# Study of different techniques of Key Management and Authentication for IoT systems with a focus on Blockchain

Laxmi Suhas Garde
MS in Computer Science
University of Southern California
Los Angeles, USA
lgarde@usc.edu

*Abstract* — **With the proliferation of connected devices (Internet-of-Things) and their proximity to human life on a day-to-day basis, the privacy and security of these devices are paramount. To assure the safety of these devices modern cryptographic strategies need to be applied that require cryptographic keys. The keys play a crucial role in any cryptographic system, and ensuring the security of keys guarantees the protection of these devices. Due to this reason, key management is critical, and efficient methods need to be implemented for resource-constrained devices like IoT devices and sensors. This paper discusses different techniques of key management and authentication for IoT devices. The paper reviews Blockchain-based methods that are useful for key management in IoT devices.**

**The paper is structured as follows: Section I introduces the concepts of key management, types and usage of keys, authentication, an overview of IoT devices and the associated vulnerabilities, and the need for security in these devices. The paper also explains the concept of Blockchains.**

**Section II briefly describes existing efficient methodologies in IoT devices for Key Management and Authentication. This section also highlights Blockchain based methods for Key Management and Authentication in IoT systems. Section III provides a security analysis of the methods described in Section II based on the referred research papers.**

**Section IV specifies the advantages and limitations of Key management and authentication in IoT devices, with and without a Blockchain-based approach. The advantages and drawbacks of the methods described in Section II are also specified here.**

***Keywords—Key Lifecycle Management, Security, Privacy, Data Integrity, Authentication, Internet-of-Things (IoT), Blockchain.***

## I. INTRODUCTION

Internet-of-things devices and systems are increasingly getting integrated into our lives. The IoT devices and their applications are immense, ranging from handheld devices and smart homes to industries and smart cities. With the increase in IoT applications and use cases, the number of IoT devices in use has increased exponentially. Based on a statistical report given by Cisco, IoT devices will account for 50% (14.7 billion connections) of all networked devices by 2023 [20].

As the number of IoT devices are increasing, the number of connections formed by these devices is massive. These devices and connections have actual data flowing through them, which may comprise sensitive user data. The data through these devices and network needs to be protected to maintain confidentiality, data integrity, and privacy. The security and privacy of IoT devices is challenging due to the limited number of resources available with the IoT devices to implement modern cryptographic techniques. Lightweight computational methods need to be studied by researchers and implementors to provide appropriate levels of security for IoT devices. The basics of cryptography include handling the keying material, and this paper provides an overview of 'Key Management' and 'Authentication' for IoT devices. This paper also outlines Blockchain-based approaches for key management in IoT devices.

### *A. Key Management*

#### Keys

Keys play a vital role in cryptography for encryption or decryption of the data to be secured or for the creation and verification of digital signatures in a secured communication. In cryptography, there are different types of keys: symmetric, asymmetric, public, private, and pre-shared. Symmetric key cryptography uses the same key for the encryption and decryption of data. Asymmetric key cryptography (Public key cryptography) uses two separate keys i.e., a public key and a private key. The public key mainly encrypts the data, and the private key decrypts the data. As keys are crucial in security systems, key management becomes a critical task.

#### Key Management

Key management involves a set of processes and standards that ensure the security of cryptographic keys. Key management deals with the lifecycle of cryptographic keys that involves key creation, exchanges, distribution,

deletion, storage, usage, rotations, and replacement. The steps of key lifecycle management processes are depicted in Figure 1 [2].

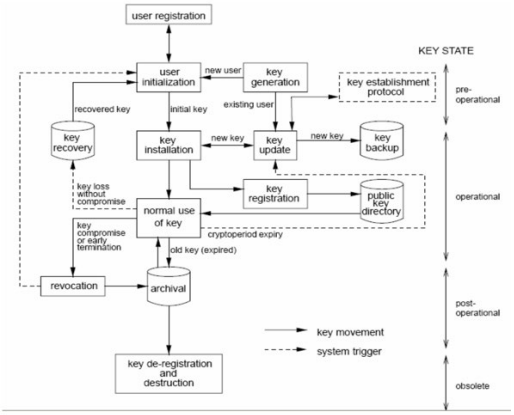


Figure 1: Key Life cycle management [2]

Key Generation, Distribution, and Installation: Key generation is the first step in key management. A cryptographic key is generated using random number generation or by using cryptographic algorithms like Rivest Shamir & Adleman Key Exchange (RSA), Elliptic Curve Cryptography (ECC), and Diffie Hellman Key Exchange. Key Distribution is the transportation or exchange of cryptographic keys toward a subsequent key installation [14].

The Diffie Hellman Key Exchange Algorithm is used for key generation, and key distribution. [2], and a key distribution center (KDC) based key generation and distribution mechanism is available.

Key Update, Backup, and Storage: Key updates apply when the keying material gets replaced by a new material [2]. The cryptographic keys are stored in a secure storage media which can be recovered easily.

Key Deletion: This is the process in which the cryptographic key is first de-registered or uninstalled, followed by removing all copies of that key.

**Key Management Techniques**

Some of the existing techniques in key management are listed below:

1. Mutual Key Management: It is also referred to as the symmetric key management approach that supports key generation for each session. This approach is slow and works best for messages of small sizes and numbers. It is not suitable for IoT systems where a large number of messages are generated continuously [1].

