NVEO
Natural Volatiles & Essential Oils

Developing a Systematic Blockchain System for Security and Privacy Management in Iot

Dr. K. Sai Manoj

CEO Amrita Sai Institute of Science and Technology / Innogeecks Technologies

Abstract

Presently, BlockChain (BC) gained significant interest because of its undeniable nature and related advantages of security and privacy, BC has the power to resolve the limitations of Internet of Things(IoT) such as data protection and privacy. At the same time, BC has high computation complexity, restricted scalability, high bandwidth overhead and latency that is unsuitable to IoT. In this paper, efficient Lightweight integrated Blockchain (ELIB) model is developed to meet necessitates of IoT. The presented model is deployed in a smart home environment as an important illustration to verify its applicability in various IoT scenarios. The resource constrained resources in a smart home takes the advantages from a centralized manager which generates shared keys to transmit data, process every incoming and outgoing requests. The presented ELIB model generates an overlay network where highly equipped resources can merges to a public BC which verifies dedicated security and privacy. A set of three optimizations are carried out in the presented ELIB model include lightweight consensus algorithm, certificate ess (CC) cryptography and Distributed Throughput Management (DTM) scheme. A detailed simulation takes place under different scenarios in terms of processing time, energy usage and overhead. The ELIB attains a total of 50% saving in processing time on comparing to baseline method with the minimum energy consumption of 0.07mJ. The obtained experimental outcome indicated that the ELIB shows maximum performance under several evaluation parameters.

Keywords: IoT, Blockchain, Security, Privacy, and Certificate less cryptography

1. Introduction

Recently, the field of BlockChain (BC) becomes more familiar which keeps hold of structured peer to peer digital ledger *T*s and distribute the *T* details to all coordinated nodes in the chain [1]. The concept of centralization is eliminated in BC network. The nodes in the network regularly monitor the new *T*s in the network and activate the remaining nodes to participate in the network. The computationally complicated, complicate-to-solve and easy-to-verify puzzle are the various processes managed by the new block in BC and this new block have been enhanced by the new puzzle. The consensus algorithm has the specific computation resources and manages the number of blocks to get activated by the node. The blocks in the network has been restricted to get mined in order to protect the node from adversaries mining of blocks. The new coming node (miners) has to solve the puzzle and this puzzle is not same for all the available nodes. It is randomly generated to all available new miners. The existing Consensus algorithm is been based with the following techniques Proof of Work (POW) [2] or Proof of Stake (POS) [3] is been typically implemented by the Existing BC. The POS technique needs is highly complicated and needs memory resources to solve a cryptography puzzle. In addition, the POW request high computational resources and the encrypted message are communicated between the nodes to protect the nodes against eavesdropping. The encrypted message is decrypted by the Public Keys (PK) and these PK is randomly changed in BC and these are process is continuously updated to protect the nodes from the adversaries. The BC is implemented in a cryptocurrency termed as Bitcoin and the techniques have been broadly distributed in other cryptocurrencies called as altcoins [4]. The BC technology has been explored in non-economical applications like medicinal field, safety to swarm of robots, and so on [5,6]. This paper have designed in such a way that the new BC techniques can resolve the security and privacy issues to the devices linking billions of daily usage devices to the Internet called Internet of Things (IoT). The overview of BC network architecture is shown in Fig. 1. The traditional security and privacy techniques are found to be unsuccessful to IoT because of the issues mentioned below [7]:

- Resource constraints: The various resource-dependent parameters are computation, bandwidth and memory which are limited in the IoT devices and these features are inefficient to fulfill the complicated security issues.
- Centralization: Centralized brokered *T* structured in the current IoT in which every device is monitored, verified, identified and are linked via cloud servers. The scalability issues in Current IoT ecosystem and cannot connect the billions of devices. The above setback will make a Cloud servers to remain obstruct during a failure and interrupt in the whole network.
- Lack of privacy: Current IoT applications postulated a concise content to the Service Providers (SP) for receiving customized services. The traditional privacy preserving techniques depends upon the noisy informative or reviewed data to the data requestor [8].

The BC technology has various advantages and various challenging techniques are presented to resolve the same issues in IoT. The presented techniques of BC are not manageable to develop the IoT applications because of complex consensus algorithms, security overhead, throughput and latency. There are lots of researches done and detailed surveyed in the domain of IoT security and privacy takes place [9–15]. [9] projected a peer-to-peer host identity protocol to secure IoT. The projected Host Identity protocol (HIP) reduces the 40 bytes to a maximum of 25 bytes in the header size of the Low Power Wireless Personal Area Networks (6LowPAN) and these redundant data have been eliminated by minimizing the network overhead as well as removing unnecessary header fields. The researchers also projected a lightweight key distribution approach by the distribution of key among the low resource IoT devices and users [16– 20]. To overcome the resource constraints, the high resource availability device is over placed on the low resource devices. The high resource is computationally lightweight for some applications, eliminating the 6LowPAN and HIP header fields for minimizing the functionalities. The most prominent issue is the scalability approach and is limited to the high resource devices and must be within wireless range of every IoT devices. [10] projected a new reliable and access control method to give authenticated support to build IoT secure against unreliable users and access. The projected approach is based on two authentication authorities namely: (i) Registration Authority (RA), and (ii) Home Registration Authority (HRA).

The devices and the users are designed separately for the reliable and authentication process where the authentication for the devices and users are done separately. The RA process is developed to assist the authentication procedure for the devices. The users are registered with separate techniques HRA. The process of RA is when a user wants to get the information from a specific device. The user sends the request message as the primary message to particular device. Then, the RA checks the user is an authenticated user with HRA and sends the acknowledgment to the required user. Once the authenticated user is recognized by the sender, then the reliable shared keys is communicated between the sender and receiver device. The projected security analysis is the best security model compared to the man-in the middle attack. Fundamentally, each device is integrated with the RA. Then, each and every specific user has a corresponding HRA and these two said techniques are used as the bottleneck for scalability. In the proposed ELIB technique, it is designed as a tiered structure and the whole nodes in the network are managed by a single public BC and manage the nodes in a distributed manner. The LBM manages the overlay nodes and the devices to manage the smart homes in an independent way. The LBM is used specifically for the smart phones. The projected techniques achieve better security attack compared to the existing techniques.

