*Computers, Materials & Continua*
DOI: 10.32604/cmc.2022.029637
***Article***

Tech Science Press

# Fast and Efficient Security Scheme for Blockchain-Based IoT Networks

**K. A. Fasila*** **and Sheena Mathew**

School of Engineering, Cochin University of Science and Technology, Kerala, India
*Corresponding Author: K. A. Fasila. Email: fasila7@gmail.com
Received: 07 March 2022; Accepted: 12 April 2022

**Abstract:** Internet of Things (IoT) has become widely used nowadays and tremendous increase in the number of users raises its security requirements as well. The constraints on resources such as low computational capabilities and power requirements demand lightweight cryptosystems. Conventional algorithms are not applicable in IoT network communications because of the constraints mentioned above. In this work, a novel and efficient scheme for providing security in IoT applications is introduced. The scheme proposes how security can be enhanced in a distributed IoT application by providing multilevel protection and dynamic key generation in the data uploading and transfer phases. Existing works rely on a single key for communication between sensing device and the attached gateway node. In proposed scheme, this session key is updated after each session and this is done by applying principles of cellular automata. The proposed system provides multilevel security by using incomparable benefits of blockchain, dynamic key and random number generation based on cellular automata. The same was implemented and tested with the widely known security protocol verification tool called Automated Validation of Internet Security Protocols and Applications (AVISPA). Results show that the scheme is secure against various attacks. The proposed scheme has been compared with related schemes and the result analysis shows that the new scheme is fast and efficient also.

**Keywords:** Cellular automata based key generation; dynamic key generation; IoT security; No-share key Exchange; blockchain for IoT; mutual authentication

## 1 Introduction

Internet of things (IoT) [1] was introduced by Kevin Ashton in 1999. IoT is a collection of heterogeneous interconnected things [2]. Uniquely identifiable devices involved in IoT networks can communicate with each other. They can be connected to the network with the help of gateway nodes through Internet Protocol [3]. With widely increased usage of IoT applications, attacks on IoT systems also increased, and this leads to crucial security requirements. Considering the power requirements and constrained resource capabilities, conventional algorithms are not always suitable for providing security in IoT networks. With the development in the fields of internet and ubiquitous

computing, many applications or services have been provided by the service providers to the users. Such applications will contain private information of users and hence, all security objectives become very relevant. Data sensed by the IoT devices will be uploaded to cloud storage and these can be accessed by authorized users anywhere anytime. Cloud services are of 2 types- public and private. Public platforms are open to all kinds of users whereas, private cloud platforms are accessible only to authorized users. So, proper authentication of users must be ensured to permit access to information stored in cloud. For this, several techniques like authentication protocols were proposed. In the network model of an IoT application, there are 3 levels. The first level consists of IoT devices such as sensors and actuators and these devices have features like low memory, low battery life, less computational and processing capabilities. They are connected to a gateway node in the second level, and finally, gateway nodes are connected to the cloud storage in the third level. This is the network model of a conventional IoT application.

In case of distributed IoT applications, there may be several such gateway nodes and sub networks, each with the same architecture. It is highly beneficial if the security of participating entities in a distributed environment can be assured. For ensuring the authentication of nodes and integrity of data, blockchain can be used. Blockchain [4] is an upcoming paradigm with distributed ledger technology as its backbone. Applications of blockchain are not restricted to crypto-currencies. Blockchain has the capability to provide a 'tamper-proof' ledger of transactions and this will help to maintain integrity of data. Leveraging blockchain technologies to IoT will definitely improve the security aspects of the system. Authentication credentials of involved entities are strictly verified in the blockchain with the help of cryptographic protocols. Hence, adopting blockchain technology for implementing distributed IoT applications will strengthen the security in terms of integrity, authentication, access control, etc.

The works given in the literature review section proposed several security schemes. Some of these cover encryption only while some others focus on key management, authentication, etc. Complete schemes that ensure security goals in all the phases of IoT network communication are comparatively a few. Here, a new scheme, that considers security in all the phases, is proposed. Highlights of the work are the following.

- The work proposes a lightweight security scheme, which ensures security of data.
- Gateway nodes participating in the communication will become blockchain members, and this ensures the authentication.
- Communication between a gateway node and its connected constrained devices is always protected by a dynamic group key.
- Provides three levels of protection for the data. Three levels are encryption with a master key, encryption with the dynamic group key, and access control by attribute-based encryption. Last one also refers to authenticated encryption.
- Key management does not involve a central authority. Hence, it avoids several issues caused by a centralized entity such as bottleneck and center point of failure.
- Constrained nodes have to store less information with small number of keys.
- Constrained nodes do not have to perform any asymmetric encryption or authentication schemes.
- Blockchain-based data transfer ensures message integrity.
- Scheme does not involve any costly computations since the operations involved are lighter compared to other schemes.
- The proposed scheme is resilient to various types of attacks and proves to be safe after security analysis.

- This paper is arranged as follows. Next section discusses a few related works in similar systems. This is followed by the detailed explanation of proposed model, implementation and results, verification with the formal analysis tool Automated Validation of Internet Security Protocols and Applications (AVISPA) in further sections.

## 2 Related Works

Several research works were done in IoT security and these include authentication protocols, key management schemes, and lightweight encryption methods and so on. Wolf et al. [5] proposed a new threat model and also discussed various types of issues faced by IoT systems and Cyber-Physical Systems. Relevance of fog nodes was discussed in [6]. Possible attacks like physical attacks, security protocol attacks, and application security attacks on IoT systems were studied in several research works [7–12]. Proper security measures must be developed and used, to provide protection from all such attacks. Since, IoT systems are based on low power devices with resource limitations, lightweight methods should be designed. Elliptic Curve Cryptography based authentication protocols were proposed in [13,14]. The authentication protocol introduced in [15] is applicable to heterogeneous wireless sensor networks. But, all these are prone to attacks such as password guessing, smart card theft, and node impersonation. After all, these protocols are not concerned about the mutual authentication. A user authentication and key establishment technique for heterogeneous wireless sensor networks was given in [16]. This was proved to be insecure [17] to attacks like password guessing, smart card stealing, node capturing, and replay attack.

The device authentication protocol proposed in [18] was based on public and private key pairs. However, the execution of asymmetric computations is not suitable for constrained devices, as they involve more computational overhead and complexity. A two factor authentication scheme was proposed in [19], in which mutual authentication between the device and a server was explained well. The steps involved fuzzy extractor generation and this created computational overhead. Another important aspect in IoT security is the Group Key Management. An approach for proper group key management was proposed in [20] and this was based on the use of hash functions and lightweight cryptographic operations. Lightweight operations such as bitwise-XOR and one way hash functions are suitable for constrained devices.