2. Group Key Management (GKM): The GKM technique uses the concept of a group key. The group key is assigned to a group by identifying similar members that can be grouped. This approach reduces the number of keys generated, as the same key can be used by all members belonging to a specific group. The group member identification and group generation are handled centrally. As IoT devices have a distributed nature, this approach is not suitable for IoT systems. The GKM method has performance issues, managing the messages for entities leaving and joining groups [1].

**Key Management Interoperability Protocol (KMIP)**

KMIP is a communication protocol that allows for the storage and maintenance of keys and certificates. It supports client-server communication, where cryptographic systems need keys and key management systems for creating and managing those keys. KMIP provides a standard for key management across all platforms and facilitates cryptographic operations on a key management server. It defines message formats used in communication that can be used for the manipulation of cryptographic keys [21].

KMIP server stores and handles managed objects like certificates, symmetric/asymmetric keys, and user-defined objects. Clients can then use the standard KMIP protocol to access these Managed Objects based on their security implementation on servers. KMIP provides various operations on managed objects like create (create new managed objects, keys), register, get (retrieve a managed object), locate a list of managed objects, export, import, etc. [21].

### *B. Authentication*

Authentication is the process of validating a user's identity who is trying to gain access to a secure system.

**Types of Authentications**

1. Password-Based Authentication: It is a simple authentication technique where the user supplies a password to access the server. The user enters the password on the client device (computer), and the password and username mapping are checked on the server (already registered on the server). If the entries match, authentication of that user is successful.

2. Token-Based Authentication: This is an authentication mechanism that verifies the identity of the user by using a token. Once the user logs in using credentials to access a service, the credentials are verified, and an authenticated token is generated and given back to the user (web browser). This token can be used by the user for the specified time till its expiry for authentication on multiple servers. Single sign-on is based on this type of authentication.

3. Digital Certificate-Based Authentication: This type of authentication uses certificates and SSL protocol to authenticate the user to a server. The client sends the

user's certificate and digitally signed data to the server using SSL. The server verifies and authenticates the user's identity based on the certificate and digital signature [22].

**Authentication Factors**

The authentication factor is a piece of information that authenticates the identity of the user [3].

- Something you know: This is based on the knowledge factor, or on what the user knows, e.g., a password or PIN [3] [4].
- Something you have: This factor is based on a piece of information that is possessed or owned by the user, e.g., smart card, RSA SecureID, smartphone, and cards [4].
- Something about you: This factor is based on a biometric factor related to the user i.e., a biometric pattern like an iris scan, face detection, or fingerprint of the user [3] [4].

**Authentication Techniques**

1. Single Factor Authentication: The authentication mechanism considers only one of the authentication factors to validate the user's identity.
2. Two-Factor Authentication (2FA): Authentication mechanisms that consider any two authentication factors. The commonly used 2FA mechanism is "something you know" with "something you have", e.g., the use of a text password (something you know) and RSA SecureID token (something you have) to access a secure banking application.
3. Multi-Factor Authentication (MFA): Authentication mechanisms that consider two or more authentication factors are categorized as multi-factor authentication (MFA) systems. Two-factor authentication is a type of multi-factor authentication.

Figure 2 shows the evolution of authentication techniques used in modern systems.


Figure 2: Evolution of authentication systems [4]

### *C. Internet-of-Things*

The IoT (Internet of Things) devices are a network of physical objects with embedded sensors, interconnected to exchange information and data with other devices on the internet. The IoT devices are created based on specific application use-case and employ cost-effective sensors linking to wireless communication systems transferring information to a centralized system. IoT has applications in eclectic domains like the military, healthcare, industries, smart homes, and commodity hardware. Due to the rapid growth of IoT devices and networks, the application data and devices revealed to the network have also increased. Therefore, the security and privacy of these devices and networks is crucial.

**Internet-of-Things: Need for Security**

The majority of IoT devices and applications are not designed to handle security and privacy concerns which lead to issues of confidentiality, data integrity, authentication, and key management in the IoT networks.

**Types of attacks on IoT devices**

1. Denial of Service attacks: IoT devices are susceptible to these attacks where the user is denied access to a resource like a computer or a network. Due to inadequate storage capacity and memory, IoT devices are more vulnerable to this attack. The prevention mechanisms also require a higher computational capability, making IoT devices even more vulnerable [5] [6]. DoS attacks are launched to clog the network and consume resources: bandwidth, disk space, memory, and processor time [5].
2. Eavesdropping and Traffic monitoring: This attack occurs when the IoT devices and networks are constantly monitored. This imposes a serious threat to data integrity and confidentiality, as the entire data traffic is monitored by the attacker [5].
3. Physical attacks: This attack involves tampering with the IoT devices' hardware. The IoT devices are distributed, present outdoors, and are at risk of being tampered by humans, nature, or any other physical element. Due to its multi-domain environment, distributed nature, and lack of common standards, IoT systems security becomes challenging.