On the other hand, Bitcoin is structured as a decentralized authority providing a higher level of security and privacy to the users. In the year 2013, a new BC techniques called Ethereum had been projected [3] to provide high security and privacy which techniques generate smart contracts with a small fee. Ethereum BC is vividly used in many applications in recent domains like agriculture [11], crowd funding [12], and micro blogging [13]. [14] projected a framework which helps the energy producers for negotiating the selling price with their customer and mark a feasible contract to do the better sale. The proposed framework Distribution System Operator (DSO) entails that the trade is highly confidential and does not allow producer or customer to follow their contract. The double spending had been prevented by a lock key. The broad category of the adversaries is overcome by the projected framework and it is clearly said in the Security analysis. The above architecture endures a low scalability and broadcasts every *T* and block to entire network. The above setbacks are overcome by the ELIB, the techniques overcome by restricting the node count to handle BC. The researchers in [15] projected a BC-based multi-tier architecture to exchange the data from the IoT devices with organizations and people. Though various models have been presented, there is still a need to enhance the performance of the authentication systems. It is also needed to develop a model incurring low overhead, energy consumption and processing time.

This paper presents an efficient Lightweight integrated Blockchain (ELIB) model is developed to meet necessitates of secure IoT. The presented model is deployed in a smart home environment as an important illustration to verify its applicability in various IoT scenarios [21]. The presented model contains two major levels namely smart home and overlay. The presented ELIB model operates in three levels namely (i) consensus algorithm, certificateless cryptography (CC) method and Distributed Throughput Management (DTM) scheme. The consensus algorithm restricts the number of new blocks created by cluster heads (CHs) in a tunable consensus period. For reducing the computational overhead linked with ensuring new blocks which are appended to public BC, ELIB make use of CC method. At the end, the DTM scheme is presented to alter particular system variables in a dynamic way for ensuring the throughput of the public BC is not considerably varies from the *T* load in the network. A detailed simulation takes place under different scenarios and the obtained experimental outcome indicated that the ELIB shows maximum performance under several evaluation parameters.

The remaining structure of the paper is arranged here. Section 2 elaborates the ELIB model in detail. Section 3 discusses the experimental validation and Section 4 draws the highlights of the ELIB model.

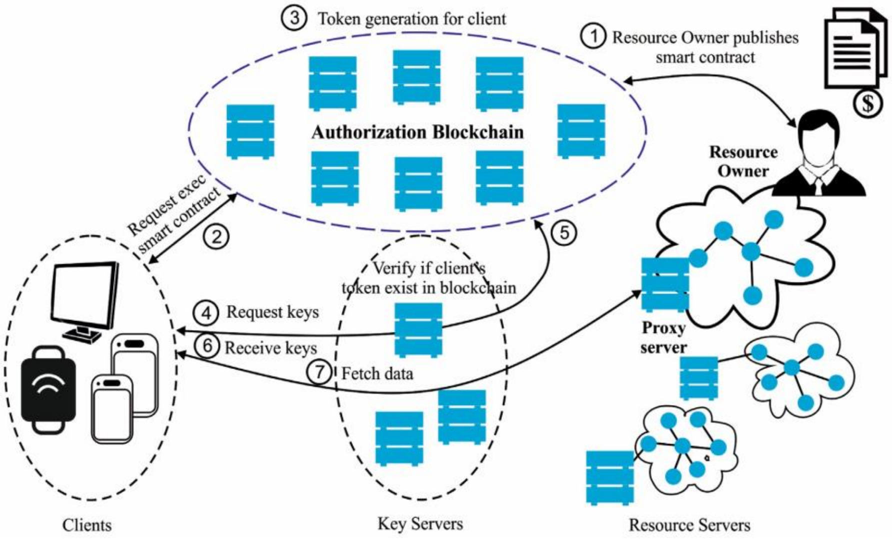

③ Token generation for client
① Resource Owner publishes smart contract
Authorization Blockchain
Resource Owner
Request exec smart contract
②
⑤
Verify if client's token exist in blockchain
④ Request keys
⑥ Receive keys
⑦ Fetch data
Proxy server
Clients
Key Servers
Resource Servers

Fig. 1. Overview of BC Architecture

2. Proposed ELIB Model

The presented ELIB model involves two basic concepts as shown in Fig. 2. The *T* is a fundamental element to exchange control information between any entities. It is also noted that the data flow is entirely different from *T*s. Next, block manager (BM) is another element which has the capability to manage BC. It comprises three processes namely generating, verifying and storing every *T*s and blocks of *T*s. The functionalities of BMs in overlay and smart home tiers are not identical which are discussed in the following subsections.

2.1. Overlay

Here, every individual node in the overlay is called as PK. The nodes utilize a new PL for generating every new *T* for ensuring anonymity. The overlay contains a number of different elements called as overlay nodes, includes a smart home (indicated as Local BM (LBM), mobility entities, Service Provider (SP) servers, and cloud storages (utilized by smart home components to store data). The overlay network considers a significant number of nodes. Hence, for ensuring scalability, it is assumed that the public BC is controlled using a group of the overlay nodes. It is also assumed that the clustering technique is employed for grouping the nodes intro cluster where every cluster elects a Cluster Head (CH). The CH generally performs the action of BC management and called as Overlay Block Mangers (OBMs). In addition, CH processes the inflow and outflow of *T*s which are created from or to their cluster members. A node is chosen as a CH is likely to stay online for an extensive time period and also it should have enough resources to process blocks and *T*s. As the basic processes are carried out by the CHs, ELIB is not affected by IoT device dynamics.

*T*s which are created by the overlay nodes are kept safe by the use of various functions. The *T*s in the overlay undergo classification into (i) single signature *T*s and (ii) multisig *T*s contain the signature of Rr as well as Re. In addition, most of the *T*s in ELIB are multisig and is depicted in Fig. 3. The identifier holds the initial field of the *T* and the next field is occupied by a pointer to the earlier *T* of the similar Rr node. Hence, every *T* generated using a Rr is linked altogether. The next signature is included during the reception of the *T* from the Rr. The 7th field indicates the *T* output and is fixed by the Rr which has a set of three components such as number of *T*s created by the Rr has been received by Re, number of *T*s avoided by Re and hash of the PK that the Rr that is used for subsequent *T*. The initial 2 fields offer past details which are essential to compute the Rr's reputation level. The final field is needed for further authentication of the Rr as the overlay nodes can modify the PK employed to generate every new *T*. The last field in a multisig *T* offers details related to the specific action. An individual signature *T* possess a unique structure, however, it does not exclude the Re PK and signature, metadata, and outputs [0] and [1] since a single overlay node is included. It is noted that multisig and single signature *T*s undergo organization as individual ledgers because the corresponding output is not identical.