Encryption algorithms can be symmetric or asymmetric. Advanced Encryption Standard (AES) is proved to be one of the strongest symmetric algorithms. Several AES-based lightweight encryption schemes [21–25] were introduced for IoT security. However, these methods did not handle the authentication of communicating entities. Another interesting scheme for IoT security was based on attribute-based encryption [26,27] and this is capable of achieving multiple security goals such as access control and data confidentiality simultaneously. A lightweight cipher technique based on Cellular Automata (CA) was proposed in [28]. Principles of CA were introduced by Neumann [29] and Ulam [30], and these concepts could be used to model complex systems. CA concepts were used to conduct researches and studies regarding biological processes such as self-reproduction and spread of diseases [31]. The lightweight algorithms based on CA are stronger because of the feature of rule vector selection. But, most of the schemes lack an efficient key management scheme.

As it was mentioned previously, CA consists of a grid like structure. In a one-dimensional CA [31], the cell before the leftmost cell and the cell after the rightmost cell, are assumed to hold the value zero. Value of the current cell, $C_i$ at time $t + 1$ is decided by the rule number and the value of that cell at time t, value in the left cell and the right cell. This can be represented as follows.

$$C_i t + 1 = Ri\left(C_{i-1}t, C_i t, C_{i+1}t\right), \tag{1}$$

where Ri is the rule number.

Sample rule 60 for a rule vector of length eight is given below in Fig. 1. All CA rules can be represented as mathematical functions [32]. The computations involved are lightweight operations like XOR and complement. For example, Rule 60 can be represented as follows.

$$C_i t + 1 = C_{i-1} t \oplus C_i t \tag{2}$$

Secure distributed IoT application can be implemented with the help of blockchain [33]. Even though blockchain was designed for crypto-currency applications, its scope is far beyond that.

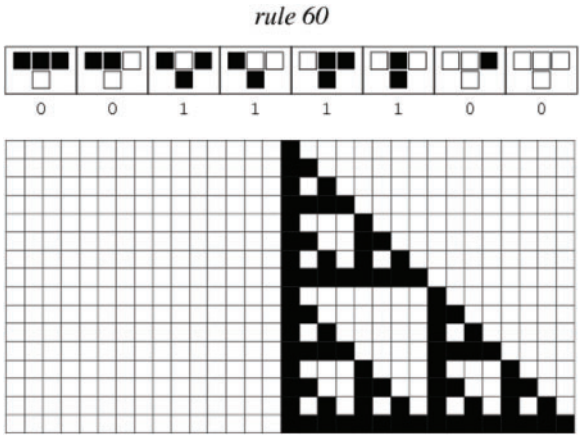


**Figure 1:** Rule 60 of CA

Blockchain is a distributed ledger of all occurring transactions. It was initially proposed for ensuring security of transactions with crypto-currencies. Blockchain architecture contains a set of blocks in which one will be designated as the root/genesis node. This genesis node will keep the history of all transactions occurred so far. Each block contains a proof of work, a link to its previous block and the relevant transaction data. In a blockchain-based technology, the end node performs the computation, which is treated as a transaction and signs it with his private key. Then, the user broadcasts this transaction to other members in peers. All nodes in peer will validate this transaction based on some criteria. The validated transaction will be embedded to a block by the designated root node, called miner, and then appended to the block chain only after a consensus agreement is fulfilled. Basic structure of a blockchain is given in Fig. 2.

A user in blockchain signs the transaction with private key and sends it to the associated peers. Peer nodes are responsible to validate this transaction. Miner node packs the validate transaction into a block. Block will contain hash and timestamp description and this will be used to ensure the message integrity. The hash value inside a block depends on the previous block and hence, if a block contents get changed, it is easily identifiable and can be discarded. If the verification is completed successfully, the block will be added to the blockchain. Thus, it is clear that only properly validated user can modify a data item in an authorized manner.

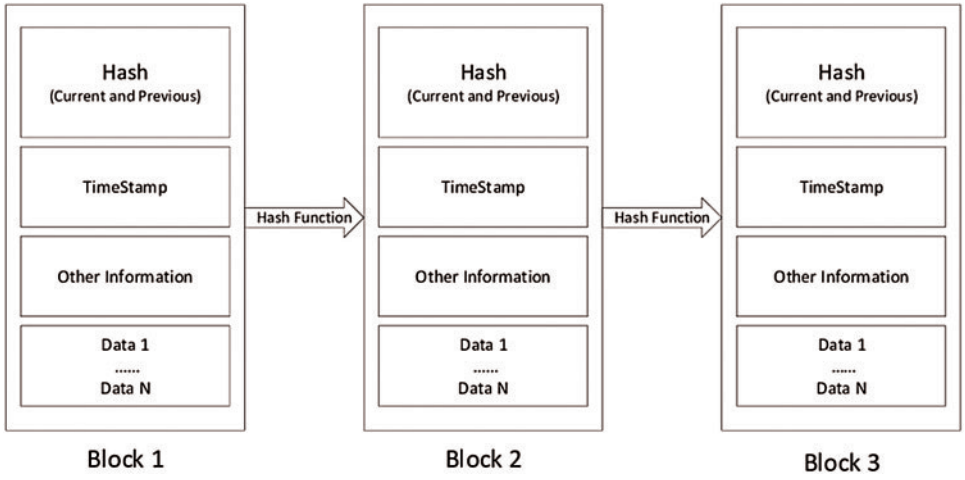


**Figure 2:** Basic structure of blockchain

There are three types of blockchain networks-private, public and consortium. In a public blockchain, there are no restrictions to join, whereas private blockchain is more centralized. In a private blockchain, all members will not have equal rights. While forming blockchain with the gateway nodes in an IoT network, it will ensure data integrity and user authentication of participating nodes. The computational power of IoT gateway nodes is more compared to sensor level devices, but less compared to the cloud servers. When implemented as a blockchain participant, gateways can take either the role of a full node in which each gateway will be performing all functionalities, or they can be implemented as a thin client [34]. In the proposed algorithm, blockchain concept is used for minimal functionalities only. Hence, gateway nodes can be selected to form blockchain networks. These nodes are capable enough to control access to a data item and to ensure data integrity. Gateway devices are components in conventional IoT architecture. So, no additional infrastructure is required to form blockchain with these gateway nodes.

