A Blockchain-based Three-factor Mutual Authentication System for IoT using PUFs and Group Signatures

Meriam Fariss, Ahmed Toumanari
Laboratory of Applied Mathematics and Intelligent Systems Engineering (MAISI)
National School of Applied Sciences (ENSA), Agadir, Morocco[1]

***Abstract*—The widespread adoption of Internet of Things has brought many benefits to society, such as increased efficiency and convenience in various aspects of daily life. However, this has also led to a rise in security threats. Moreover, resource-constrained feature of IoT devices makes them vulnerable to various attacks that compromise the user's privacy and sensitive information confidentiality. It is therefore essential to address the security concerns of IoT devices to ensure their reliable and secure operation. This paper proposes a blockchain-based three-factor mutual authentication system for IoT using Elliptic Curve Cryptography, physical unclonable functions and group signatures. The main purpose is to achieve a secure mutual authentication among different involved entities while providing anonymous group member authentication and reliable auditing. The AVISPA tool is utilized in the paper to formally prove that the proposed system satisfies the security and privacy requirements.**

Keywords*—Internet of Things; blockchain; mutual authentication; physical unclonable functions; biometrics; group signatures; elliptic curve cryptography

I. Introduction

Internet of Things (IoT) is a rapidly developing technology that has gained significant traction in various fields such as healthcare, military, smart cities and houses [1]–[4]. It involves smart devices that collect thousands of gigabytes of data and use this collected data to make instant decisions that are immediately shared with remote users and servers. Nonetheless, the absence of inherent security measures renders the IoT-based architectures susceptible to many security breaches and privacy violations [5]. Many surveys and researches have been conducted to show the security challenges in IoT [6]–[8].

Authors in [9] indicate that there were 50 billion connected devices by the end of 2020 and this number is expected to increase to 14.7 billion by 2023. As the number of connected devices in IoT continues to increase, there are numerous challenges and issues that arise, particularly in regards to security and privacy. To overcome these challenges, new and emerging technologies such as fog computing and blockchain are integrated with IoT.

Blockchain has gained significant attention from researchers due to its ability to protect IoT devices and security-critical data [10]–[12]. By incorporating blockchain technology into IoT devices, it can provide an effective solution to the security and privacy challenges facing IoT devices. Blockchain can ensure the integrity and authenticity of data, and provide a secure platform for sharing data between devices. Additionally, blockchain can help to create a decentralized and trustless network, which is essential for secure communication and transactions between IoT devices. Blockchain's security stems from its use of cryptographic techniques, such as hash functions, digital signatures, and encryption, to ensure the integrity and confidentiality of data stored on the blockchain.

Besides blockchain, fog computing is an emerging technology that brings important enhancement to the security of IoT devices [13]. The limited resources of these latter leave them vulnerable to security threats. To address this issue, fog computing can enhance their capabilities by offering localized compute, storage, and networking for a cluster of IoT devices. By performing processing and storage tasks closer to IoT devices at the fog node instead of moving the data to a cloud server, fog computing reduces latency and increases network efficiency due to its high-quality services and quick response time. This can help address security concerns by reducing the amount of data that needs to be transmitted to the cloud, which in turn reduces the attack surface for cyber criminals. Additionally, fog computing can provide an additional layer of security by enabling real-time threat detection and response. This can help detect and mitigate security threats more quickly, reducing the potential damage that can be caused by such attacks.

A. Our Contribution

To address the aforementioned security threats while taking into consideration the resource constrained feature in IoT environment, we propose in this paper a blockchain-based secure mutual authentication system for IoT using Physical Unclonable Functions (PUFs) and group signatures providing the following advantages:

1) Permissioned blockchain: to achieve more control, privacy and high transparency over the network, we use a permissioned blockchain where only selected nodes are allowed to participate in consensus.

2) Group signature scheme with two authorities: In our proposed scheme, we distinguish between the group manager and the opening manager roles. The former is in charge of assigning private signing keys to group members, while the

latter can open signatures. This enhanced security, as both have their own secret key, mitigates the risk of untrustworthy authorities.

3) Three-factor authentication: the proposed authentication protocol requires the user to provide three types of credentials: the first factor is something the user knows (password), the second factor is something the user has (hardware token) and the third factor is something the user is (biometric characteristic like a fingerprint, iris scan or facial recognition). Hence, the security of the system is strengthened, as it is much more difficult for an attacker to obtain all three types of credentials.

4) PUFs: they are used to generate a unique private key for every token, which can be used for cryptographic operations such as signing and encryption. The private key is generated by applying a one-way function to the PUF's response, which ensures that the private key cannot be reverse-engineered from the response. The private key is securely stored in the hardware token and can only be accessed by authorized users with the appropriate credentials.

5) Fog computing: provides a trusted entity with more computing and storage resources that supervises a group of IoT devices, controls access and manages communication between devices and remote users. This improves security by providing a local point of control, reducing data transmission, and improving network efficiency, reliability and scalability.

B. Organization

The remainder of this paper is organized as follows. In section II, we present an overview of the related work. Section III is dedicated to the basic concepts of blockchain namely smart contracts and the Practical Byzantine Fault Tolerance (PBFT) consensus algorithm. The cryptographic primitives adopted in this paper are presented in Section IV. The description of our proposed blockchain-based protocol is presented in Section V. We dedicate Section VI to the informal security analysis of our proposed protocol and the formal security analysis using the widely used AVISPA Tool. Finally, we draw our conclusions and present our future work in Section VII.