To establish a secure IoT system the following security and privacy challenges need to be addressed:

- End-to-End Security: The verification of identity on both the IoT devices and hosts is necessary. Protocols like TLS and IPsec need to be used for negotiating session keys, and cryptographic algorithms need to be implemented. Both ends in IoT systems with E2E security need to rely on the fact that the communication is secure, and the data is not modified in transit [5].
- Authentication and Identity Management (AIM): These processes manage and secure information. Identity Management identifies objects, and authentication validates the identity between two communicating parties. Managing AIM is essential for IoT devices as multiple users need to authenticate

each other. An efficient AIM approach needs to be defined [5] [6].

- User Privacy and Data Security: IoT devices are ubiquitous and distributed. The privacy and data security of these devices is a concern. The integrity and confidentiality of the data stored, managed, and shared by IoT devices need to be protected with proper mechanisms in place [5].

- Authorization and access control: The security requirements that can be introduced in IoT devices and systems security frameworks to overcome the above challenges are: Lightweight public key infrastructure (PKI), Lightweight Key Management systems that enable key creation, distribution, and establish trust, developing lightweight cryptographic techniques protecting the data stored and in-transit, building techniques supporting AIM, resilience to attacks, location, and client privacy [6].

### *D. Blockchain*

Blockchain is based on a distributed, decentralized ledger technology that eliminates the need for third-party validations during a transaction on a peer-to-peer network, establishing trust and transparency. In Blockchain, all the participants in a network have access to a distributed ledger which contains immutable records of every transaction, stored as a 'block' of data. The blocks store information based on the specifications given by the user. Every block connects to the blocks created before and after it, forming a chain of blocks (data). The blocks are immutable, and the transactions are blocked together creating a Blockchain. Blocks are stored as cryptographically secured records of data (hashes) following smart contracts establishing greater security and trust. Figure 3 shows a simple Blockchain architecture comprising blocks linked together as a chain. Each block contains a header containing the hash value of that block, the previous block's hash value, timestamp, nonce, and blockchain address of the block creator. The block contains the list of transactions in the body section [7] [8] [9].

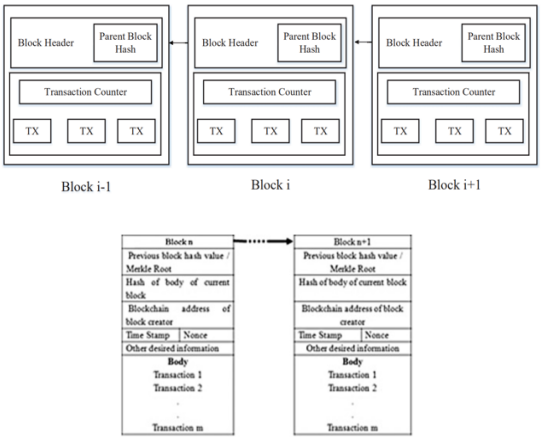


Figure 3: Simple Blockchain Architecture and Block Structure [7] [8]

As the transactions are recorded only once, the duplication of records does not occur, which promotes the reuse of the same transaction information as opposed to the traditional systems for record keeping. With the evolution of IoT and the increase in the amount of data generated, and the need for improved security requirements, blockchain is efficient for handling record keeping or performing validations.

**Blockchain for Internet-of-Things Systems**

Due to heterogeneous connectivity in IoT devices, they have higher chances of security breaches and need proper mechanisms to ensure data security, privacy, and authentication. Blockchain has a distributed architecture and establishes trust among the peer nodes. These advantages of Blockchain can be leveraged to manage operational, security, and privacy requirements for devices in an IoT network [8].

With the use of Blockchain technology, IoT devices can exchange information without any need for a trusted third party to establish trust. Blockchain can significantly reduce operational costs and enhance the performance of IoT devices as no intermediary systems are required. Data security and integrity are maintained, as Blockchain uses hashing mechanisms that can detect any mismatch in the data stored and transferred in IoT devices [8].

## II. METHODOLOGIES

### *A. Key Management and Authentication in IoT Devices*

In an interconnected network such as IoT, where data flows through multiple nodes, a single key approach to secure data is insufficient. Also, modern cryptographic techniques require large key sizes and extensive storage which is challenging to manage in IoT devices having limited computational capability. To ensure security and data privacy, a lightweight key management strategy is essential for IoT devices. A few of the current key management and authentication methods are listed below.

**Methods**

**1) Smart Object based Key Management and Authentication**

The technique incorporates a set of Smart Objects (SO) or Smart middleware that stores, records, and processes the IoT data. As per figure 4, all nodes in an IoT network register to a Smart Object (SO) before data can be transferred securely through the SO. The SO assigns keys to secure the data. The key-sharing process in this technique uses a symmetric key encryption strategy. A message flows from a source node to the destination via the intermediate nodes (registered to the SO), all consecutive nodes share a pair of mutual keys to encrypt-decrypt the protected message, and this happens until the message reaches the destination node securely (Figure 5). Key sharing works by referencing a key-table structure

that stores the connection relation between any two nodes and the key (if any) shared between them. During direct or intermediate message transfers, the key-table structure is referenced for secure transfers [1].