In ELIB model, both data as well as *T* flow are kept apart. Hence, in reply to accessing or monitoring the *T*, the Re device transmits the data to the Rr in an individual data packet once it is ensured that the Rr has authorization for accessing data. Likewise, to store *T*, the data generated by the Rr is transmitted definitely from *T*. In contrast to *T*s which broadcast the data, data packets are unicast and could be sent through the optimum routes via overlay network [22].

The overlay *T*s are saved in the public BC which is controlled by the OBMs. Every block in the BC comprises of two major elements such as *T*s and block header. The former element holds the hash of earlier block, block generator ID, and verifier signatures. The hash of the earlier block in the public BC verifies the nature of immutability. When an attacker tries to modify an earlier saved *T*, then the hash of the equivalent block is saved in the subsequent block will no longer be reliable and interpret the attack. The ‘‘block generator ID’’ and ‘‘signatures of the verifiers’’ fields are explained here. The multiple *T*s are linked and undergo processing as a single block. Every block can save a maximum of *T_max* *T*s. The value of T max has influence the BC performance in such a way that with higher *T_max*, many *T*s are saved in an individual block.

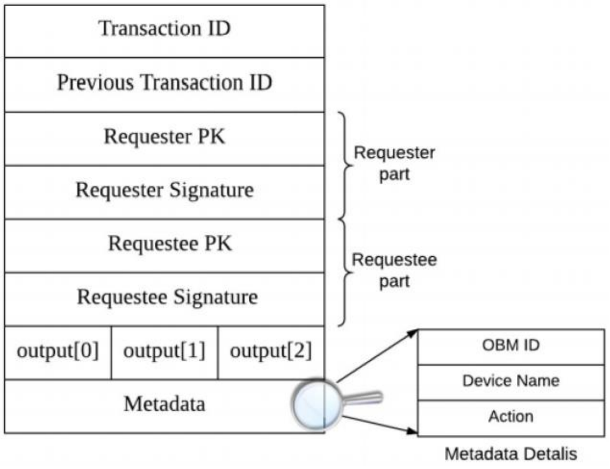


Fig. 2. Process of Storing, Observing and Monitoring *T*s.

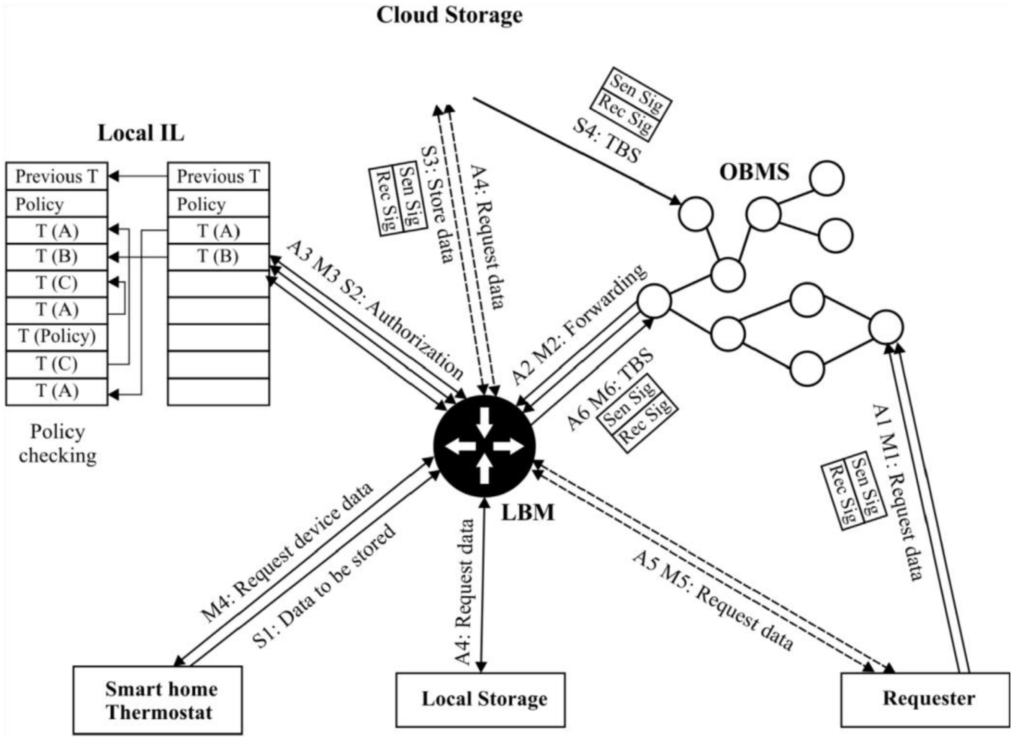


Fig. 3. Multisig *T*.

Once the *OBM* gets a *TY*, it initially verifies whether the Re of this *T* is present in the cluster. The *OBM* has a key list (basically access control list) comprising of Rr/Re PK pairs that denotes the Rrs which allows the transmission of *T*s to particular Res. This key list gets updated through a cluster member for giving permission to other overlay nodes for sending *T*s to it. A Re might fix the Rr value in *OBM* key list as ‘‘*broadcast*’’ that indicates the reception of every *T* which contains its PK as the Re PK. When the Rr and Re of the incoming *TY* match to the entry from the list of keys, then the OBM transmits the *T* to the Re (that lies inside a cluster and hence straightaway linked to the OBM). When the Re in *Y* do not belongs to the cluster present in *OBM*, then the *T* will be broadcasted to every other *OBM*. Every pending *T* is saved in a *T* pool at every OBM. In case of the running pool size is equivalent to T_{max}, then *OBM* initiate the procedure of generating a new block by the use of consensus technique.

2.2. Consensus Algorithm

In ELIB, a time-dependent consensus technique is presented to replace the existing resource-intensive methods like *PoW* and *PoS* which are generally employed in *BC*. The consensus technique ensures that a block generator is arbitrarily chosen between nodes and is restricted to the number of blocks which can be generated. For introducing arbitrary nature between block generators, every *OBM* has to wait for an arbitrary amount of time called as waiting-period before the process of new block creation. As the waiting-period varies for every *OBM, an OBM* can get a fresh block generated by another *OBM* which holds few or every *T*s that presently lies in the pool of *T*s of the *OBM*. At this point, the *OBM* should eliminate the *T*s from its pool since it is saved in *BC* by other *OBM*. It is required to *OBMs* for waiting for an arbitrary time period and also minimizes the waiting period arbitrarily and decreases the number of duplicate blocks which is created concurrently. On the generation of new block is created, it will be broadcasted to another overlay nodes so that it could be included to the *BC*.