Several IoT-based applications have been already developed using blockchain. A few of them are given in [34–38]. All these works show that using blockchain in IoT applications improves security. In some of these, the extra space in memory of participating nodes is used for storage of data. Some other blockchain-based algorithms have proposed the dissemination of supporting functionalities to blockchain, rather than giving the entire control to cloud storage. Blockchain has been used to develop IoT applications in several use cases including healthcare, vehicular networks, energy management, agriculture etc. Open source platform called Hyperledger fabric [39] can be used for implementing blockchain-based applications.

## 3 Proposed Method

In conventional IoT applications, IoT devices form the basic layer. They can join or leave at any time depending on the application, and they are registered under a gateway node. These constrained nodes are responsible for sensing and data acquisition. The gateway nodes in middle layer provide data aggregation and upload to the cloud storage. Distributed IoT applications can be implemented in a hybrid manner. The term 'hybrid' is used in the sense that, the system can be built up with the help of a blockchain of gateway nodes and simultaneously, cloud storage can be used for data storage. Gateway nodes are included as participants in blockchain network. These nodes are selected because of the

relatively higher resource capabilities, compared to IoT nodes like sensors. The network architecture followed in this proposed method is given in Fig. 3.

When a new member gateway node joins the blockchain, user credentials are allocated and these will be used for ensuring the authentication of that entity. Also, if any gateway node in the system tries to modify data that have been stored already, it is not possible without the consent from all other member nodes. The potential benefits of using blockchain in distributed IoT applications can be summarized as follows:

- Authentication of all gateway nodes will be verified with the support of cryptographic suite provided by the blockchain platform, here it is Hyperledger Fabric.
- The secure communication between gateway and server is always ensured by the blockchain with the help of its default security techniques like cryptographic credentials, hash values and timestamps.
- Data uploaded by a gateway node cannot be tampered by an untrusted entity. This ensures data integrity.

It can be concluded that, once the data are uploaded to the storage by gateway nodes, blockchain paradigm helps to build a strong security foundation. Hence, the major concern in proposed work is ensuring the security of data in the collection and uploading phases of a blockchain-supported IoT application.

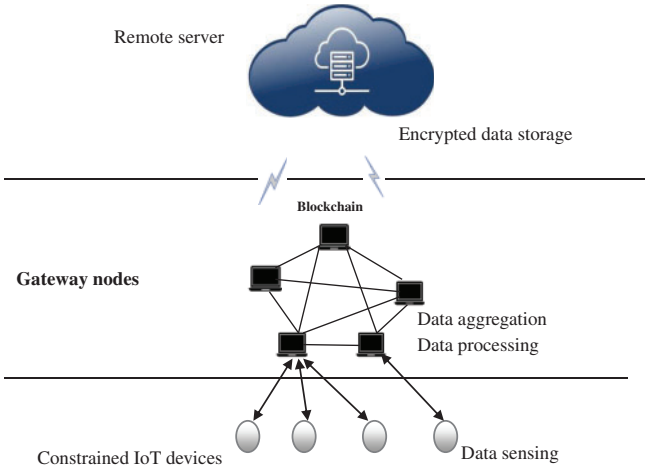


**Figure 3:** Network model in proposed system

Major assumptions are:

- Device registration is completed by the corresponding gateway node.
- Constrained nodes will perform symmetric encryptions and lightweight operations that include only XOR, multiplications, etc. to avoid complicated computations.
- For encryption, an efficient lightweight symmetric encryption algorithm can be used. For implementation purpose, AES-128 is used.
- Gateway nodes and server are able to perform both symmetric as well as asymmetric encryption.
- In the context of blockchain, gateway nodes are also referred as organizations.

Different phases in the proposed scheme are as follows.

1) Registration Phase:

Devices are registered successfully under a particular gateway node and several gateway nodes are forming the blockchain. During registration phase, when a new device is added successfully under a gateway node, a master key will be generated. Similarly, gateway node has a shared key, $K_{GA}$ for secure communication between the device and gateway.

2) Blockchain Formation Phase:

Gateway nodes, which are also known as organizations, construct the blockchain. Distributed ledger is shared through a channel, to which the organizations are connected. Smart contracts are written and installed to the channels and this will be instantiated by a blockchain participant. All the required cryptographic materials and certificates are generated when a new member is added to the blockchain. These include a private key-public key pair and the same can be used to ensure user authentication in all upcoming transactions. Access control to the cloud stored information is restricted by the blockchain entities with the help of attribute-based encryption. Key will be shared to the requesting node, only if the attributes specified by the key-owner matches with that of requesting device. When any node tries to modify the stored data, it has to be approved by all the participating entities. Then only, the change is considered as a legitimate modification and this ensures integrity of data. Hence, authentication, data integrity and access control will be handled by the blockchain.

3) Key management phase:

This is the strongest part of the proposed system that provides multiple levels of security. When the device is registered under a gateway node, it generates a master key, $K_M$. This is the symmetric key used for first level encryption at the device. Since it is a symmetric key, it should be shared with the gateway node. Similarly, the session key used for second level of encryption will be generated by the gateway node. This key is denoted as $K_{GA}$ and it is also a symmetric one.

- Master key exchange: Master key is exchanged between two nodes by using No-Share Key Exchange (NSKE) algorithm [40]. According to this algorithm, the master key will be reformed to a $4 \times 4$ matrix and with only 4 handshaking steps, the key is exchanged with the intended recipient. The computations involved are only matrix multiplications and matrix inverse calculations. The four message exchanges will be encrypted by using the recipient's public key. The request sent from a gateway node will be verified by the blockchain. Access control is granted based on attribute-based encryption.
- Dynamic Group key($K_{GA}$) generation: The initial value of $K_{GA}$ will be received by the device when it is registered under the gateway node, G. Both the device and the gateway node will have the set of CA rules preloaded in memory. For improving security, this group key is generated dynamically for every session. This step relies on the previous value of $K_{GA}$.

Let the initial value obtained by the new device is $K_{GA}{}^0$ (this means, value of $K_{GA}$ at time, $t = 0$). The value of $K_{GA}$ at time, $t = i$ can be computed from previous key value, $K_{GA}{}^{i-1}$ and a selected CA rule number. CA rules are either reversible or irreversible. Since it is not necessary to retrieve the previous value of $K_{GA}$, it is not mandatory to use reversible CA rule. There are 255 CA rules and these can be used to compute next state from a given state. If the current value of the group key is given as input and a CA rule is selected randomly, it will be applied to that value and output will be a new $K_{GA}$. This transition is shown in Fig. 4.