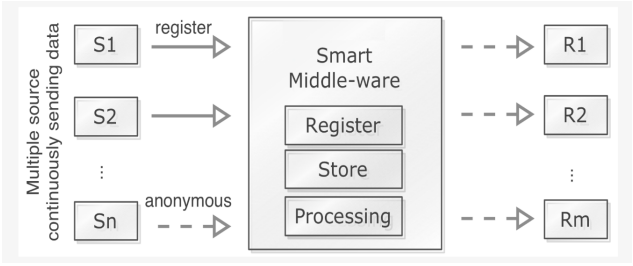


Figure 4: Smart Object node with data flow from source to destination nodes [1]

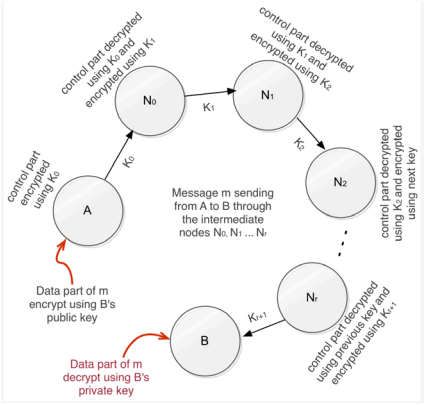


Figure 5: Key sharing process node A to node B [1]

The key generation and distribution process follow an algorithm that first checks if no pair $\{A, B\}_K$ exists, where A, B are the nodes and K is the key. Then the algorithm checks if no connection $CT_{AB}$ exists between nodes A and B, and if the above condition is satisfied, A generates a new mutual key and adds its $CT_A$ with K and a random value v. A hash key k (small k) is created using a conversion function. Then it checks if a connection $CT_{BA}$ exists, and if it exists, B receives hash value k and random value v from A, and the entry for connection $CT_{BA}$ is updated if the stored hash matches. If $CT_{BA}$ does not exist, B receives k from A, and it is added to $CT_B$. Once the secure registration is available, the message is sent using parameters A, B, m (message), and K [1].

Authentication: In the proposed algorithm above, the hash key 'k' generated is used for authentication in this methodology. In a situation where the physical pairing between two nodes is lost, the receiver authenticates the sender by using this hash key, provided they have a mutual key K and a random value v stored in the global table structure. When the pairing is lost, the sender transmits the random value v from its global table to the receiver, and the receiver matches the random value with k using a hash function. If the hash matches, the authentication is successful [1].

**2) Key Management and Mutual Authentication using Hybrid Approach**

The proposed technique in the paper source [11] showcases a hybrid key management scheme for Wireless Sensor Networks (WSNs) based on Elliptical Curve Cryptography and a hash function to generate pre-distribution keys. The hybrid approach supports mutual authentication between the sensor nodes. The hybrid approach functions in four phases:

Phase 1: Parameter Selection for Elliptic curve

Initially, in Phase 1, the server generates a pool of keys using the Elliptic curve cryptographic equations. The parameter selection is important as it improves network connectivity by reducing the number of links compromised by attackers [11].

Phase 2: Unique Key Generation

In Phase 2, unique seed keys are generated after the ECC parameters are finalized in Phase 1. Both phases 1 and 2 are executed before sensor nodes deployment.

Phase 3: Identity-Based Key ring Generation

The key ring generation method uses the node ID, unique key, and hash function assigned in Phases 1 and 2. The server randomly chooses 'm' other nodes to generate a key ring using a hash function and stores node IDs, and respective keys in the node memory. This phase is known as the key pre-distribution phase [11].

Phase 4: Key Establishment and Authentication

In this phase, all nodes share their node IDs with other nodes in the network. When one node A is in the proximity of another node B, A first checks if B belongs to the same key ring. If yes, A sends a message request with a timestamp and hash value C to node B. Once node B receives this message from A, it computes a hash value C' and compares it with C. If the hash matches, the nodes are mutually authenticated and generate a session key for any exchanges. The key establishment phase works in two modes: direct and indirect [11].

**3) Lightweight Device Authentication and Key Management LDAKM**

The paper source [10] proposes a method using a network model comprising an edge node connecting multiple IoT devices, a cloud server, and a trusted authority. The communication between IoT devices and edge nodes needs to be secured, and the communication between an edge node and cloud server. The method proposed in the paper [10] uses lightweight cryptographic operations like bitwise XOR operations and hashing due to the resource constraints in IoT devices.

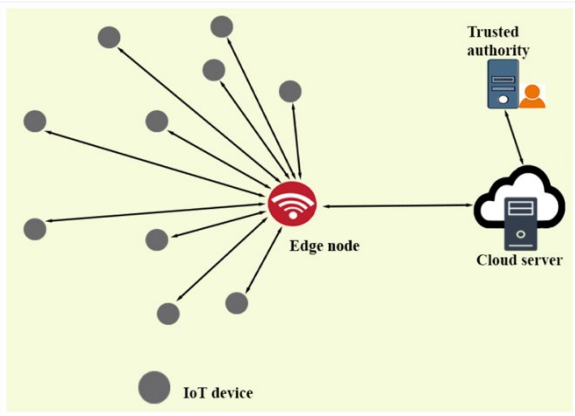


Figure 6: Network model used in LDAKM scheme edge based IoT environment [10]

The proposed scheme has a registration phase in which the Trusted Authority (TA) registers all the IoT devices, Edge nodes, and cloud servers before they are installed in the deployment area. The next phase after the registration is the 'Authentication and Key Agreement phase'. The key management secures the authentication and key management between IoT devices and the Cloud server using a trusted edge node [10].