II. Related Work

Every year, numerous studies are conducted to secure exchanged data over the unattended IoT area. These studies suggested reliable architectures and frameworks to overcome the challenges and security threats in order to achieve secure mutual authentication between all involved parties in the IoT environment. Among these security solutions, blockchain technology brings many solutions for tackling security and privacy concerns in the context of IoT. In 2022, researchers in [14] developed a secure and efficient authentication mechanism for fog computing using blockchain technology. The proposed approach aims to overcome the limitations of traditional authentication methods while maintaining high levels of security and performance. Another secure IoT system was proposed in 2022 [15]. This paper presented a new approach to managing device identities in IoT systems based on blockchain technology. The approach enhanced data security through two methods: a lightweight time-based identification protocol that validates data using hub identification, and a blockchain application that provides secure data storage and sharing among multiple parties with easy access and immutability. In [16], researchers proposed a blockchain-based scheme where certificateless cryptography, Elliptic Curve Cryptography (ECC), and pseudonym-based cryptography (PBC) are employed. The goal is to achieve users' privacy and to hide the true identity of IoT devices using pseudonym-based cryptography. In 2020, Huang et al. [17] presented in their paper a blockchain-based authentication framework for IoT networks to achieve fast decentralized authentication while preserving privacy. The framework satisfied various security requirements such as strong key protection, identity anonymity, single registry, and traceability. In 2017, Cha et al. [18] suggested a blockchain-connected gateway design that is claimed to ensure security and user privacy. However, in 2020, Yavari et al. [19] revealed that [18] is vulnerable to various attacks, including secret disclosure, replay, traceability, and token reuse attacks. They proposed an improved blockchain-based authentication protocol that provides secure access management and anonymity. The paper in [20] proposed a solution to address security risks in IoT, particularly in decentralized authentication by providing a secure framework using blockchain technology which supported certificate issuance, update, revocation, and audit functions through the use of a smart contract. Authors in [21] presented a multi-layer security model for IoT devices operating in multi-hop cellular networks that utilized blockchain's distributed technology. The proposed model offered a viable approach to deploying decentralized blockchain technology for securing cellular-enabled IoT networks. After analyzing the limitations of traditional IoT authentication and security mechanisms, authors in [22] proposed a blockchain-based model to address these issues namely the single-point-failure issue.

III. Blockchain

Blockchain is a secure and transparent digital ledger that records transactions across a decentralized network. Nodes in the network must reach a consensus before new blocks of transactions can be added. The ledger includes various types of transactions, and each block contains a header with the previous block hash, timestamp, version, nonce, difficulty target, and Merkle root [23]. Blockchain can be divided into three types: public, permissioned, and private. Public blockchains like Bitcoin and Ethereum are open to anyone, transactions are validated by consensus mechanisms, and no central trusted authority is required. In contrast, only trusted participants are allowed in private and permissioned blockchains, but there are significant differences between them. In private blockchains, a single private entity controls the network, while in permissioned blockchains, a consortium of organizations adds an access control layer and allows multiple organizations to validate transactions, making it more decentralized than private blockchains.

A. PBFT

Introduced in the late 90s, Practical Byzantine Fault Tolerance (PBFT) [24] is a consensus algorithm used in

permissioned blockchain networks to ensure secure and consistent agreement on the network's state even in the presence of malicious nodes. The network security is ensured by the PBFT algorithm as long as the number of faulty nodes is under a predefined threshold $f=(n-1)/3$ where n is the total number of nodes. In the PBFT consensus, eligible nodes can switch from primary or leader nodes (during a period of time called view) to secondary nodes to reach a consensus on the state of the system. When the leader node is non faulty, the PBFT consensus works as follows:

- Client requests: The client initiates a request and sends it to the network.
- Pre-prepare message: The leader verifies the request message and broadcasts the pre-prepare message to all other replica nodes, containing the client's request and a sequence number.
- Prepare messages: Upon receiving the pre-prepare message, each replica node verifies its legitimacy and broadcasts a Prepare message.
- Commit messages: When no less than two-thirds of the total consensus nodes have sent Pepare messages, the leader node broadcasts a commit message to all other nodes (ie. a consensus has been reached on the client's request).
- Reply message: When at least two third of the received commit messages are valid, nodes return a reply message to the client containing the result of the request transaction.
- State Update: the ledger state is then updated in every consensus node.

If the system encounters a verification failure or network interruption case, then the View change protocol is executed to select another primary node in the network, that is responsible for carrying out the consensus process from the prepare phase through the following steps:

- View_Change_Request: after detecting an exception message, every node in the network broadcasts a view change request to all other participating nodes.
- New_View_Prep: after verifying the View_Change_Request and broadcasting an acknowledgment (no less than ⅔ of the total nodes), the nodes collaborate to prepare a new chosen primary node to substitute the previous one.
- New_View: through a voting process, nodes must reach a consensus to select the new primary node. This latter will take the responsibility of processing requests and generating new blocks.

B. Smart Contract

To address the trust problem in a decentralized environment, smart contracts are programs where required conditions are implemented using a Turing complete language (like go language in Hyperledger Fabric, Solidity in Ethereum). The smart contract byte code is stored in the blockchain platform with a unique address and automatically executed when predefined conditions are verified. It is replicated across all the blockchain consensus nodes. Hence, no third trusted party is needed to make decisions. The main benefits of smart contracts are speed, efficiency, trust and transparency. It also benefits from the security immutability features offered by the blockchain.

IV. Cryptographic Primitives

A. Group Signatures

Group signature is a type of digital signature scheme that was first proposed by Chaum and Van Heyst [25], and then many other contributions on group signature schemes were made in order to allow group members to anonymously sign messages while being traceable by a designated authority. In 2015, [26] proposed a novel short group signature scheme along with two group membership revocation methods that only disclose revocation information to verifiers. This section briefly presents the digital signature scheme by [26] which is adopted in our proposed protocol.

- System Setup phase

In this phase, the system parameters are initialized. The input is a security parameter λ and the output is (PP, sk, gpk, $trace$). Public parameters $PP=(q,G_1,G_2,G_T,e,P_1,P_2,h())$, where G_1, G_2 and G_T are three cyclic groups of λ-bit prime order q and $e:G_1 x G_2 \rightarrow G_T$ is a bilinear map. P_1 and P_2 are the generator points of G_1 and G_2 respectively, and $h:\{0,1\}^* \rightarrow Z_q$ is a secure hash function.