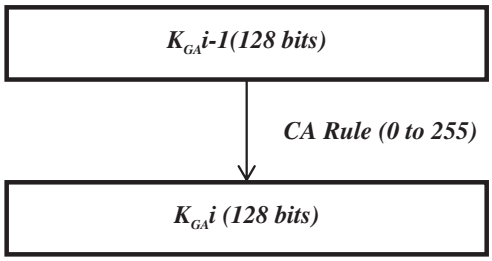


**Figure 4:** Transition of $K_{GA}^{i-1}$ to $K_{GA}^{i}$

4) Mutual authentication phase:

When a device wants to begin a communication session with the gateway node, it must undergo a mutual authentication phase, after which the authentication of device to gateway node, G and the authentication of gateway to device will be completed. After this phase, for every session, device selects a random number, RN and exchanges it with the gateway node. This consists of following steps.

a. The device (say, A) sends a message, "Message_1" encrypted with the current group key $K_{GA}^{i}$, to the gateway node as shown below.

Message_1 = E({IDa, IDg, M1, M2, TSi}, $K_{GA}^{i}$)

Where, E is the encryption algorithm

$K_{GA}^{i}$ is the $i^{th}$ group key

IDa is the unique identity of device, A.

IDg is the identity of gateway node, G.

M1 = H(IDa || TSi) $\oplus$ Na

M2 = H(IDa || IDg || TSi, Na)

H is the hash function commonly agreed between A and G.

TSi is the Timestamp value when A sends the Message_1. Na is the nonce value generated randomly by the device, A. '$\oplus$' denotes the XOR operator and '||' denotes concatenation.

b. On receiving this message at time, TSi', G decrypts the message with $K_{GA}$ and checks whether |TSi' - TS| > T where T represents the maximum permitted transmission delay between the nodes. If the value is greater than T, the message is discarded.

Otherwise, G proceeds to compute the value of Na from the known values and received M1 component. This is done as follows. G retrieves the component M1 from the message.

It computes Na' as follows..

Na' = M1 $\oplus$ H(IDa || TSi). Since $\oplus$ operator is complementary, the result will be the nonce produced by A, if not modified by an adversary. This calculated nonce value is used to compute M2' by using the following equation,

M2'= H(IDa || IDg || TSi, Na')

Then, G checks whether M2'= M2. If both are same, this means no attack occurred.

c. If this verification step is successful, in next step, G sends a message, "Message_2" to A as follows.

Message_2 = E({IDg,IDa,M3,M4,Ng,TSj}, $K_{GA}$)

Where, IDg, IDa, Na, H, $\oplus$ and $\|$ are same as that of Message_1 components.

$M3 = H(IDg \| TSj) \oplus Ng$

$M4 = H(IDg \| IDa \| TSj \| Ng \| Na)$

TSj is the timestamp at which G sends Message_2 to A. Ng is the nonce generated randomly by the gateway node, G.

d. On receiving this, A decrypts the message with $K_{GA}$ and checks whether the received timestamp Tj' differs from the actual timestamp Tj more than the specified delay T. i.e., if $|TSj' - TSj| > T$, Message_2 is discarded. Otherwise, A computes Ng from the known values of IDg and TSj as given below.

$Ng' = M3 \oplus H(IDg \| TSj)$

Using this computed nonce, Ng' A computes $M4' = H(IDg \| IDa \| TSj \| Ng' \| Na)$. Then A checks whether $M4 = M4'$. If this comparison returns TRUE, that means no adversary affected this transmission.

To summarize these, steps a and b proves the authentication of A to G, and steps c and d verifies the authentication of G to A.

If all these steps are executed successfully, mutual authentication phase is considered to be successful and after this, the device selects the CA rule randomly. This rule number is represented as RN. Device shares the value of RN with the authenticated gateway node, G. When G receives this final message, A deletes the current value of $K_{GA}$ from its local memory and computes the new value. To conclude, device A will have IDa, IDg, $K_M$, $K_{GA}^{i+1}$, where $K_{GA}^{i+1} = RN(K_{GA}^{i})$ and gateway node, G will have IDg, IDa, $K_{GA}^{i}$, RN finally in the memory.

5) Data encryption and uploading phase:

After successful completion of registration, mutual authentication phases and dynamic computation of new group key, $K_{GA}^{i+1}$, data item to be uploaded is encrypted in 2 levels. Flowchart with device A, is given in Fig. 5 and steps are given below.

a) A computes the message, $X1 = E(\{X\}, K_M)$, where X is the data to be uploaded, E is the encryption algorithm and $K_M$ is the master key. This first level encryption is device specific. Each device will have its own master key.
b) A applies second level encryption on X1 and gets $Y = E(X1, K_{GA}^{i+1})$
c) A sends the message Y to gateway node, G.

6) Data exchange:

Gateway node G receives the message Y and performs the following steps.

a) G decrypts it with the computed $K_{GA}^{i+1}$ value.
b) G gets the value of $X1 = D(Y, K_{GA}^{i+1})$.
c) G encrypts the message X1 with the public key, $K_{BU}$ of recipient node, B and uploads it to the server node.

Finally, the message can be retrieved by the node, B after decryption with its private key, $K_{BR}$. The master key, $K_M$ can be shared with the node, B with NSKE algorithm.