The steps for authenticated secure communication between IoT devices and Cloud servers are as follows. The steps include calculations that mainly include hash, XOR, and OR operations:

- When an IoT device wants to send a secure message to the Cloud server, the IoT device picks a random nonce and current timestamp to compute parameters M1 and M2 using hash functions. The IoT device sends a message (MSG1) to the Edge Node.

- Once MSG1 is received by the Edge Nodes, it verifies the timestamp of the message by using predefined conditions. If the condition is valid, the Edge Node fetches the required IDs from the received message and performs another set of computations. The IoT device is authenticated by the Edge Node and can access the Cloud Server resources via Edge Node. Edge Node then sends a hashed MSG 2 with nonce and timestamp and sends it to the Cloud Server.

- The Cloud Server receives the MSG2 and performs similar computation as in Step 2, and if the checking is valid, the Cloud server sends a response back to the Edge node, and the edge node then sends a response back to the IoT device. This process establishes secure communication between the IoT device and the cloud server by using lightweight computational techniques [10].

### *B. Key Management and Authentication in IoT Devices using Blockchain*

**Methods**

**1) Blockchain-based distributed authentication and key management for IoT Networks**

The Blockchain-based architecture design proposed in [12] contains three layers: devices, fog-blockchain layer, and cloud-blockchain layer. In figure 7, the device layer contains sensors or IoT devices, the fog layer contains access managing nodes (AMN). The devices are grouped based on domains, and each domain is handled by an AMN. AMNs are grouped to form a network in the fog layer. These AMNs are responsible for the generation, distribution, and management of keys for the devices connected to them. The AMNs on the same network share a Blockchain structure for key management and authentication-related transactions.

The fog layer connects to the Cloud-Blockchain layer that stores and manages multiple blockchains. The cloud layer has manager nodes with high computational capabilities to handle the constrained resource availability in IoT devices. The manager nodes store all the data generated by the devices at the lower levels, and the data is present in an encrypted format in the cloud layer [12].

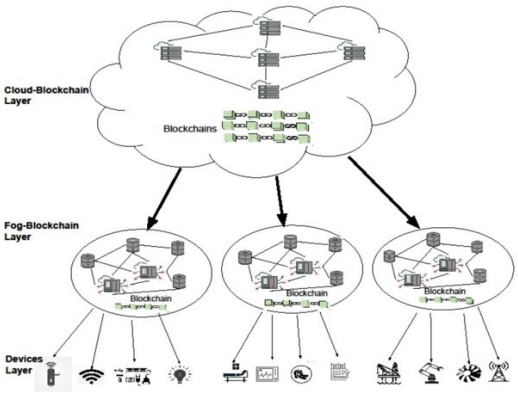


Figure 7 Blockchain-based IoT Architecture [12]

System Variables: These are the variables that need to be accepted by all entities of the system. The list of system variables as proposed in the scheme is given in figure 8.

| | |
|---|---|
| $H$ | One-way hash function mapping the set ¡0,1,• • ,p-1¿ onto itself |
| $h$ | Cryptographic hash function |
| $D\_id$ | Unique identity given to the Device |
| $AMN\_id$ | Unique identity of the Access Managing Node |
| $pk\_D$ | Permanent Public Key of Device |
| $prk\_D$ | Permanent Private Key of Device |
| $pk\_AMN$ | Permanent Public Key of AMN |
| $prk\_AMN$ | Permanent Private Key of AMN |
| $puk\_k$ | Public key from the generated key set |
| $prk\_k$ | Private key from the generated key set |
| $N$ | Number of key pairs generated per device |
| $E_{key}$ | Encryption using key |
| $D_{key}$ | Decryption using key |

Figure 8 System Variables Notation table [12]

One-way Hash Chain: One-way hash chains generate a set of cryptographic keys from a single key. In this technique, seed and cryptographic hash functions, the successive application of the hash function to the seed generates hash values known as a hash chain.

The key management and authentication scheme proposed works in the following phases:

- System Initialization and Device registration phase:

The manager nodes are responsible for selecting the system variables, and the variables are announced by the manager nodes to the access managing nodes (AMNs) at the fog layer. AMNs manage keys associated with the devices connected to them, and each device at the device layer generates a public/private key pair.

AMN registers new devices by providing a license (unique device identity, unique identity of the AMN, permanent public key of the device, and a signature using the private key of the AMNs). The uniqueness of a device's identity is checked using the smart contracts written for the device registration in Blockchains. If the device is unique, with a valid transaction, then the license and registration details of the device are stored in the Blockchain, which can be accessed only by AMNs in that network [12].

- Key management and Authentication phase:

For communication between registered devices, when a registered device A contacts another device B in the same domain, the shared data is secured using encryption. Device A generates a seed encrypted using the public key of the AMN and sent to its AMN along with the registered license. Then the AMN generates N number of public/private key pairs and hash value using the one-way hash chaining. The AMN creates a transaction to store the hashed value on the Blockchain. Once the transaction is verified by all AMNs in the network, the generated key set is encrypted using the public key of the registered device A. When A wants to connect with B, A proves its authenticity to B and establishes a session key for communication.

A sends a message to B, and B authenticates A by checking the hash value. If the hash is not valid, B reports it to the AMN, and if the hash matches, it accepts the request from A and establishes a session key for communication [12].