For protecting the overlay against a malicious *OBM* might significantly create a massive block count with false *T*s leads to an appending attack, the periodicity with which *OBM* create blocks is limited so that only an individual block can be created over an interval indicated by a consensus period. The default (and maximum) value for consensus-period is 10 min that is identical to the mining duration. A least value of consensus period is identical to double the highest end-to-end delay in the overlay for ensuring that there is enough time to disseminate blocks produced by other *OBMs*.

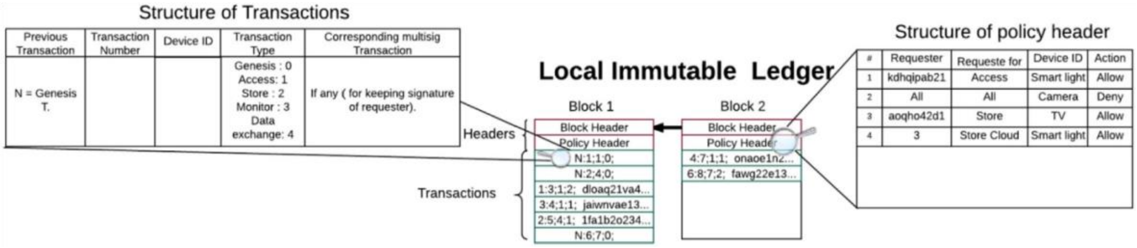


Fig. 4. The Structure of the Local IL.

Each *OBM* monitors the frequency of another *OBMs* which produces blocks. Some of the non-compliant blocks are eliminated and the trust linked with the accountable *OBM* is reduced. For preventing *OBMs* from always maintaining a lower waiting period, the nearby *OBMs* check that an *OBM* creates fresh blocks in the beginning of the waiting-period. When the number of those blocks crosses a threshold, the *OBMs* discards the blocks created by their neighbor depending upon the application.

2.3. CC Method

CC method is applied in the BC-based IoT systems. The public ledger of BC model provides a suitable way of broadcasting any IoT device's public key. In addition, the CC model mainly decreases the repetitiveness provided by classical models and it provides an effective process of authenticating the IoT devices. By the use of CC, verification of IoT devices will be simply done. For instance, when an IoT device post a *T* with the signature using private key SK_A, and includes its public key PK_A and $IDID_A$ to the *T*. A miner can verify that: (1) this *T* is really signed with a private key linked to PK_A, and (2) PKA does belong to ID_A. By this manner, it can be simply ensured that the *T* is generated by the *IoT* device with ID_A.

In CC, a key generation center (KGC) produces a partial private key depending upon the identity of the user. The user makes use of partial private key and its secret value for establishing a private key. As the secret value is known only to the end-user, KGC does not have the ability to determine the private key. The user generates the public key depending upon the secret value and makes it public. Here, a set of five important steps are presented to establish the keys PK_A and SK_A for a user *A*. Step 1: $Setup(1^{\lambda}) \rightarrow (K, MSK)$: Setup function gets the security parameter λ and gives the system parameters *K* and secret master key *MSK*. It is executed using KGC and it can recognize the MSK value.

Step 2: *PSkeyGen*(*K, ID_A, MSK*) → (*PSK_A*): Partial private key (PPK) generation technique receives the system parameters *K*, a user *A's* identity *IDA* ∈ {0, 1}∗, and *MSK*, and gives PPK. It is executed using KGC and the output will be transported to entity

Step 3: *SValGen*(*K, ID_A*) → (*X_A*): Secret value generation algorithm receives the system parameters *K* and user *A's* identity *ID_A*, and provides secret value *X_A*. This technique gets executed by the user and *X_A* is transformed the partial private key to a private key. This technique gets executed by the user.

Step 4: *SKeyGen*(*K, PSK_A, X_A*) → (*SK_A*). Private key generation technique receives the input as system parameters *K*, the PPK *PSK_A* and the secret value *X_A*, and return the private key *SK_A*. This technique gets executed by the user and only this user knows his private key.

Step 5: *PKeyGen*(*K, X_A*) → *PK_A*: The public key generation technique receives the system parameter *K* and secret value *XA* for constructing the public key *PK_A*. This technique gets executed by the user and *PK_A* broadcasted to the public.

2.4. Distributed throughput Management (DTM)

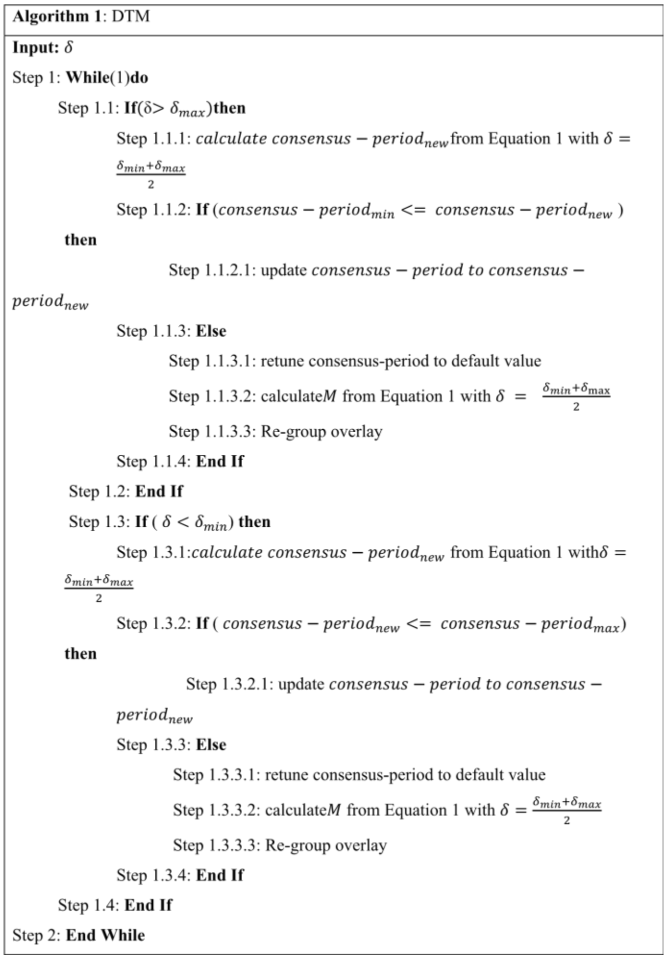
The traditional consensus techniques employed in BC restrict the throughput of BC that is determined as the total *T* count saved in *BC* for every second, as resolving the cryptographic puzzle is highly demanding. For example, Bitcoin *BC* is restricted to a set of seven *T*s per second due to *POW* [23]. For *IoT*, these restrictions are not acceptable as diverse number of interactions is present among different number of nodes. InELIB, a Distributed Throughput Management *(DTM)* scheme is determined to efficiently observe the *BC* exploitation and build suitable modification for ensuring that it still remain in a limited range. During the closure of each consensus-period, every *OBM* determine the utilization (*α*) as the ratio of the number of fresh *T*s created to the number of *T*s appended to the *BC*. It is noted that every *T* and blocks are broadcasted to every OBM, the utilization determined by every OBM which are identical to one another. The main intention of DTM is to verify that *α* stays in a particular desirable range *(α_{min},α_{max})*. A network with a collection of N nodes is assumed which contains a *M* number of *OBMs* and *R* indicates the average rate during the generation of node's *T*s takes placed. The utilization can be determined using Eq. (1):