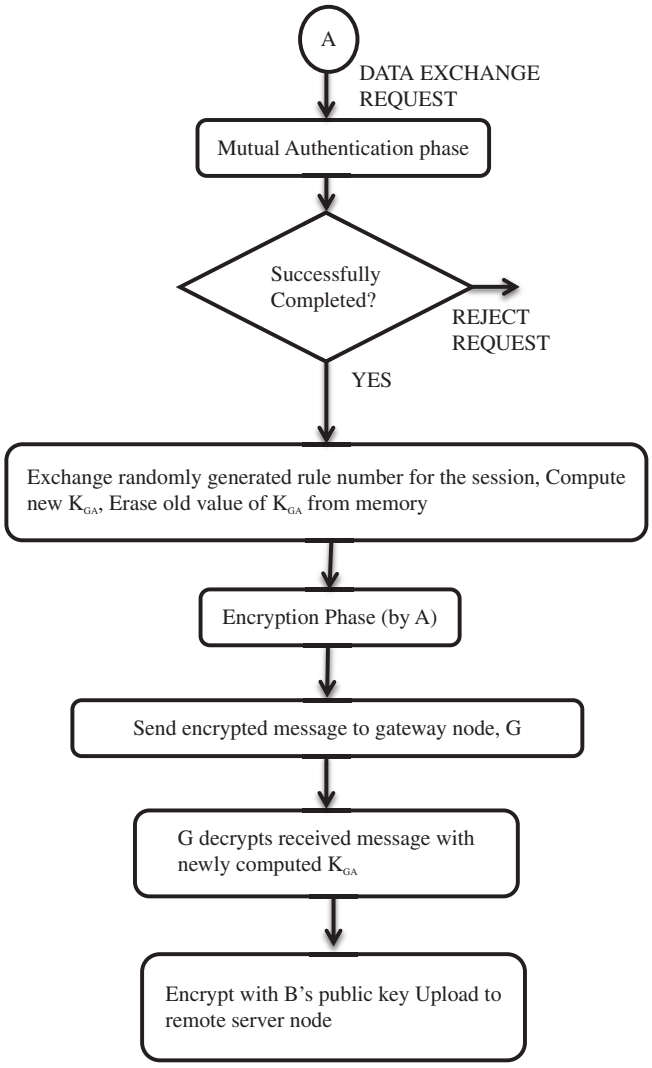


**Figure 5:** Flow chart for data encryption and uploading phase

## 4 Implementation and Results

An electronic health record based IoT application was developed. Data sensed by the sensors were connected with the help of a Raspberry Pi machine. Since, the application was dealing with a distributed scenario, as mentioned in the previous sections; blockchain paradigm was used to implement it. Here, it was implemented with the Hyperledger fabric platform. Hyperledger fabric is a type of consortium blockchain. In this, all participating entities will be having known identities. Since we are forming blockchain of gateway nodes that are already part of an IoT application, all nodes will be having their own identities.

Blockchain implementation was assisted by nodejs, Go and Docker installations. Two gateway nodes were added as organizations with two peers each. When added to the blockchain, each organization will be assigned a public key-private key pair. These nodes are capable to perform asymmetric encryption. The test case developed here consists of two organizations. A device with ID,

"device_1" is registered successfully with the "Organization_1." Master key of device_1 is generated successfully and shared with Organization_1 with No-Share Key Exchange algorithm. Algorithm used for encryption here is AES with 128 bit key length. Even though AES is not a lightweight algorithm, for implementation purpose, it is used. Since NSKE algorithm works on the basis of non-linear computations, with 128 bit length key, the proposed method provides security. Data generated by the device are encrypted by using $K_M$ and then by using new $K_{GA}$ value and uploaded to the storage. Here, for testing purpose, storage used is MongoDB. Master key will also be shared with the recipient organization, only if its attributes are matched with the specified set of attributes. This attribute-based encryption scheme provides another layer of security in addition to the two-level encryption in the proposed method. This provides access control also and this checking functionality is executed by the blockchain participants.

When a user requests for data, blockchain will check whether the attributes of that user matches with the set of attributes specified by the data owner. The code for testing this is written in the smart contract (chaincode in Hyperledger fabric) in Go language. The same will be tested and if attributes match, access to encrypted data will be given. Once the encrypted data are received, the same can be decrypted by the authenticated receiver.

## 5 Security Analysis

Here, a detailed analysis of the proposed method is given. First section gives the informal analysis, in which the performance of the proposed algorithm against each attack is analyzed. This discussion is followed by the formal analysis of the new protocol using the verification tool AVISPA.

### *5.1 Informal Analysis*

Proposed method is resistant to almost all kinds of possible attacks with the three levels of encryption and with the default security methods in blockchain.

- Replay attack: Timestamp values are embedded in the authentication messages. This is enough to provide protection against replay attacks. For example, in initial step, when the device, A sends the message to gateway node, G, timestamp value, TSi is embedded in the message. When it is received at G, let the timestamp be TSi'. The node G computes the difference between these two and if this difference is more than a permissible delay, T, that means an intruder has involved in between and the message is discarded. Hence, the proposed model is protected against replay attack.
- Man in the middle attack: Even though a third party stands in between, it is not possible to involve in the communication because, the exchange of master key depends on private key matrices. Since the attacker is not aware of the private key values of involved parties, master key can be shared only between properly authenticated users. Blockchain members are authenticated by the peer participants for every communication. Cryptogen tool in hyperledger fabric provides certificated credentials for the verification of user identities. Also, all message exchanges are encrypted with a dynamically generated group key.
- Eavesdropping attack: Network used is based on Dolev-Yao model [41]. This means, all message transmissions over the network are visible to an adversary also. Even though there is an unintended observer, he/she will never be able to retrieve the two keys involved in data protection. This is because, the master key used is not exchanged over network as such, but exchanged with NSKE algorithm. Even though the steps are observed by the attacker, it is never possible to retrieve the private key matrices from the handshaking steps of NSKE. Second layer

of security is that, the group key value of $K_{GA}$, is computed dynamically for each session. When the rule number is selected by the device, immediately it calculates the new value of $K_{GA}$ and deletes the old value. The adversary is not aware of the previous value of $K_{GA}$ and hence, it is not possible for him to get the new key value. Furthermore, all the transmissions between the device and the gateway are encrypted with the current session group key.

- Brute Force attack: In this attack, the unintended third party will try several possibilities of the key to retrieve the message. In proposed model, there are two keys used for providing data security. The master key size is 128 bits and the group key is also 128 bits long. Even though current value of group key is obtained, it is not possible to compute the next value of group key without getting the rule number. There are 255 rule possibilities. So, the security of key space relies on $2^{128} \times 2^{128} \times 255$ possibilities. Time required to complete brute force analysis by an attacker will be more compared to the expiry time of the data validity.
- Node tampering: All IoT systems are prone to physical theft since the lower layer involves small IoT devices. The nodes or the devices may be captured and attacker can perform attacks like power analysis to obtain the contents of local memory. This information can be used further to retrieve the private information like key values. In the proposed method, local memory of the device consists of the group key. But, this group key value will be replaced immediately with the newly computed $K_{GA}$. As soon as the rule number is determined, current value of $K_{GA}$ will be removed from the memory. The dynamic generation of group key can provide a significant level of security for each session. Complete protection can be achieved by using any tamper resistance mechanism. One such technique, which can be easily adopted in the proposed system, is the sharing of location information during mutual authentication phase. If the location information differs from the details that were shared during registration phase, node can be considered as malicious.
- Privileged insider attack: In this attack, a privileged insider will reveal some information to the attacker. But, in the proposed method, encryption in each step relies on two symmetric keys- $K_M$ and $K_{GA}$. The value of $K_{GA}$ is computed dynamically for each session and hence, it is not possible to retrieve the contents without getting this new value and the rule number in each session.
- Offline/online password guessing: Guessing the secret key is applicable if same value is used for all session encryptions. Proposed method relies on dynamic values of key involved. Each session is initiated with a mutual authentication phase and hence, only authenticated entities will get the secret values. During the mutual authentication phase, guessing of nonce is also not possible because the attacker does not know the pre-shared hash function.