The Group Manager chooses randomly two secret parameters d and s in Z_q^* where (d,s) represents its private key sk. The group public key is $gpk = (D, S, U)$ where $D = d.P_1$, $S = s.P_2$ and $U = u.P_1$. The secret parameter u represents the private tracing key $trace=u$ only known by the openening manager that uses it in *GTrace* algorithm to find the member's real identity.

- Enroll

In this phase group members are enrolled by the group manager. The input is (PP, sk) and the output is a private key $gsk_i=(x_i , Z_i)$ is generated for each group member GM_i by the group manager who chooses randomly a distinct x_i for each member and sets $Z_i = z_i.P_1$ where $z_i = (d- x_i) (sx_i)^{-1} \in Z_q^*$. After that, the group manager computes $tag_i = x_i.Z_i$ and maps it with the relevant group member's identity in a members table. This table also contains $status_i$ that shows if the member is allowed to access the network or is revoked.

- GSign

During this phase, a group member can sign his/her messages. The input is $(PP, gpk, gsk_i, message)$ and the output is the signature $\sigma = (C_1, C_2, c, w)$. Each group member can sign his/her messages using his/her private key gsk_i as follows: choose $k \in Z_q^*$ randomly, computes $C_1=k.P_1$,$C_2=x_i.Z_i+k.U$ and $Q=e(U,S)^k$, computes $c=h(message,C_1,C_2,Q)$ and $w=kc+x_i$.

- GVerify

This is the signature verification phase. The input is $(message, \sigma, gpk)$ and the output is the verification result if the

message has been signed by a group member or not. The verifier computes:

$$Q' = \frac{e(C_2,S).e(P_1,P_2)^w}{e(cC_1+D,P_2)} \quad (1)$$

and checks if $c = h(message, C_1, C_2, Q')$ to confirm or not the validity of the signature.

- MRev

This is the group membership revocation algorithm. The group manager publishes a Revocation List (RL) that contains $tag_i = x_i.Z_i$ for revoked members. The revocation algorithm MRev operated by the verifier takes as input the signature $\sigma=(C_1, C_2, c, w)$ for each member in the RL and the output is: for each member in RL, verifiers can test whether the value of $tag_i = x_i.Z_i$ belongs to a revoked member as follows: compute $e(C_2 - tag_i , S)$ and compare it to Q'. If the equality holds, then the signature σ belongs to a revoked member. Hence, the signature is rejected.

- GTrace

This is the tracing algorithm that takes as input *(trace, message, σ)* and outputs the signer identity using the tracing key $trace = u$. The group opener computes $tag_i = x_i.Z_i = C_2 - u.C_1$, then it searches in the table mapping each tag_i with the corresponding member identity.

B. Fuzzy Commitment Scheme

Fuzzy commitment scheme F is a cryptographic primitive that allows a party to commit to a message without revealing the message itself. It was first introduced in 1999 by Juels and Wattenberg [27]. It is performed in two phases. The commitment phase where the committer creates a commitment by applying a one-way function to a random secret value and the message, which prevents the receiver from determining the message from the commitment. The opening phase where the committer discloses the secret value and committed message. The receiver can verify the commitment by applying the same one-way function to both values.

Fuzzy commitment enhances privacy and security in biometric based authentication systems, by generating a commitment value based on biometric data, such as a fingerprint or face scan, and a secret key.

C. Physical Unclonable Functions

A Physical Unclonable Function (PUF) [28] is a hardware security primitive that is designed to generate a unique signature for a physical device, based on its manufacturing process and the physical variations that occur during the manufacturing process. The mathematical model of a PUF is based on a challenge-response mechanism, where a unique response is generated for every challenge. A PUF can be represented as: R = P(C). The challenge-response pair (CRP) is unique to each PUF. The PUF takes a challenge C as input and produces a response, which is a unique digital fingerprint that can be used for device identification. The response R is generated based on the physical characteristics of the device, such as the pattern of its silicon crystal lattice or the precise positions of transistors in the circuit.

V. The Proposed BC-Auth Scheme

In this section, we discuss the details of our proposed system which is based on blockchain technology, group signatures, Elliptic Curve Cryptography, Physical Unclonable Functions, the Message Authentication Code (MAC) and biometrics. In this BC-Auth, we adopt the permissioned blockchain type, where only legitimate nodes are allowed to access the blockchain, in order to achieve high privacy protection. This blockchain is managed and maintained by permissioned nodes on the basis of the Practical Byzantine Fault Tolerance consensus (PBFT) (see Section III A). These consensus nodes have enough computational, communication and storage resources to operate hundreds of thousands transactions within seconds.

Similar to the Bitcoin block structure, the block in our design is composed of the previous block hash, block version, timestamp, block size, the Merkle root, transaction counter and the recorded transactions in this block.

A. BC-Auth Model

We describe, in this section, the BC-Auth scheme model. The participants involved in this model are:

- Group manager: a trusted entity that enrolls new legitimate group members and generates their private keys and the group public key. It also revokes malicious users and maintains the Revocation List.
- Opening manager: the group manager cannot trace group members. The entity responsible for members' tracing is the opening manager using a secret parameter *trace*.
- Group members: the remote users that are allowed to access the IoT devices remotely.
- Consensus nodes: in our model, we work with the permissioned blockchain where the participating nodes are chosen and authorized to maintain the blockchain under the PBFT consensus mechanism.
- Fog node: a trusted entity with additional computing and storage resources that oversees a group of IoT devices and manages access to them. It also ensures the communication between these devices and the remote users.
- Devices: IoT devices are resource-constrained devices. Each IoT device corresponds to a single fog node. They collect data from the physical world to control and manage the industrial processes.