$$\alpha = \frac{N * R * Consensus - period}{T\_max * M} \qquad (1)$$

Here, Eq. (1) suggested that there exist two different methods through the adjustment of utilization can takes place: (i) modifying the consensus-period that dictate the occurrence of the number of block included in the BC; or (ii) modifying *M*, since every *OBM* generates an individual block inside the consensus period. The next one imposes high overhead since it needs to reconfigure the whole overlay network. Hence, when *α* exceeds *α* max, in the initial instance, *DTM* verifies whether the consensus period could be decreased. At that point, the new value for the consensus period is determined by the use of above equation and considers that *α* is equivalent to the middle point of the specified range *(α_{min},α_{max})* that verifies a stable operating point for the network. In contrast, when the consensus-period could not be decreased, it is needed to recluster using a new M value which also is determined using Eq. (1). This characteristic enables the ELIB model to achieve scalability where an increased number of participating nodes delivers higher throughput.

2.5. Smart Home

Smart home contains a set of different *IoT* devices controlled by the use of a *Local BM*(*LBM*). As the IoT devices are generally limited in resources, the encryption of local *T* takes place by the use of symmetric encryption, for which a shared key is generated among two parties, and utilize lightweight cryptographic hash function. In every smart home, the ELIB controls the *local Immutable Ledger (IL)* that is identical to the format to a BC, and process local as well as overlay *T*s which are produced from smart home. The ELIB model can be linked to the Internet gateway or an individual middlebox *F −secure* [24] that plays an intermediate role among the *IoT* devices and gateway.



```
Algorithm 1: DTM
Input: $\delta$
Step 1: While(1)do
    Step 1.1: If(δ> $\delta_{max}$)then
        Step 1.1.1: $calculate\ consensus - period_{new}$ from Equation 1 with $\delta = \frac{\delta_{min}+\delta_{max}}{2}$
        Step 1.1.2: If ($consensus - period_{min} <= consensus - period_{new}$ ) then
            Step 1.1.2.1: update $consensus - period\ to\ consensus - period_{new}$
        Step 1.1.3: Else
            Step 1.1.3.1: retune consensus-period to default value
            Step 1.1.3.2: calculate$M$ from Equation 1 with $\delta = \frac{\delta_{min}+\delta_{\max}}{2}$
            Step 1.1.3.3: Re-group overlay
        Step 1.1.4: End If
    Step 1.2: End If
    Step 1.3: If ( $\delta < \delta_{min}$) then
        Step 1.3.1:$calculate\ consensus - period_{new}$ from Equation 1 with$\delta = \frac{\delta_{min}+\delta_{max}}{2}$
        Step 1.3.2: If ( $consensus - period_{new} <= consensus - period_{max}$) then
            Step 1.3.2.1: update $consensus - period\ to\ consensus - period_{new}$
        Step 1.3.3: Else
            Step 1.3.3.1: retune consensus-period to default value
            Step 1.3.3.2: calculate$M$ from Equation 1 with $\delta = \frac{\delta_{min}+\delta_{max}}{2}$
            Step 1.3.3.3: Re-group overlay
        Step 1.3.4: End If
    Step 1.4: End If
Step 2: End While
```

The local *IL* save every local *T* and the overlay *T*s for which the Re is the ELIB model. As depicted in Fig. 4, every block in the local IL comprises a block and policy header. The block header manages the hash of the preceding block for ensuring immutable nature identical to the public BC. The policy header is present in the form of an *Access Control List (ACL)* that represents the rules to process local as well as overlay *T*s. As seen in the figure, the policy header contains a set of four parameters. The ‘‘*Requester*’’ indicates the *ID* of the entity which generates the *T*. To receive any incoming overlay *T*s that are mutisig *T*s which can be *PK* of the Rr. The second variable in the policy header represents the permitted action that includes any of the following: *store locally, store cloud, access, monitor, and monitor periodic*. The third variable indicates the target device.

Every *T* is saved in the *T* part of *IL* for auditing. In every *T*, a set of five fields are saved as shown in Fig. 4. Every smart home is linked to a local storage space where a device acts as a backup drive which is utilized by smart home devices for local data storage. It can be linked to the ELIB or it could be an individual device. It is considered that the local storage is sage. It is also assumed that a smart device is enabled to save data to the local storage, ELIB creates a shared key which is utilized by the device for authentication with the storage.

3. Performance Evaluation

This section provides a detailed validation of different view of ELIB. A set of two simulation tools namely Cooja and Network Simulator 3 (NS3) is used for simulation purposes. In simulation, a set of 50 overlay nodes are considered in a network. A set of five Rrs are considered which creates a total of 4 *T*s per second. A set of two main measures used for evaluation namely POW processing time and overhead. The POW processing time indicates the amount of time consumed by an off-the-shelf device for resolving POW and the value should be low for better performance. Next, the overhead indicates the processing time for every transaction from when a transaction is attained till the proper reply is given to the requester. The value of overheads should be low to indicate better results.