In addition to the resistance against these types of attacks, authentication and integrity are ensured. Gateway nodes are participants of blockchain and hence, the involved entities are authenticated. Device-gateway communication is also authenticated with the mutual authentication phase. Nonce values are generated by the device and the gateway node and these are exchanged in hashed form. Message authentication is ensured with the help of this pre-shared hash function. This maintains integrity of the exchanged messages. In addition to all these, all steps in a particular session are encrypted with the corresponding group key.

### *5.2 Formal Analysis*

Formal security analysis of the proposed method is done with the widely accepted verification tool called AVISPA [42]. It is a security protocol verification and analysis tool in which new security protocols can be represented in High Level Protocol Specification Language (HLPSL) and then

analyzed with the built-in backend compilers. AVISPA consists of four different back-end compilers: On-the-fly Model-Checker (OFMC), Constraint-Logic-based Attack Searcher (CL-AtSe), SAT-based Model-Checker(SATMC) and Tree Automata based on Automatic Approximations for the Analysis of Security Protocols (TA4SP).

Different phases of the proposed model have been written in HLPSL and tested in AVISPA. For illustration, gateway role is given in detail below in Fig. 6. These HLPSL codes were compiled with the backend compilers provided in AVISPA. The proposed protocol was proved to be "SAFE" under OFMC, "SAFE" under ATSE, INCONCLUSIVE under SATMC and INCONCLUSIVE under TA4SE compilers. OFMC backend compiler verifies the protection status of protocol against passive intruder and here, our method was proved to be "SAFE." Results from OFMC and ATSE compilers prove that, proposed method is safe from possible passive and active attacks like man in the middle attack, replay etc. The system is based on Dolev-Yao model. According to this threat model, all the information exchange will be through a public insecure channel. This means, the proposed method satisfies the security goals (secrecy and authentication) specified in the environment section.

```
role gateway(A,G:   agent,
Kga:   symmetric_key,  H:       hash_func,
SND,RCV:  channel(dy))
played_by G def=
%% Variables declaration
local State: nat,  IDa,IDg,TS1, TS2, Km, M1, M2, M3,M4 : text, Rn,Na,Ng    :message
const m:     protocol_id, init State := 1
%%Transition rulesof steps taken by gateway node in HLPSL
transition
1. State = 1/\RCV({IDa'.IDg'.M1'.M2'.TS1'}_Kga)
/\M2'=H(IDa'.IDg'.TS1'.xor(H(IDa'.TS1'),M1') ) =|>
State' := 2 /\Na' := xor(H(IDa'.TS1'),M1') /\Ng' :=new()/\M3' := xor(H(IDg.TS2),Ng') /\M4' :=
H(IDg.IDa'.TS2.Ng'.Na')/\SND({IDg.IDa'.M3'.M4'.Ng'.TS2}_Kga)
   2.     State = 3  /\RCV({IDa'.IDg'.Rn'.Km'}_Kga)
```

**Figure 6:** HLPSL representation of gateway role

## 6 Performance Analysis

Performance of the proposed scheme is analyzed in terms of computational cost, computation time and the key space.

*role gateway( A,G: agent*,
*Kga: symmetric_key, H: hash_func*,
*SND,RCV: channel( dy) )*
*played_by G def=*
*%% Variables declaration*
*local State: nat, IDa,IDg,TS1, TS2, Km, M1, M2, M3,M4 : text, Rn,Na,Ng :message*
*const m: protocol_id, init State := 1*
*%%Transition rulesof steps taken by gateway node in HLPSL*
*transition*
*1. State = 1/\RCV({IDa'.IDg'.M1'.M2'.TS1'}_Kga)*
*/\M2'=H(IDa'.IDg'.TS1'.xor(H(IDa'.TS1'),M1') ) =|>*
*State' := 2 /\Na' := xor( H(IDa'.TS1'),M1') /\Ng' :=new( )/\M3' := xor( H(IDg.TS2),Ng')*
*/\M4' := H(IDg.IDa'.TS2.Ng'.Na')/\SND({IDg.IDa'.M3'.M4'.Ng'.TS2}_Kga)*
2. State = 3 /\RCV({IDa'.IDg'.Rn'.Km'}_Kga)

### *6.1 Computational Cost*

The authentication and key establishment phases of the proposed scheme are compared with two related schemes. First is matrix based key management scheme [43] and second is Lightweight Device Authentication and Key Management Scheme (LDAKM) [44]. As mentioned in [45] and [46], computational cost of a security scheme depends on encryption, decryption and the hash computations for authentication. Cost of an AES encryption/decryption with 128 bits key would be 9 μJ (milli Joules). A hash operation energy cost is 40 mJ. Cost of encryption is less compared to the cost of hash operations. The cost of the proposed scheme for authentication and key establishment are less compared to the same phases in matrix based key management scheme and LDAKM. Cost computation steps are explained in Tab. 1.

**Table 1** Energy cost summary of constrained node computations for the proposed scheme

| Phase | AES-128 Encryption/ decryption steps count | Hash computation count | Energy cost (mJ) |
|---|---|---|---|
| Data uploading | 2 | 0 | $2 * 9 * 10^{-6} = 18 * 10^{-3}$mJ |
| Authentication | 2 (1Encryption + 1Decryption) | 4 | $18 * 10^{-6+} 4 * 40 * 10^{-3}$ = 160.02 mJ |
| Key management | 8 (4Encryption + 4Decryption) | 0 | $8 * 9 * 10^{-6} = * 10^{-3}$mJ |

Only encryption and hash computation costs are considered. Key exchange steps do not involve any encryption or hash computations. However, for security, all these message exchanges are sent in encrypted form and this is done with group key. Both the protocols in [43] and [44] were proved to have better performance and lower cost compared to several related schemes. As it is clear from the Fig. 7, our proposed scheme is having better performance in terms of computational cost compared to these two.