B. Smart Contract and PBFT in BC-Auth

In our proposed design, each logged in user interacts with the smart contract to broadcast his/her request transactions where *status=0*. These pending transactions are firstly verified by consensus nodes via $GVerify_{gpk}$ to confirm that they're signed by a group member. If the verification fails, the transaction is discarded. Secondly, the opening manager monitors the smart contract to retrieve new verified group members transactions and sends allowed transactions (*status=1*) to the blockchain after verifying that the signer does

not belong to the RL. These latter are verified by consensus nodes via $Verify_{pko}$. Finally, the fog node monitors the smart contract to find new valid transactions. When a sufficient number of consensus nodes agree that the transaction is valid (i.e. successfully verified by $GVerify_{gpk}$ and $Verify_{pko}$), it can then be chained in the blockchain.

A new block is added to the PBFT-based blockchain by a designed primary node PN_i of the current consensus round as follows:

- PN_i collects all the valid transactions of the current round and appends them to a new candidate block $Block_i$;
- PN_i broadcasts $Block_i$ to all consensus nodes;
- Upon receiving $Block_i$, each node verifies its validity based on many parameters such as the Block header, the block generator digital signature, the list of transactions contained in the block,
- If $Block_i$ passes this verification successfully, then each node broadcasts a prepare message along with hash ($Block_i$) to all other participating nodes.
- If the number of received messages is no less than two third of the total nodes, then each node adds the candidate block to its local copy of the ledger.

Hence, the consensus is reached and $Block_i$ is added to the blockchain. If the consensus is not reached, then View-Change protocol is executed (see Section III A).

C. BC-Auth Protocol

Our proposed protocol consists of four phases: initialization phase, member enrollment, login and mutual authentication, and member revocation phase. In this section, we describe these steps in detail.

1) System initialization phase: The initialization phase must be performed before the execution of the protocol. This phase takes as input a security parameter λ and outputs the public parameters $PP=(q,G_1,G_2,G_T,e,P_1,P_2,h())$. Using the Elliptic Curve Integrated Encryption Scheme (ECIES) [29], the group manager chooses randomly two secret parameters d and s in Z_q^*, where (d,s) represents its private key sk. The opening manager generates a random secret parameter $u \in Z_q^*$ that represents its tracing key $trace \leftarrow u$. It also has its private/public key pair (sk_o, pk_o). The group public key is equal to $gpk = (D,S,U)$ where $D = d.P_1$, $S = s.P_2$ and $U = u.P_1$. In the other side, the fog node generates a random secret parameter $sk_{fn} \in Z_q^*$ that represents its private key, and computes the corresponding public key $pk_{fn}=sk_{fn}.P_1$. We assume that each smart device has its private/public key pair (d_j, D_j) where $d_j \in Z_q^*$ and $D_j=d_j.P_1$.

2) Member enrollment phase: This phase is initiated by the user who sends, via a secure channel, an enrollment request containing its chosen ID_i, HPW_i and biometric b_i to the Group Manager. This phase outputs the group members private keys $gsk_i=(x_i, Z_i)$ and the hardware token that stores $\{\alpha, \delta, u_i, A_i, Z_i, f(),PUF_i()\}$. These private keys are uniquely generated using PUFs functions and will be used by group members to anonymously sign their transactions before being sent to the blockchain network. Fig. 1 shows the details of the member enrollment phase.

3) Login and mutual authentication: This phase is performed every time a remote group member needs to access or control a smart device. Fig. 2 shows the login and mutual authentication phase.

- Login: To achieve a secure mutual authentication between the fog node and the group member, the latter needs to successfully login by inserting the correct ID_i, PW_i and b_i. This login phase is performed locally via the hardware token.
- Request transaction: Once the group member is logged in, his/her private key is computed using the physical unclonable function characteristics. A one-time private/public key pair (m_i, M_i) is generated where $m \in Z_q^*$ is a random secret and $M_i=m_i.P_1$. The transaction is structured as follows: Computes the message $Msg = txnumber||to||M_i||D_j||request$, where $txnumber$ is the transaction number and to is the address of the smart contract. Then it computes $E_{msg}=Enc_{pkfn}(Msg)$ where Enc is an AES Encryption, and sends the transaction $Tx= \{data, GSign_{gski}(data)\}$ where $data=(to,E_{msg}, TS_i, status)$ to the blockchain. TS_i is the current timestamp. The status=0 which means that it is a pending transaction. The smart contract is invoked and the consensus nodes verify the signed transaction using the GVerify algorithm. If the verification fails, i.e. the signer is not a group member, then the transaction is discarded. Else, the opening manager monitors the blockchain to find the pending transactions. It verifies then if the signer is a revoked member, i.e. the corresponding $tag_i \in RL$ by computing $tag_i=x_i.Z=C_2-u.C_1$. The revoked transaction is discarded from the blockchain. The allowed transaction Tx' status is set to 1 and is sent to the blockchain. The consensus nodes verify the validity of this transaction by operating the Elliptic Curve Digital Signature Algorithm (ECDSA) [29] Verify using the opening manager public key pk_o.
- Transaction chaining: The transaction is considered valid when at least two-third of total consensus nodes verify the transaction successfully through $GVerify_{gpk}$ (for Tx) and $Verify_{pko}$ (for Tx'). All valid transactions recorded through a predefined period of time are chained into a pending block that can be chained into the blockchain when the PBFT consensus is reached among more than two-third of total consensus nodes.
- Response delivery: The Fog node monitors the blockchain to retrieve the new valid transactions and decrypts the message $Msg = Dec_{skfn}(E_{msg})$, where Dec is an AES Decryption, and computes the signature $R=Sign_{skfn}(M_i \,||\, request)$ using its private key and sends $\{M_i, request, R\}$ to the targeted IoT device. After successfully verifying the received request, the device encrypts the response using the one time group member's public key M_i, and signs it with its private

key d_j. If the response passes the signature verification $Verify_{Dj}$ (RES, E_{res},), then the fog node computes MAC = MAC_{SK} (E_{res}) using the secret session key SK and sends {E_{res}, MAC} to the group member. The latter compares the $MAC'=MAC_{SK'}(E_{res})$ to the received MAC. If the equality holds, this means that the secure mutual authentication is successfully established between the fog node and the group member, who can then decrypt the response using its one-time private key.