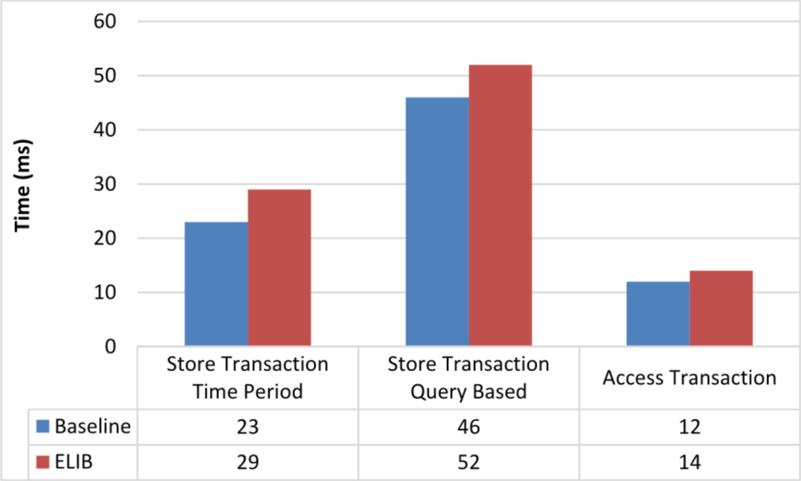
3.1. POW Processing Time

Firstly, a detailed investigation of the time consumption is made by an off-the-shelf device for resolving POW. The POW processing time indicates the amount of time con. The inefficiency of employing traditional BC and PoW is presented in the IoT context. Every block in the Bitcoin BC contains a nonce linked to it. The miner necessitates a searching process for finding the proper nonce so that the entire block fulfills a particular random condition. Especially, it is needed that the SHA-256 hash of the block holds a particular number of leading zeros. An easiest way to identify the proper nonce is the use of brute force. The leading 0's count manages the limitation to solve POW. When the sequence length is long, higher amount of resource and processing time is needed to handle the puzzle. The POS is simulated using C++ to analyze the processing time to solve the puzzle with two limitations. The conventional IoT devices are highly resource limited comparable to laptop, hence, the performance attained are conservative upper bounds that one can expect with IoT devices. A total of 2.3 s is needed to solve POW with 6 leading 0's. By extending the length of 0's to 7, the processing time gets increased to 29.22 min. Presently, Bitcoin with the blocks of 17 0's requires exponentially long time to solve in a general laptop. The experimental values ensured that solving the POW as utilized in Bitcoin leads to more delay on laptop class devices, hence, ensure the design choice of avoiding POW in ELIB model.

3.2. Smart Home Results

It utilizes IPv6 over 6LoWPAN as the fundamental communication protocol, as it is highly applicable to the resource limitations in a smart home environment. A set of three z1 mote sensors are simulated that transmits the data straightly to ELIB at every 10 s. Every simulation will last for 180 s and the outcome is average in the time period. The cloud storage is straightly linked to ELIB for storage purposes. For providing a detailed validation, store and access *T*s are simulated. To store a *T*, a set of 2 diverse and real time traffic flow patterns are employed namely periodic and query based.

In periodic type, every device will regularly save data on the cloud storage. And, in query based, every device will save the data upon receiving a user query, for instance, a tenant or house owner can query a connected security camera for checking whether anyone reached the door. Here, a set of two measures namely time overhead and energy consumption are measured and are shown in Figs. 5 and 6 correspondingly. The time overhead indicates the amount of time needed to process every *T* in LBM and is determined upon the reception of a *T* in LBM till proper reply is transmitted to the Rr. Next, the energy consumption is defined as the amount of energy spent by the LBM to process the *T*s.



	Store Transaction Time Period	Store Transaction Query Based	Access Transaction
🟦 Baseline	23	46	12
🟥 ELIB	29	52	14

Fig. 5. Evaluation of time overhead in the LBM.

Fig. 5 displays the performance of the ELIB model in terms of time overhead. The ELIB models needs more time for packet processing when comparable to the baseline which has the feature of extra encryption and hashing functions. In the poor case for the query-based store *T*, extra overhead incurred by ELIB is 20 ms that is extremely low in absolute terms.

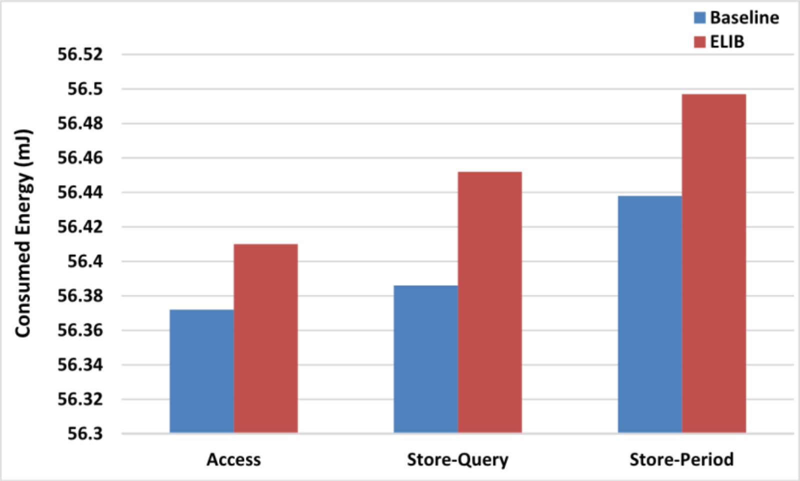


Fig. 6. Evaluation of energy consumption.

An energy consumption analysis is made as shown in Fig. 6. A total energy of 0.07 mJ is spent by the ELIB method.

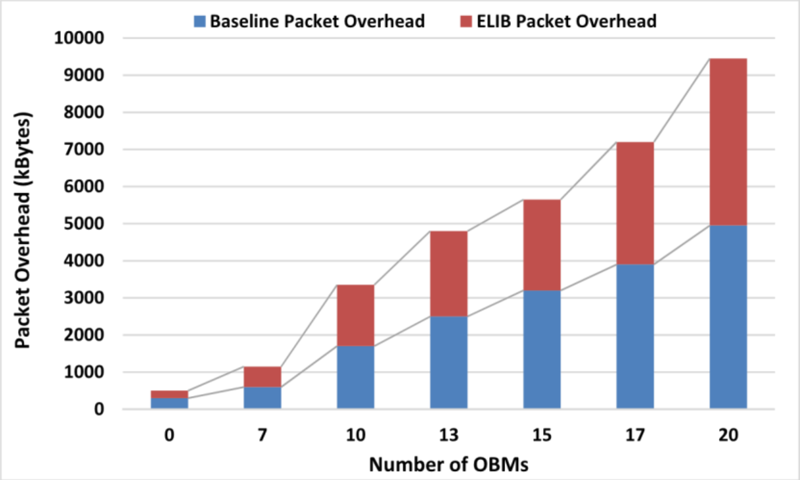


Fig. 7. Assessing the Impact of Packet overhead.