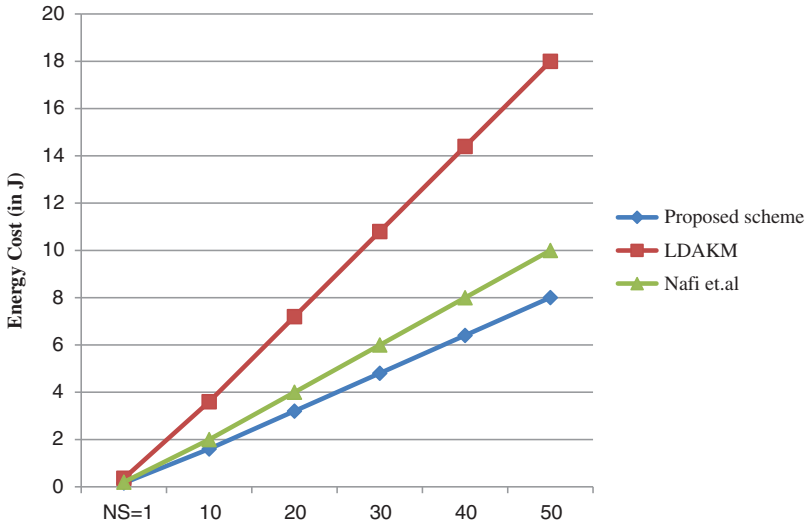


**Figure 7:** Comparison of computational cost in constrained devices

### *6.2 Computation Time*

The proposed algorithm is fast and efficient in terms of computational time involved. It has only lightweight operations like bitwise-XOR and collision resistant one-way hash functions. CA rules are selected randomly for generating dynamic session key values. Exchange of secret information is done by using NSKE algorithm, which has proved to be fast and efficient, compared to conventional key exchange algorithms. Also, it does not involve any complex mathematical operations and simultaneously, it provides non-linear (matrix) security.

### *6.3 Key Space*

Another strength of the proposed algorithm is its key space. Data encryption involves two levels. The first level contains encryption with the master key and the second level contains encryption with the dynamic session key. Both these are of 128 bits length. As the current value of session key is deleted from the local memory immediately after transferring the rule number, it is not possible for the attacker to retrieve it. Dynamic generation of new session key takes place at respective nodes only. The rule set consists of 255 possibilities. So, there is a possibility of $2^{128} \times 2^{128} \times 255$. Also, the proposed algorithm is proved to be resistant against several possible attacks. Hence, it can be concluded that the new system is efficient in terms of security, complexity and computational speed.

Further enhancements on this system can be done on the encryption algorithm. Now, AES is used for ease of implementation. This is the major limitation as constrained nodes have to perform encryption. It can be replaced by any other efficient symmetric lightweight algorithm. In addition to this, the feasibility of blockchain-based methods for malicious node detection can be analyzed.

## 7 Conclusion

The system is based on a hybrid architecture in which cloud storage can be used for storing encrypted information. Data processing and access control functionalities have been simultaneously disseminated to the blockchain entities. Gateway nodes with more computational capabilities compared to low power IoT devices, are included in blockchain. The proposed method protects data with two levels of encryption and the access control is provided with attribute-based encryption. Each session begins with a mutual authentication phase between the gateway node and the device. Encryption steps rely on two stronger keys named master key and session group key. Group key for the next session is computed by using CA rules. The NSKE algorithm and the dynamic key generation phase based on the principles of CA rules make the system highly secure and efficient. The operations involved are lightweight and include only matrix operations and simple XOR operations, and can be executed in a faster way. Hence, the computational overhead is very less compared to existing methods. The proposed algorithm is also proved to be secure with formal security protocol analysis tool AVISPA.

**Acknowledgement:** We thank Cochin University of Science and Technology, Kerala for supporting this research work. We would like to extend thanks to APJ Abdul Kalam Technological University for providing the infrastructure for completing the work.

**Funding Statement:** The authors received no specific funding for this study.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

[1] A. Korte, V. Tiberius and A. Brem, "Internet of things (IoT) technology research in business and management literature: Results from a co-citation analysis," *Journal of Theoretical and Applied Electronic Commerce Research*, vol. 16, no. 6, pp. 2073–2090, 2021.

[2] P. Raj and A. C. Rajan, "Demystifying the IoT paradigm," in *The Internet of Things: Enabling Technologies, Platforms and Use Cases*, 1st ed., vol. 1, Florida, USA: CRC Press, Taylor & Francis Group, pp. 1–38, 2017.

[3] V. Chang, Y. H. Kuo and M. Ramachandran, "Cloud computing adoption framework: A security framework for business clouds," *Future Generation Computer System*, vol. 57, no. 1, pp. 24–41, 2016.

[4] G. Bramm, M. Gall and J. Schütte, "BDABE - Blockchain-based distributed attribute based encryption," in *Proc. ICETE*, Portugal, vol. 2, pp. 99–110, 2018.

[5] M. Wolf and D. Serpanos, "Safety and security in cyber-physical systems and internet of things systems," *Proceedings of the IEEE*, vol. 106, no. 1, pp. 9–20, 2018.

[6] J. Ni, K. Zhang, X. Lin and X. Shen, "Securing fog computing for internet of things applications: Challenges and solutions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 601–628, 2018.

[7] D. Etter, "What is the IoT? in IoT (Internet of Things) programming- A simple and fast way of learning IOT," 2016. [Online]. Available: https://www.goodreads.com/book/show/32729580-iot-internet-of-things-programming.

[8] S. Wu, P. L. Yeoh, W. Hardjawana and B. Vucetic, "Identifying security andprivacy vulnerabilities in 4G LTE and IoT communications networks," in *IEEE 7th World Forum on Internet of Things ( WF-IoT)*, New Orleans, LA, USA: IEEE, pp. 512–517, 2021. DOI 10.1109/WF-IoT51360.2021.9595689.

[9] A. Bahga and V. Madisetti, " IoT enabling technologies," in *Internet of Things-A Hands-on Approach*, India: VPT publishers, pp. 45–65, 2014.