4) Member revocation: The group manager maintains the RL that contains $tag_i=x_i.Z_i$ of revoked users. If the behavior of a group member is malicious, or if he does not belong to the group anymore, the group manager can then revoke its membership and add the corresponding tag_i in the public RL. This revocation list is shared only between the group manager and the opening manager.

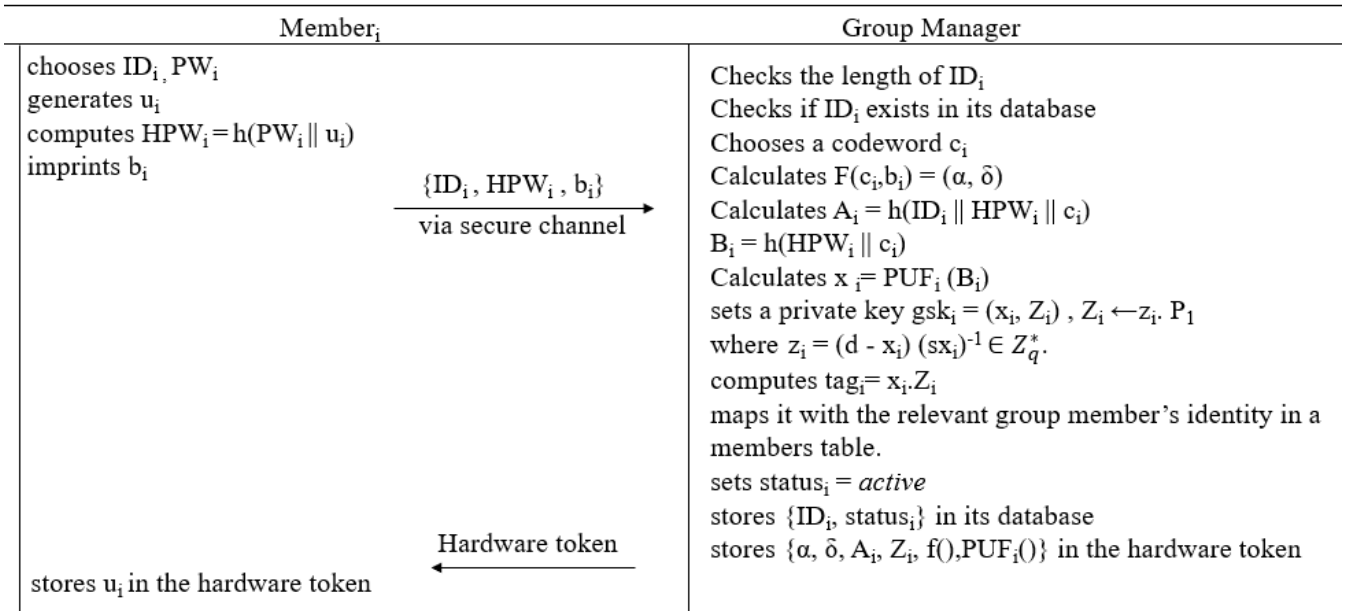

Fig. 1. Member enrollment phase.

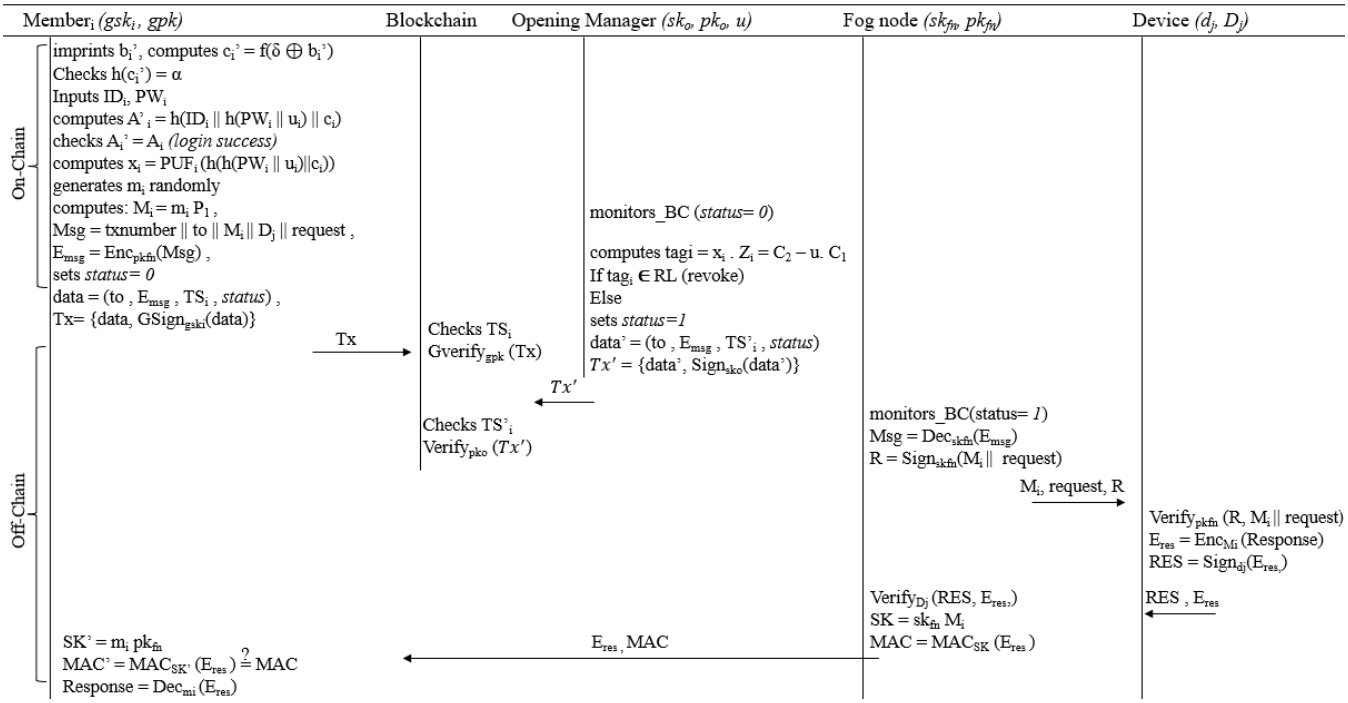

Fig. 2. Login and mutual authentication phase.