- License Revocation phase:

If a particular device is infected or found malicious, the AMN revokes the license of that device. The transaction stores the into a different or new block so that no new messages pass that device [12].

**2) Blockchain based key management for Fog enabled IoT devices**

Figure 9 shows the proposed architecture for Blockchain based key management system for Fog-enabled IoT devices [13]. The architecture incorporates a trusted authority (TA) which generates the system variables and initializes the fog devices. It contains a fog node, end user, and access point. The fog node is a fog device on a fog system which is a TPM-type device and can join/leave the system anytime. The end user is the user delegating tasks to the fog system. The fog device connects with an access point to access the parent chain. A side chain is managed by the proposed system to store group parameters generated by the fog device on joining the system. The parent chain stores the information in Blockchain [13].

The scheme proposes an 'improved DConBE' (dynamic contributory broadcast encryption) based key management, which allows the fog nodes to establish a fog system. A DConBE scheme is implemented using an asymmetric group setting to reduce any computational overheads, and it considers key transmission without considering encryption or decryption. This ensures dynamicity and authentication [13].

The Threshold Anonymous Announcement (TAA) provides a controlled dynamicity for fog devices and systems. In TAA, each signer has a TPM device that signs any message. The signer verifies that the user is accredited by the trusted authority (TA) [13].

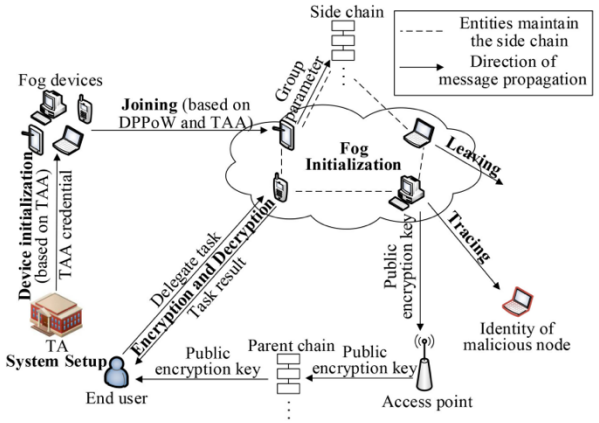


Figure 9: Architecture of Blockchain-based key management for Fog-enabled IoT devices [13]

The scheme comprises the following stages as described in the source [13]:

Initial stages include a globe setup, following the device initialization and fog initialization. The globe setup includes generating the system parameters. In device initialization, the fog device is initialized which comprises a TPM device that internally pre-stores secrets. The fog device obtains a credential issued by TA, which is used to generate a signature using TAA. In the fog initialization stage, using TAA and DConBE, the fog devices negotiate a group size, group encryption key, and fog device's

decryption key. A side chain is initialized by fog nodes to record group parameters.

The next stages in the scheme are 'joining and leaving'. In the joining stage, a fog device joins as a member of a fog system. The computing power of the fog device is verified, and the device then joins the fog system using TAA and DConBE, like the fog initialization stage. Similar to joining, is the leaving stage, where a fog device leaves the fog system, updating the group encryption key and each fog device's decryption key.

Encryption and Decryption are the next stages of the proposed system. The user that knows the group encryption key delegates these tasks to a subset of nodes in the fig system. The next stage is tracing which traces any malicious activities in the system. The fog device with malicious intent is identified, and appropriate action is taken to secure the system [13].

## III. Security Analysis

This section provides a security analysis of all the methods discussed in Section II.

- In Smart object-based key management and authentication, the proposed method is tested for performance in a smart home system. The paper [1] highlights that this approach reuses the existing connections, which enhances the performance and is suitable for IoT devices as the number of messages in IoT devices is large, even though the message size is small. For the key generation and distribution time, it is derived that the delay depends on the smart object's load and the frequency of the network at that time. Figure 10 from the paper [1] shows the performance comparisons of the proposed method with the other methods.

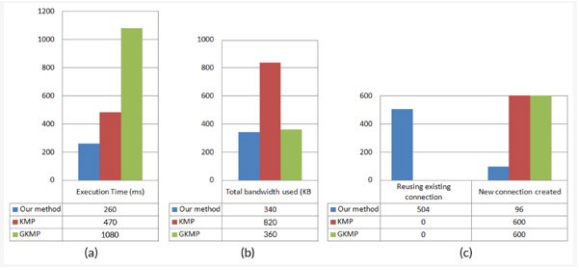


Figure 10: Performance comparison of proposed Smart Object method, KMP and GKMP based on execution time, total bandwidth, and connection reuse [1]

- The paper [10] highlights that LDAKM has remarkably better performance, in terms of communication costs, as compared to other models considered in the comparative study. The proposed model had better computational costs than half of the models but provided better security features than all the models.

- Paper [11] provides an experimental analysis of the hybrid approach used for key management and authentication. From the analysis, it is inferred that the method takes a smaller number of bytes to form secure communication between sensors. The energy consumption is approximately 30% less than the other compared schemes and reduces the time delay in the packets to communicate with neighboring sensors.

- The paper [12] provides a block-chain based key management and authentication scheme suitable for IoT devices. The proposed scheme is highly resilient towards Man-in-the-middle attacks, Denial of service attacks, and Sybil attacks and provides scalability along with authentication and data integrity [12].