[10] G. George and S. M. Thampi, "Combinatorial analysis for securing IoT-assisted industry 4.0 applications from vulnerability-based attacks," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 1, pp. 3–15, Jan, 2022.
[11] P. Gope and B. Sikdar, "Lightweight and privacy-preserving two-factor authentication scheme for IoT devices," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 580–589, 2019.
[12] B. Russel and D. V. Duren, "Practical internet of things security, Packt publishing," 2016. [Online]. Available: https://www.packtpub.com/product/practical-internet-of-things-security-second-edition/9781788625821.
[13] H. L. Yeh, T. H. Chen, P. C. Liu, T. H. Kim and H. W. Wei, "A secured authentication protocol for wireless sensor networks using elliptic curves cryptography," *Sensors*, vol. 11, pp. 4767–4779, 2011.
[14] W. Shi and P. Gong, "A new user authentication protocol for wireless sensor networks using elliptic curves cryptography," *International Journal of Distributed Sensor Networks*, 2013. [Online]. Available: https://journals.sagepub.com/doi/10.1155/2013/730831.
[15] M. Turkanovic, B. Brumen and M. Holbl, "A novel user authentication and key agreement scheme for heterogeneous Ad Hoc wireless sensor networks," *Ad Hoc Networks*, vol. 20, pp. 96–112, 2014.
[16] M. S. Farash, M. Turkanovic, S. Kumari and M. Holbl, "An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the internet of things environment," *Ad Hoc Networks*, vol. 36, no. 6, pp. 152–176, 2016.
[17] R. Amin, S. H. Islam, G. Biswas, M. K. Khan, L. Leng *et al.,* "Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks," *Computer Networks*, vol. 101, no. 4, pp. 42–62, 2016.
[18] N. Li, D. Liu and S. Nepal, "Lightweight mutual authentication for IoT and its applications," *IEEE Transactions on Sustainable Computing*, vol. 2, no. 4, pp. 359–370, 2017.
[19] P. Gope and B. Sikdar, "Lightweight and privacy-preserving two-factor authentication scheme for IoT devices," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 580–589, 2019.
[20] Y. Kung and H. Hsiao, "GroupIt: Lightweight group key management for dynamic IoT environments," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 5155–5165,2018, 2018.
[21] W. Yu and S. Köse, "A lightweight masked AES implementation for securing IoT against CPA attacks," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 64, no. 11, pp. 2934–2944, 2017.
[22] K. Shahbazi and S. B. Ko, "Area-efficient nano-AES implementation for internet of things devices," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 29, no. 1, pp. 136–148, 2021.
[23] D. D. Hwang, K. Tiri, A. Hodjat, B. Lai, S. Yang *et al.,* "AES-based security coprocessor IC in 0.18 micrometer CMOS with resistance to differential power analysis side-channel attacks," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 4, pp. 781–792, 2006.
[24] N. B. H. Youssef, H. Mestiri, M. Machhout and R. Tourki, "Assessing CPA resistance of AES with different fault tolerance mechanisms," in *Proc. Asia and South Pacific Design Automation Conf. (ASP-DAC)*, Chiba, Japan, pp. 661–666, 2016.
[25] N. B. H. Youssef, H. Mestiri, M. Machhout and R. Tourki, "Implementation of CPA analysis against AES design on FPGA," in *Proc. Int. Conf. Commun. Inf. Technol. (ICCIT)*, Hammamet, Tunisia, pp. 124–128, 2012.
[26] V. Goyal, C. Pandey, A. Sahai and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. 13th ACM Conf. on Computer and Communications Security*, Virginia, USA, pp. 89–98, 2006.
[27] A. Sahai and B. Waters, "Fuzzy identity-based encryption cryptology, Report 2004/086," 2019. [Online]. Available: https://eprint.iacr.org/2004/086.
[28] S. Roy, U. Rawat and J. Karjee, "A lightweight cellular automata-based encryption technique for IoT applications," *IEEE Access*, vol. 7, pp. 39782–39793, 2019.
[29] V. Neumann and A. W. Burks, "Theory of self-reproducing automata," USA: University of Illinois Press, 1966.

[30] S. Ulam, "Random processes and transformations," in *Proc. Int. Congress of Mathematics*, Rio de Janeiro, pp. 264–275, 1950.
[31] G. Kumaresan and N. P. Gopalan, "Programmable cellular automata-based random key generation for one-time pad encryption using unity attractors," *Journal of Applied Security Research*, vol. 16, no. 1, pp. 19–36, 2021.
[32] S. Nakamoto, "Bitcoin: A peer-to-Peer electronic cash system," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf.
[33] A. Adamatzky, "Introduction to mathematical basis of cellular automata, Springer," 2009. [Online]. Available: https://link.springer.com/referenceworkentry/10.1007/978-0-387-30440-3_322#:~:text=A%20cellular%20automaton%20is%20a,structures%20called%20lattices%20or%20arrays.
[34] T. M. Abdullah, A. M. S. Abdullah, A. M. Shamim and M. B. A. Miah, "Blockchain-based security management of IoT infrastructure with ethereum transactions," *Iran Journal of Computer Science*, vol. 2, no. 3, pp. 189–195, 2019.
[35] P. P. Ray, D. Dash, K. Salah and N. Kumar, "Blockchain for IoT-based healthcare: Background, consensus, platforms and use cases," *IEEE Systems Journal*, vol. 15, no. 1, pp. 85–94, 2021.
[36] I. A. Omar, R. Jayaraman, M. S. Debe, K. Salah, I. Yaqoob *et al.,* "Automating procurement contracts in the healthcare supply chain using blockchain smart contracts," *IEEE Access*, vol. 9, pp. 37397–37409, 2021.
[37] G. Zyskind, O. Nathan and A. Pentland, "Decentralizing privacy: Using blockchain to protect personal data," in *IEEE Security and Privacy Workshops*, San Jose, CA, USA, pp. 180–184, 2015.
[38] G. Subramanian and A. S. Thampy, "Implementation of blockchain consortium to prioritize diabetes patients' healthcare in pandemic situations," *IEEE Access*, vol. 9, pp. 162459–162475, 2021.
[39] A. Davies, "Pros and cons of hyperledger fabric for blockchain networks," 2020. [Online]. Available: https://www.devteam.space/blog/pros-and-cons-of-hyperledger-fabric-for-blockchain-networks/.
[40] P. R. Mahalingam and K. A. Fasila, "Zero-share key management for secure communication across a channel," in *Design and Analysis of Security Protocol for Communication,* Beverly, England: Scrivener Publishing, Wiley, pp. 95–107, 2020.
[41] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.
[42] M. Nafi, S. Bouzefrane and M. Omar, "Matrix-based key management scheme for IoT networks," *Ad Hoc Networks*, vol. 97, pp. 102003, 2020.
[43] Y. Glouche, T. Genet, O. Heen and O. Courtay, "SPAN, security protocol ANimatorfor AVISPA," 2006. [Online]. Available: http://people.irisa.fr/Thomas.Genet/span.
[44] M. Wazid, A. K. Das, S. Shetty, J. J. P. C. Rodrigues and Y. Park, "LDAKM-EIoT: Lightweight device authentication and key management mechanism for edge-based IoT deployment," *Sensors*, vol. 19, no. 24, pp. 5539, 2019.
[45] G. D. Meulenaer, F. Gosset, F. X. Standaert and O. Pereira, "On the energy cost of communication and cryptography in wireless sensor networks, networking and communications580–585, 2008. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/?doi%3D;10.1.1.157.348%26;rep%3D;rep1%26;type%3D;pdf.
[46] A. S. Wander, N. Gura, H. Eberle, V. Gupta and S. C. Shantz, "Energy analysis of public-key cryptography for wireless sensor networks," in *Proc. Third IEEE Int. Conf. on Pervasive Computing and Communications*, Kauai, HI, USA, pp. 324–328, 2005.