VI. Security Analysis

A. Informal Security Analysis

In this section, we provide an informal overview of the security aspects of our protocol, highlighting its robustness against various security threats and its important security features. Furthermore, we present a comparison of the proposed BC-Auth scheme with other competing schemes in terms of functionality and security features. The results of this comparison are summarized in Table I.

1) Mutual authentication: First, the consensus nodes and the opening manager authenticate the user by verifying the request transactions via group signature scheme. Then, the IoT device authenticates the fog node by verifying its digital signature (ECDSA), confirming that the fog node is authorized to access the device's data. Finally, mutual authentication is achieved between the fog node and group members through the use of MAC that provides a way to verify the authenticity and integrity of messages exchanged between the two parties. By using MAC, the fog node can authenticate the group members and vice versa, ensuring that only authorized parties can access the system.

2) Single registration: The process of single registration involves the issuance of private keys to each group member by the group manager, which is based on a unique identifier (PUF). Even when some members are revoked by the group manager, legitimate members can continue to use their private keys for signing transactions, which eliminates the need for multiple registrations and minimizes the risk associated with key management.

3) Suitable for IoT: The proposed protocol is designed to be compatible with the constraints and requirements of IoT devices, which typically have limited processing power, memory, and energy resources. To address these constraints, the proposed protocol leverages the fog node as an intermediary between IoT devices and the blockchain network, allowing IoT devices to offload some of the computation and communication tasks to the fog node. This approach reduces the computational burden on IoT devices and enables them to participate in the blockchain network securely and efficiently. Additionally, the fog node can act as a gateway for IoT devices that are not directly connected to the internet, providing them with secure and reliable access to the blockchain network. Therefore, the proposed protocol is suitable for IoT applications that require secure and efficient communication with the blockchain network.

TABLE I. Security Requirements Comparison

Security requirement	[30]	[31]	[32]	Our protocol
Anonymity	No	Yes	No	Yes
Traceability	Yes	No	Yes	Yes
Confidentiality	No	Yes	Yes	Yes
Revocation	No	Yes	No	Yes
Mutual Authentication	Yes	Yes	No	Yes
Timely Tempo detection	No	No	No	Yes

4) Resistance to cloning and counterfeiting: The proposed system uses PUF to generate members' private keys, which are then securely stored in a hardware token. PUF is a technique that leverages the inherent randomness of physical systems to generate unique and unclonable keys. By utilizing PUF, the system can ensure that the private keys of group members cannot be cloned or counterfeited, which provides a higher level of security. Even if an attacker gains access to the hardware token, they will not be able to clone or counterfeit the private key, as it is generated using PUF, which is a unique physical characteristic of the device. This resistance to cloning and counterfeiting is important because it ensures that the private keys of group members cannot be compromised, which would otherwise compromise the entire system.

5) Resistance to man in the middle attack: In the proposed protocol, a MAC is used to verify the authenticity and integrity of the exchanged messages between the communicating parties. MAC is generated by computing a cryptographic hash function over a shared secret key and the message. This shared secret key is only known to the legitimate parties, ensuring that any changes made to the message by an attacker will be detected by the receiving party. This ensures that the messages cannot be tampered with or intercepted by a malicious party without detection, preventing man-in-the-middle attacks.

6) Resistance to stolen hardware token: In the proposed system, the private keys of group members are securely generated and stored in hardware tokens using PUF. Additionally, biometric authentication is used to ensure that only the legitimate owner of the hardware token can access the private key. In the event of a hardware token being stolen, the group manager can revoke the token, rendering the private key unusable. This approach provides resistance to stolen hardware tokens and protects the system against attacks that attempt to use a stolen token to impersonate a legitimate group member.

7) Session key agreement and resistance to replay attacks: In the proposed system, session key agreement is used to establish secure communication between the fog node and the user. In every session, a fresh private/public key pair is generated for every user. This ensures that each session has a unique session key, which is used to encrypt responses and to generate a session MAC between the fog node and the user. By using a fresh private/public key pair for every session, the system resists replay attacks. If the same key is used in every session, an attacker could use a previously intercepted session key to forge or replay messages, which would compromise the security of the system. However, by using a fresh private/public key pair for every session, the system ensures that each session has a unique session key, which makes it much more difficult for an attacker to replay messages.

8) Timely tempo detection: In the proposed system, there is a login phase that locally checks the user credentials before granting access to the system. This means that the system can quickly detect and reject unauthorized users who try to access the system without valid credentials. By doing so, the system can prevent potential security breaches and minimize the

communication and computation costs. Therefore, the timely tempo detection helps to enhance the security of the system and protect it from unauthorized access.

9) Traceability: In the proposed system, group signatures are utilized to provide traceability for suspicious transactions. Only the opening manager has the ability to trace these transactions through a mechanism called GTrace. Other parties who wish to trace suspicious transactions must compromise the ElGamal encryption, which is infeasible under the current security assumptions. Thus, the use of group signatures in the system allows for traceability by authorized parties while preserving the anonymity of group members for regular transactions.

10)User anonymity: User anonymity in this context refers to the fact that the proposed system ensures that users can sign transactions without revealing their real identities. The system utilizes a group signature scheme, where a member's identity is not disclosed, and only the opening manager knows the true identity of the member who signed the transaction. Additionally, for each new transaction, a one-time public key is used, which further obscures the identity of the signer.

The comparison results in Table I show that our proposed protocol provides anonymity, traceability, confidentiality, revocation, mutual authentication, and timely tempo detection, making it a more comprehensive and robust solution for securing IoT systems than protocols [30], [31], and [32]. The lack of these features in the other protocols makes them vulnerable to security breaches, privacy violations, and man-in-the-middle attacks.