Another experimental analysis is made in terms of packet overhead under varying number of OBMs and the comparative analysis is shown in Fig. 7. As shown in figure, it is clear that a minimum of 200 kB packet overhead is attained by the presented ELIB model whereas a maximum of 300 kB packet overhead is achieved by the baseline method. At the same time, a total of minimum of 550 kB packet overhead is obtained by the presented ELIB model whereas a maximum of 600 kB packet overhead is achieved by the baseline method under the presence of 7 OBMs. In the same way, a total of minimum of 1650 kB packet overhead is attained by the presented ELIB model whereas a maximum of 1700 kB packet overhead is achieved by the baseline method under the presence of 10 OBMs. Likewise, a total of minimum of 4500 kB packet overhead is attained by the presented ELIB model whereas a maximum of 4950 kB packet overhead is achieved by the baseline method under the presence of 20 OBMs.

It is recalled that in Bitcoin every overlay nodes controls the BC in a distributed way. In contrast, the BC management is restricted to choose overlay nodes, i.e., OBMs.

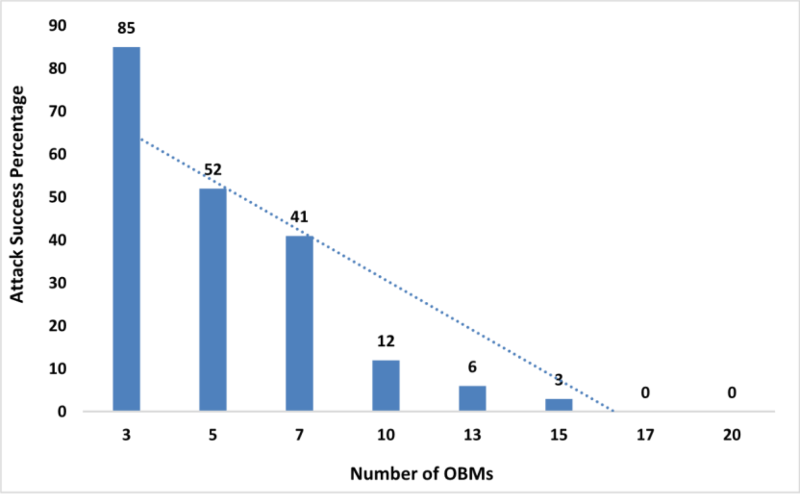


Fig. 8. Impact of the number of OBM on attack success percentage.

It is noted that the baseline method effectively detects the attack as every *T* in a block is verified and the results are depicted in Fig. 8. It is obsolete that the number of successful attacks gets significantly reduced with an increase in number of OBM. It is expected that packet overhead gets increased with an increase in number of OBM. It is known that every *T* in traditional BC should undergo verification by the use of an overlay node. Contrastingly, ELIB make use of CC scheme where the *T* count which should be verified get gradually decreased since OBM creates the trust in one another. Here, the processing time to validate fresh blocks in ELIB with a baseline scheme identical to the traditional BC.

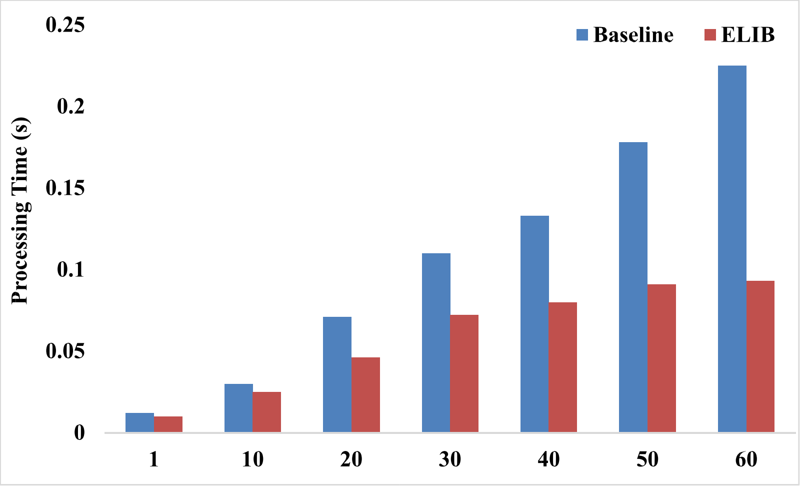


Fig. 9. Average Processing time on OBMs for Validating Fresh Blocks.

The experimentation runs for a total of 3 min and the results are shows for an average of 10 rounds. The total amount of time needed by every OBM for validating a fresh block is shown in Fig. 9. Every another task is disregarded compared to the evaluation of other processes (e.g. ensuring key lists, generating fresh blocks, and so on) compared to the evaluation of fresh blocks since the earlier method is not influenced using the CC method.

Fig. 9 shows the processing time as a function of the number of blocks successfully verified. From this figure, it is shown that the processing time is identical for every method since OBMs have yet to gain trust from one another. But, as the time gets progressed and many blocks are created and ensured, the OBM creates straight trust in one another. As a result, only a fraction of the whole *T* in a fresh block needs to be ensured in ELIB that eliminates the processing time compared to the baseline, where every *T* in the block are ensured. In addition, when the number of blocks gets verified is increased, consequently less number of *T*s is also required to be ensured since the trust in other OBMs continues to rise. In steady state, the ELIB attains a total of 50% saving in processing time on comparing to baseline method with the minimum energy consumption of 0.07 mJ. At the same time, it has a minimum 4500 kB packet overhead under the presence of 20 OBMs. A detailed simulation takes place under different scenarios and the obtained experimental outcome indicated that the ELIB shows maximum performance under several evaluation parameters.

4. Conclusion

This paper has focused on BC, an efficient technique which offers security and privacy in IoT, its application in the IoT context offers diverse considerable issues includes computation complexity, bandwidth, delay and overhead. This paper has presented an ELIB model is developed to meet necessitates of security and privacy in IoT. The presented model is deployed in a smart home environment as an important illustration to verify its applicability in various IoT scenarios. The presented model contains two major levels namely smart home and overlay. The presented ELIB model operates in three levels namely consensus algorithm, CC model and DTM scheme. A detailed simulation takes place under different scenarios. In steady state, the ELIB attains a total of 50% saving in processing time on comparing to baseline method with the minimum energy consumption of 0.07 mJ. At the same time, it has a minimum 4500 kB packet overhead under the presence of 20 OBMs. The obtained experimental outcome indicated that the ELIB shows maximum performance under several evaluation parameters. In future, the proposed work can be improved with an intention to minimize the energy consumption and undergo deployment in diverse applications.

Reference

Kosba, Ahmed, et al. "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts." *2016 IEEE symposium on security and privacy (SP)*. IEEE, 2016.

Vukolić, Marko. "The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication." *International workshop on open problems in network security*. Springer, Cham, 2015.

