineering with Management and Entrepreneurship programme. He is a senior member of IEEE, a member of IET and an associate member of the institute of information security professionals (IISP) and a member of Technical Programme Committees for various prestigious conferences. He also sits on the Editorial boards of Springer/ACM Journal on Wireless Networks, Elsevier Journal of Health Policy and Technology and Emerald Journal of Information Management and Computer Security.