## IV. Advantages And Drawbacks

### *A. Advantages*

IoT devices have low computational capabilities, and due to restricted resource availabilities, key management, and authentication act as tools to manage the privacy and security of these devices. Efficient and lightweight key management techniques play an important role in implementing cryptographic techniques for IoT devices.

Blockchain-based key management is advantageous for IoT devices. Blockchains are distributed and provide greater trust between the communicating nodes. It removes the need for a trusted third party by reducing the communication cost and resources required for third-party verifications. It provides low-cost, high-performance, and secure transactions for IoT devices.

### *B. Drawbacks*

The drawback of Blockchain based key management is that Blockchain uses asymmetric cryptography to ensure the integrity of data and privacy, but when an attacker gets hold of the private key of that Blockchain, then the data is at risk.

The other drawback of Blockchain is its adaptability. The field is still under study and research phase and requires high operating costs.

Table I and Table II specifies the methods discussed in Section II with their respective advantages and drawbacks.

TABLE I. KEY MANAGEMENT & AUTHENTICATION APPROCAHES FOR IOT DEVICES

| Reference Paper | *Method* | *Advantages - Drawbacks* |
|---|---|---|
| SO Based on Efficient Key management for IoT devices | Smart Objects (SOs) approach handling heterogeneous data sources to provide a unified representation of data and to ensure the level of security and reliability associated with each data object. [1] | Advantages: Enhanced performance due to the reuse of connections. Prevention of attacks like MITM, Masquerade attack, and target-oriented attacks.<br><br>Drawbacks: It simulates delay introduced by the network through bandwidth computation, which needs to be improved. [1] |
| LDAKM-EIoT | A lightweight cryptographic operation like bitwise XOR, hashing for Edge-based IoT devices [10] | Advantages: Lightweight. Provides better security features at lower computation and communication costs. Prevents MITM, Replay attacks [10] |
| HYBRID APPROACH | Hybrid key management for WSNs to pre-distribute and establish secure, authenticated communication links between nodes using symmetric/asymmetric key cryptography [11] | Advantages: Conserves energy. Packet broadcast delay is lesser. Increase link formation in nodes, enhancing authentication and improving prevention of attacks [11] |

TABLE II. BLOCKCHAIN BASED KEY MANAGEMENT & AUTHENTICATION APPROCAHES FOR IOT DEVICES

| Reference Paper | *Method* | *Advantages - Drawbacks* |
|---|---|---|
| Authentication and KM in Dist. IoT using Blockchain | Provides efficient solutions using Blockchain, Cloud, and Fog computing [12] | Advantages: Scalable, data integrity, resilience from attacks. Reduced block preparation time for bulk transactions [12] |
| Blockchain based KM in Fog-Enabled IoT devices | Manage secure keys and establish secure group channels by use of improved DConBE and TAA schemes [13] | Advantages: Provides data recoverability, conditional anonymity, nonrepudiation, conditional anonymity, and resource authentication.<br><br>Drawbacks: Not tested in real-world scenarios. Scalability unknown [13] |

## V. CONCLUSION

This paper concentrates on Key Management and Authentication in IoT devices, their types, methods, and techniques. The paper describes IoT devices and the need for security in these systems. The purpose of this paper is for the reader to understand the concept of key management and authentication in IoT devices, and how Blockchain-based systems can be leveraged for the same in IoT devices. The paper describes efficient methods of Key management and authentication in IoT devices and discusses Blockchain technology methods that can be utilized for Key management in IoT devices. The security analysis, advantages, and limitations of the methods are described, which provides a detailed view and comparison of these techniques and approaches, giving meaningful insights.


## ACKNOWLEDGMENT

I would sincerely like to thank Professor Clifford Neuman, who gave me invaluable input and insights while working on my research for this paper. I appreciate the efforts, guidance, and support provided by the Professor toward the successful completion of this paper.


## REFERENCES


[1] T. Tabassum, S. A. Hossain, Md. A. Rahman, M. F. Alhamid, and M. A. Hossain, "An Efficient Key Management Technique for the Internet of Things," *Sensors*, vol. 20, no. 7, p. 2049, Apr. 2020, doi: 10.3390/s20072049. https://www.mdpi.com/1424-8220/20/7/2049

[2] Bardis, Nikolaos & Doukas, Nikolaos. (2008). A New Approach of Secret Key Management Lifecycle for Military Applications. WSEAS Transactions on Computer Research. 3. 294-304. https://www.researchgate.net/publication/342376506_A_New_Approach_of_Secret_Key_Management_Lifecycle_for_Military_Applications

[3] I. Velásquez, A. Caro, and A. Rodríguez, "Authentication schemes and methods: A systematic literature review," Information and Software Technology, vol. 94, pp. 30–37, Feb. 2018, doi: 10.1016/j.infsof.2017.09.012.https://www.sciencedirect.com/science/article/pii/S0950584916301501

[4] A. Ometov, S. Bezzateev, N. Mäkitalo, S. Andreev, T. Mikkonen, and Y. Koucheryavy, "Multi-Factor Authentication: A Survey," Cryptography, vol. 2, no. 1, p. 1, Jan. 2018, doi: 10.3390/cryptography2010001.https://www.mdpi.com/2410-387X/2/1/1

[5] Køien, Geir & Abomhara, Mohamed. (2014). Security and privacy in the Internet of Things: Current status and open issues. 10.1109/PRISMS.2014.6970594.https://www.researchgate.net/publication/269687360_Security_and_privacy_in_the_Internet_of_Things_Current_status_and_open_issues.