Wood, Gavin. "Ethereum: A secure decentralised generalised transaction ledger." *Ethereum project yellow paper* 151.2014 (2014): 1-32.

Mohanty, Sachi Nandan, et al. "An efficient Lightweight integrated Blockchain (ELIB) model for IoT security and privacy." *Future Generation Computer Systems* 102 (2020): 1027-1037.

Yue, Xiao, et al. "Healthcare data gateways: found healthcare intelligence on blockchain with novel privacy risk control." *Journal of medical systems* 40.10 (2016): 218.

Ferrer, Eduardo Castelló. "The blockchain: a new framework for robotic swarm systems." *Proceedings of the future technologies conference*. Springer, Cham, 2018.

Ferrer, Eduardo Castelló. "The blockchain: a new framework for robotic swarm systems." *Proceedings of the future technologies conference*. Springer, Cham, 2018.

De Montjoye, Yves-Alexandre, et al. "openpds: Protecting the privacy of metadata through safeanswers." *PloS one* 9.7 (2014): e98790.

Dorri, Ali, Salil S. Kanhere, and Raja Jurdak. "Towards an optimized blockchain for IoT." *2017 IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation (IoTDI)*. IEEE, 2017.

Dorri, Ali, Salil S. Kanhere, and Raja Jurdak. "Blockchain in internet of things: challenges and solutions." *arXiv preprint arXiv:1608.05187* (2016).

Harari, Gabriella M., et al. "Using smartphones to collect behavioral data in psychological science: Opportunities, practical considerations, and challenges." *Perspectives on Psychological Science* 11.6 (2016): 838-854.

Haus, Michael, et al. "Security and privacy in device-to-device (D2D) communication: A review." *IEEE Communications Surveys & Tutorials* 19.2 (2017): 1054-1079.

Bertino, Elisa, and Elena Ferrari. "Big data security and privacy." *A Comprehensive Guide Through the Italian Database Research Over the Last 25 Years*. Springer, Cham, 2018. 425-439.

Chaudhry, Amir, et al. "Personal data: thinking inside the box." (2015).

Saramäki, Jari, and Esteban Moro. "From seconds to months: an overview of multi-scale dynamics of mobile telephone calls." *The European Physical Journal B* 88.6 (2015): 164.

Haddadi, Hamed, et al. "Personal data: Thinking inside the box." *arXiv preprint arXiv:1501.04737* (2015).

Sahraoui, Somia, and Azeddine Bilami. "Compressed and distributed host identity protocol for end-to-end security in the IoT." *2014 International Conference on Next Generation Networks and Services (NGNS)*. IEEE, 2014.

Liu, Jing, Yang Xiao, and CL Philip Chen. "Authentication and access control in the internet of things." *2012 32nd International Conference on Distributed Computing Systems Workshops*. IEEE, 2012.

Liu, Jing, Yang Xiao, and CL Philip Chen. "Internet of things' authentication and access control." *International Journal of Security and Networks* 7.4 (2012): 228-241.

Mohanty, Sachi Nandan, et al. "An efficient Lightweight integrated Blockchain (ELIB) model for IoT security and privacy." *Future Generation Computer Systems* 102 (2020): 1027-1037.

Aitzhan, Nurzhan Zhumabekuly, and Davor Svetinovic. "Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams." *IEEE Transactions on Dependable and Secure Computing* 15.5 (2016): 840-852.

Kang, Jiawen, et al. "Enabling localized peer-to-peer electricity trading among plug-in hybrid electric vehicles using consortium blockchains." *IEEE Transactions on Industrial Informatics* 13.6 (2017): 3154-3164.

Mengelkamp, Esther, et al. "A blockchain-based smart grid: towards sustainable local energy markets." *Computer Science-Research and Development* 33.1-2 (2018): 207-214.

Risius, Marten, and Kai Spohrer. "A blockchain research framework." *Business & Information Systems Engineering* 59.6 (2017): 385-409.

Chavan, Amrita B., and K. Rajeswari. "The design and developement of decentralized digilocker using blockchain." *International Journal of Computer Science Engineering and Information Technology Research (IJCSEITR)* 9.2 (2019): 29-36.

Miraz, Mahadi Hasan, Mohammad Tariq Hasan, and Farhana Rahman Sumi. "Understanding, Supervision, Strategy and Acceptance Effect into the Blockchain Employment in Malaysia." *International Journal of Mechanical and Production Engineering Research and Development (IJMPERD)* 10.3 (2020): 8339-8360.

Miraz, Mahadi Hasan, Mohammad Tariq Hasan, and Mofijul Hoq Masum. "Factors Affecting Consumers Intention to Use Blockchain-Based Services (Bbs) in the Hotel Industry." *International Journal of Mechanical and Production Engineering Research and Development (IJMPERD)* 10 (2020): 8891-8902.

Miraz, Mahadi Hasan, Mohammad Tariq Hasan, and Farhana Rahman Sumi. "The Innovation of Blockchain Transparency & Traceability in Logistic Food Chain." *International Journal of Mechanical and Production Engineering Research and Development (IJMPERD)* 10.3 (2020): 9155-9170.

Satyanarayana, K. N. V., et al. "Mobile app &Iot based smart weather station." *International Journal of Electronics, Communication and Instrumentation Engineering Research and Development (IJECIERD)* 7.4 (2017): 7-14

H. A. R. S. H. A. D., and R. I. J. H. I. Dey. "A Heartbeat Detection Method Based on Iot and Monitoring System Using Arduino Uno And Thing-Speak." *International Journal of Electronics, Communication & Instrumentation Engineering Research and Development (IJECIERD)* 8.3 (2018): 11-16

Dr. K. Sai Manoj, CEO of Amrita Sai Institute of Science and Technology / Innogeecks Technologies has more than 12 years of experience in financial services, IT Services and education domain. He is doing active research pointing to the industry related problems on Cloud Computing, Cloud Security, Ethical Hacking, Blockchain (DLT) and Artificial Intelligence. He was awarded a Doctor of Science degree in Cyber Security & Cloud Computing. He obtained a PhD Degree in Cloud Computing,M.Tech, in Information Technology. He published research articles in various scientific journals like Scopus, Web of Science and also in various UGC approved journals with Thomson Reuter id. Also, he presented innovative articles at high Standard IEEE and Springer Based Conferences. He has various professional certifications like Microsoft Certified Technology Specialist (MCTS), CEHv9 & v11, ECSA, CHFI, Chartered Engineer (C.Eng.,) from IEI, Paul Harris Fellow recognition by Rotary International and Outstanding Industry and Academic Contributor award from ASSOCHAM.