B. Formal Security Analysis using AVISPA Tool

In this section, we provide a formal security analysis of our protocol, Fig. 3, based on the widely used Automated Validation of Internet Security Protocols and Applications (AVISPA) Tool [33]. This latter takes in input a formal model of a security protocol and verifies its robustness against a set of security properties. AVISPA tool can also be used to perform security analysis on blockchain protocols. It is capable of analyzing smart contracts and consensus protocols. Avispa supports four backends for analysis: OFMC (On the fly Model Checker), CL-AtSe (Constraint-Logic-based Attack Searcher), SATMC (SAT-based Model-Checker) and TA4SP (Tree Automata based on Automatic Approximations for the Analysis of Security Protocols). Each backend has its own strengths and weaknesses.

In our analysis we adopt the OFMC and CL-AtSe backends and we use the syntax provided by the High-Level Protocol Specification Language (HLPSL) supported by AVISPA. In this HLPSL specification, we define the principal roles representing our model: group manager, group opener, group member, consensus node, fog node and IoT device. In addition, there are two composed roles, session and environment, and a goals section where the security goals are specified. As you can see in Fig. 3, the obtained analysis results show that our proposed protocol is safe under the OFMC and CL-AtSe backends.

```
% OFMC
% Version of 2006/02/13
SUMMARY
 SAFE
DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
 /home/span/span/testsuite/results/BC-Auth.if
GOAL
 as_specified
BACKEND
 OFMC
COMMENTS
STATISTICS
 parseTime:0.00s
 searchTime: 1.67s
 visitedNodes: 550 nodes
 depth: 11 plies
```

```
SUMMARY
 SAFE
DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
 TYPED_MODEL
PROTOCOL
 /home/span/span/testsuite/results/BC-Auth.if
GOAL
 As Specified
BACKEND
 CL-AtSe
STATISTICS
 Analysed : 518 states
 Reachable: 489 states
 Translation: 0.02 seconds
 Computation: 0.35 seconds
```

Fig. 3. Analysis result using OFMC and CL-AtSe backends.

VII. Conclusion

In this paper, we proposed a blockchain-based secure mutual authentication system for IoT using PUFs and group signatures. The proposed BC-Auth scheme provides several advantages, including the use of permissioned blockchain for more control and privacy, a group signature scheme with two authorities for enhanced security, three-factor authentication for stronger user verification, PUFs for unique private key generation, and fog computing for improved security and efficiency. The proposed system aims to strengthen IoT security and efficiency by mitigating risks and making it more difficult for attackers to obtain user credentials.

We also proved the security of our protocol informally and formally using the AVISPA tool and provided a security comparison with other blockchain based authentication protocols.

In our future work, we are working on:

1) The practical implementation of the proposed BC-Auth protocol using the Hyperledger Fabric which is a permissioned blockchain platform designed for enterprise use cases with modular architecture and privacy features,

2) The practical simulation of the proposed BC-Auth protocol to prove its performance in terms of computational and communication costs.

3) Evolving our protocol by proposing a blockchain based multiple managers' group signature scheme to avoid the risks related to the single authority in the case of one group manager model.

References

[1] S. Selvaraj and S. Sundaravaradhan, "Challenges and opportunities in IoT healthcare systems: a systematic review," SN Applied Sciences, vol. 2, no. 1. Springer Nature, Jan. 01, 2020, doi: 10.1007/s42452-019-1925-y.

[2] P. Fraga-Lamas, T. M. Fernández-Caramés, M. Suárez-Albela, L. Castedo, and M. González-López, "A Review on Internet of Things for Defense and Public Safety," Sensors (Basel, Switzerland), vol. 16, no. 10. Oct. 05, 2016, doi: 10.3390/s16101644.

[3] K. Szum, "IoT-based smart cities: A bibliometric analysis and literature review," Eng. Manag. Prod. Serv., vol. 13, no. 2, pp. 115–136, Jun. 2021, doi: 10.2478/emj-2021-0017.

[4] M. A. Ferrag, L. Shu, X. Yang, A. Derhab, and L. Maglaras, "Security and Privacy for Green IoT-Based Agriculture: Review, Blockchain Solutions, and Challenges," IEEE Access, vol. 8. Institute of Electrical

and Electronics Engineers Inc., pp. 32031–32053, 2020, doi: 10.1109/ACCESS.2020.2973178.

[5] M. Aqeel, F. Ali, M. W. Iqbal, T. A. Rana, M. Arif, and M. R. Auwul, "A Review of Security and Privacy Concerns in the Internet of Things (IoT)," J. Sensors, vol. 2022, pp. 1–20, Sep. 2022, doi: 10.1155/2022/5724168.

[6] P. P. Ray, "A survey on Internet of Things architectures," Journal of King Saud University - Computer and Information Sciences, vol. 30, no. 3. King Saud bin Abdulaziz University, pp. 291–319, Jul. 01, 2018, doi: 10.1016/j.jksuci.2016.10.003.

[7] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," Comput. Networks, vol. 54, no. 15, pp. 2787–2805, 2010, doi: 10.1016/j.comnet.2010.05.010.

[8] L. Babun, K. Denney, Z. B. Celik, P. McDaniel, and A. S. Uluagac, "A survey on IoT platforms: Communication, security, and privacy perspectives," Computer Networks, vol. 192. Elsevier B.V., Jun. 19, 2021, doi: 10.1016/j.comnet.2021.108040.

[9] N. Sharma, M. Shamkuwar, and I. Singh, "The history, present and future with iot," in Intelligent Systems Reference Library, vol. 154, Springer Science and Business Media Deutschland GmbH, 2019, pp. 27–51.

[10] S. Tanwar, N. Gupta, C. Iwendi, K. Kumar, and M. Alenezi, "Next Generation IoT and Blockchain Integration," Journal of Sensors, vol. 2022. Hindawi Limited, 2022, doi: 10.1155/2022/9077348.