[6] Ovidiu Vermesan and P. Friess, Internet of Things. Converging Technologies for Smart Environments and Integrated Ecosystems. River Publishers, 2013. http://www.internet-of-things-research.eu/pdf/Converging_Technologies_for_Smart_Environments_and_Integrated_Ecosystems_IERC_Book_Open_Access_2013.pdf

[7] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends," 2017 IEEE International Congress on Big Data (BigData Congress), Jun. 2017, doi: 10.1109/bigdatacongress.2017.85. https://ieeexplore.ieee.org/document/8029379

[8] O. Pal, B. Alam, V. Thakur, and S. Singh, "Key management for blockchain technology," ICT Express, Aug. 2019, doi: 10.1016/j.icte.2019.08.002.https://www.sciencedirect.com/science/article/pii/S2405959519301894?via%3Dihub

[9] W. Baiod, J. Light, and A. Mahanti, "Blockchain Technology and its Applications Across Multiple Domains: A Survey," Journal of International Technology and Information Management Journal of International Technology and Information Management, vol. 29, 2021,[Online].Available:https://scholarworks.lib.csusb.edu/cgi/viewcontent.cgi?article=1482&context=jitim

[10] 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, p. 5539, Dec. 2019, doi: 10.3390/s19245539. https://www.mdpi.com/1424-8220/19/24/5539#sec3-sensors-19-05539

[11] Arunkumar, Sharmila & Kumar, Pramod & Kumar, Manoj & Alazab, Mamoun. (2021). Secure Key Management and Mutual Authentication Protocol for Wireless Sensor Network using Hybrid Approach.10.21203/rs.3.rs-328155/v1.https://www.researchgate.net/publication/350207785_Secure_Key_Management_and_Mutual_Authentication_Protocol_for_Wireless_Sensor_Network_using_Hybrid_Approach

[12] S. S. Panda, D. Jena, B. K. Mohanta, S. Ramasubbareddy, M. Daneshmand, and A. H. Gandomi, "Authentication and Key Management in Distributed IoT Using Blockchain Technology," IEEE Internet of Things Journal, vol. 8, no. 16, pp. 12947–12954, Aug. 2021, doi: 10.1109/jiot.2021.3063806. https://ieeexplore.ieee.org/document/9369319

[13] T. Chen, L. Zhang, K.-K. R. Choo, R. Zhang, and X. Meng, "Blockchain Based Key Management Scheme in Fog-Enabled IoT Systems," IEEE Internet of Things Journal, pp. 1–1, 2021, doi:10.1109/jiot.2021.3050562. https://ieeexplore.ieee.org/document/9319269

[14] S. M. Matyas and C. H. Meyer, "Generation, distribution, and installation of cryptographic keys," IBM Systems Journal, vol. 17, no. 2, pp. 126–137, 1978, doi: 10.1147/sj.172.0126. https://ieeexplore.ieee.org/document/5388039

[15] H. Tan and I. Chung, "Secure Authentication and Key Management With Blockchain in VANETs," IEEE Access, vol. 8, pp. 2482–2498, 2020, doi: 10.1109/access.2019.2962387. https://ieeexplore.ieee.org/document/8943982

[16] Ievgeniia Kuzminykh, Bogdan Ghita, Stavros Shiaeles, 'Comparative Analysis of Cryptographic Key Management Systems', October 3, 2022, from https://arxiv.org/pdf/2109.09905.pdf

[17] Attkan, A., & Ranga, V, 'Cyber-physical security for IoT networks: a comprehensive review on traditional, blockchain and artificial intelligence based key-security', Complex & Intelligent Systems volume 8, pages3559–3591, 2022. https://doi.org/10.1007/s40747-022-00667-z

[18] Kim, K.-W., Han, Y.-H., & Min, S.-G., 'An Authentication and Key Management Mechanism for Resource Constrained Devices in IEEE 802.11-based IoT Access Networks', Sensors, 17(10), 2170, 2017. https://doi.org/10.3390/s17102170

[19] Fang, H., Qi, A., & Wang, X., 'Fast Authentication and Progressive Authorization in Large-Scale IoT: How to Leverage AI for Security Enhancement', IEEE Network, 34(3), 24–29, 2020. https://doi.org/10.1109/mnet.011.1900276

[20] P. Grossetete, "IoT and the Network: What is the future?," Cisco Blogs, Jun. 22, 2020. https://blogs.cisco.com/networking/iot-and-the-network-what-is-the-future#:~:text=According%20to%20Cisco.

[21] "Key Management Interoperability Protocol (KMIP)," www.ibm.com. https://www.ibm.com/docs/en/sgklm/3.0?topic=key-management-interoperability-protocol-kmip.

[22] Oracle Corporation, "Certificate-based Authentication (Sun Java System Directory Server Enterprise Edition 6.3 Reference)," docs.oracle.com, 2010. https://docs.oracle.com/cd/E19575-01/820-2765/6nebir7eb/index.html