[11] A. Panarello, N. Tapas, G. Merlino, F. Longo, and A. Puliafito, "Blockchain and iot integration: A systematic survey," Sensors (Switzerland), vol. 18, no. 8. MDPI AG, Aug. 06, 2018, doi: 10.3390/s18082575.

[12] L. Lao, Z. Li, S. Hou, B. Xiao, S. Guo, and Y. Yang, "A survey of IoT applications in blockchain systems: Architecture, consensus, and traffic modeling," ACM Computing Surveys, vol. 53, no. 1. Association for Computing Machinery, Feb. 01, 2020, doi: 10.1145/3372136.

[13] H. Sabireen and V. Neelanarayanan, "A Review on Fog Computing: Architecture, Fog with IoT, Algorithms and Research Challenges," ICT Express, vol. 7, no. 2, pp. 162–176, Jun. 2021, doi: 10.1016/j.icte.2021.05.004.

[14] O. Umoren, R. Singh, S. Awan, Z. Pervez, and K. Dahal, "Blockchain-Based Secure Authentication with Improved Performance for Fog Computing," Sensors, vol. 22, no. 22, Nov. 2022, doi: 10.3390/s22228969.

[15] F. Sabrina, N. Li, and S. Sohail, "A Blockchain Based Secure IoT System Using Device Identity Management," Sensors, vol. 22, no. 19, Oct. 2022, doi: 10.3390/s22197535.

[16] G. Cheng, Y. Chen, S. Deng, H. Gao, and J. Yin, "A Blockchain-Based Mutual Authentication Scheme for Collaborative Edge Computing," IEEE Trans. Comput. Soc. Syst., vol. 9, no. 1, pp. 146–158, Feb. 2022, doi: 10.1109/TCSS.2021.3056540.

[17] C. Huang and K. Yan, "A Blockchain Based Fast Authentication Framework for IoT Networks with Trusted Hardware," in Proceedings - 2020 IEEE 22nd International Conference on High Performance Computing and Communications, IEEE 18th International Conference on Smart City and IEEE 6th International Conference on Data Science and Systems, HPCC-SmartCity-DSS 2020, Dec. 2020, pp. 1050–1056, doi: 10.1109/HPCC-SmartCity-DSS50907.2020.00141.

[18] S. C. Cha, J. F. Chen, C. Su, and K. H. Yeh, "A Blockchain Connected Gateway for BLE-Based Devices in the Internet of Things," IEEE Access, vol. 6, pp. 24639–24649, Jan. 2018, doi: 10.1109/ACCESS.2018.2799942.

[19] M. Yavari, M. Safkhani, S. Kumari, S. Kumar, and C. M. Chen, "An Improved Blockchain-Based Authentication Protocol for IoT Network Management," Secur. Commun. Networks, vol. 2020, 2020, doi: 10.1155/2020/8836214.

[20] Y. Hu, D. Yin, and C. Huang, "BBSF: A blockchain based secure framework for the internet of things with user revocation," in Proceedings of 2020 IEEE International Conference on Progress in Informatics and Computing, PIC 2020, Dec. 2020, pp. 358–362, doi: 10.1109/PIC50277.2020.9350772.

[21] H. H. Pajooh, M. Rashid, F. Alam, and S. Demidenko, "Multi-layer blockchain-based security architecture for internet of things," Sensors (Switzerland), vol. 21, no. 3, pp. 1–26, Feb. 2021, doi: 10.3390/s21030772.

[22] D. Li, W. Peng, W. Deng, and F. Gai, A Blockchain-based Authentication and Security Mechanism for IoT, 2018 27th International Conference on Computer Communication and Networks (ICCCN). IEEE, 2018.

[23] A. M. Antonopoulos, Mastering Bitcoin : unlocking digital crypto-currencies. 2015.

[24] M. Castro, M. Research, and B. Liskov, "Practical Byzantine Fault Tolerance and Proactive Recovery," 2002.

[25] D. C. Eugtne van Heyst, "Group Signatures," 1991.

[26] T. H. Ho, L. H. Yen, and C. C. Tseng, "Simple-Yet-Efficient Construction and Revocation of Group Signatures," Int. J. Found. Comput. Sci., vol. 26, no. 5, pp. 611–624, Aug. 2015, doi: 10.1142/S0129054115500343.

[27] A. Juels and M. Wattenberg, "Fuzzy commitment scheme," in Proceedings of the ACM Conference on Computer and Communications Security, 1999, pp. 28–36, doi: 10.1145/319709.319714.

[28] C. Herder, M. D. Yu, F. Koushanfar, and S. Devadas, "Physical unclonable functions and applications: A tutorial," Proc. IEEE, vol. 102, no. 8, pp. 1126–1141, 2014, doi: 10.1109/JPROC.2014.2320516.

[29] D. Hankerson, Menezes Alfred, and Vanstone Scott, Guide to Elliptic Curve Cryptography. 2004.

[30] M. T. Hammi, B. Hammi, P. Bellot, and A. Serhrouchni, "Bubbles of Trust: A decentralized blockchain-based authentication system for IoT," Comput. Secur., vol. 78, pp. 126–142, Sep. 2018, doi: 10.1016/j.cose.2018.06.004.

[31] C. Lin, D. He, X. Huang, K. K. R. Choo, and A. V. Vasilakos, "BSeIn: A blockchain-based secure mutual authentication with fine-grained access control system for industry 4.0," J. Netw. Comput. Appl., vol. 116, pp. 42–52, Aug. 2018, doi: 10.1016/j.jnca.2018.05.005.

[32] R. Li, T. Song, B. Mei, H. Li, X. Cheng, and L. Sun, "Blockchain for Large-Scale Internet of Things Data Storage and Protection," IEEE Trans. Serv. Comput., vol. 12, no. 5, pp. 762–771, Sep. 2019, doi: 10.1109/TSC.2018.2853167.

[33] A. Armando et al., "The AVISPA tool for the automated validation of internet security protocols and applications," in Lecture Notes in Computer Science, 2005, vol. 3576, pp. 281–285, doi: 10.1007/11513988_27.