# Private and Consortium Blockchain-based Authentication Protocol for IoT Devices Using PUF

Tyson Baptist D Cunha and Kiran Manjappa

***Abstract*—In this work, a static random access memory-physical unclonable function (SRAM-PUF) based device security framework is proposed which uses the trending blockchain technology for securing the device credentials. The proposed framework produces a unique fingerprint called *PUF key* for each device based on its hardware characteristics which will act as an authenticating parameter for the devices during the authentication and re-authentication phase. The proposed work uses both consortium and private blockchains for storing device credentials and authentication, unlike the current trend of using either a secured database or only a public blockchain. The consortium blockchain is used for first-time authentication, while the private blockchain is used for repeated authentication which saves the time incurred in accessing the consortium blockchain during repeated authentication. The proposed protocol also includes mutual authentication between the entities involved and thus provides dual security (device authentication and mutual authentication) to the proposed protocol making the system more secure and robust against attacks. Security analysis of the proposed protocol is done using the Scyther tool and the protocol is also theoretically proven to be stable under various attacks using threat analysis and the real-or-random model (ROR). The performance analysis of the protocol is done by analyzing the computation and communication cost of the proposed protocol against other state-of-the-art protocols. Further, the proposed protocol is also evaluated in the blockchain testbed which includes Raspberry PI and Arduino components. The results conveyed that the introduction of a private blockchain reduces the time incurred in the device re-authentication.**

***Index Terms*—Authentication, blockchain, PUF, ROR, Scyther, SRAM.**

## I. Introduction

WITH increased Internet of things (IoT) devices and their connectivity, the challenges of maintaining, authenticating, and securing these devices have become a subject of concern nowadays. Further, maintaining the integrity of the data associated with these devices has also become a big challenge that can not be ignored [1]. IoT devices need to authenticate themselves first before getting connected to a network and providing its services. But, with advancements in technology, many malicious practices of counterfeiting devices and fake authentication to steal confidential data from networks have been noticed in the real world [2]. Different authentication schemes for IoT devices can be found in the literature [3], [4] that use cryptographic functions making them both resource and power-hungry. Hence, such cryptography-based authentication schemes do not suit these low-computation, low-power IoT devices. Thus, a low computation cost authentication scheme is in demand for these resource-constrained IoT devices [5] to authenticate themselves to the network. The physical unclonable functions (PUFs), a hardware authentication scheme, that requires low computation cost and consumes low power, has come to the rescue of the IoT devices.

### A. Physical Unclonable Function (PUF)

PUF [6], a function that resides in the device, takes a set of different challenges as input and generates a set of different responses as the output. These are called challenge-response pairs (CRPs) of the corresponding device and CRPs will be stored in a secured place for future reference. Whenever the authenticity of the device is required, a random challenge from the stored CRPs can be given as input to the PUF function residing in the device. Against the received input, the PUF takes the response, $r^{'}$, from the device and sends it to the calling function. If the received response $r^{'}$ matches with the stored response $r$, then the device authenticity is proved. On the other hand, if an adversary has replaced the internal component of the device with a cloned component, or an internal component of the device has been removed from the device or the device itself has been cloned, a CRP check for that device will definitely fail because the response $r^{'}$ from the altered device will be different than the $r$ which is from the original device. Thus, PUF can be used during any stage of the device life cycle to check the authenticity of the device. Each device will exhibit uniqueness in responses even if the same challenge is given as input at different time intervals. This uniqueness is mainly due to the unpredictable properties of the hardware that occur during the manufacturing process of the device. Even the manufacturer cannot control these exact variations as it is a unique threshold voltage for each transistor on a chip. Also, as CRPs are unique and unclonable, they can act as an unique identifier (ID), or a digital fingerprint, for device authentication over an unsecured channel.

Since these generated device IDs play a vital role in verifying the authenticity of the device, securing them from unauthorized access is another challenge one should look at. Fig. 1 shows the two approaches for storing these CRPs securely namely, a secured database approach and a blockchain approach. Fig. 1(a) shows the secured database approach,

Manuscript received May 4, 2023; approved for publication December 22, 2023; approved for publication by WU, YIK CHUNG Division 1 Editor, Febuary 27, 2024.

T. B. D Cunha and K. Manjappa are with National Institute of Technology Karnataka Ringgold Standard Institution - Information Technology, India, emails: dcunhatyson@gmail.com, kiranmanjappa@nitk.edu.in.

K. Manjappa is the corresponding author.

Digital Object Identifier: 10.23919/JCN.2024.000014



1229-2370/24/$10.00 © 2024 KICS

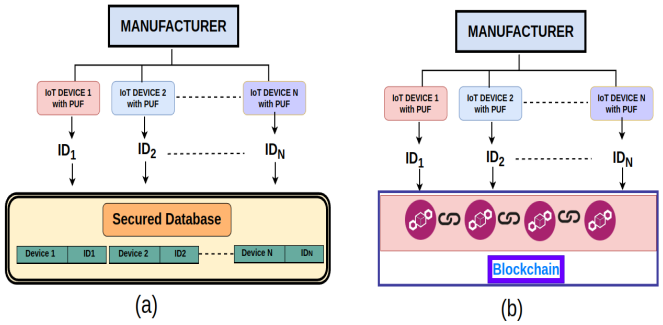


Fig. 1. (a) Authentication using secured database approach and (b) authentication using blockchain approach.

where the generated device IDs are stored securely in a database but, it has a threat of a single point of failure (SPoF) and a single point of attack (SPA) [7]. Usually, trusted third parties are hired to maintain these databases but, the trustiness of the trusted third parties is always a concern. On the other hand, blockchain technology can be the best solution for storing these device IDs securely as well as for device authentication as shown in Fig. 1(b). Blockchain technology is today's trend in data security and access restriction [8]. In the blockchain approach, the generated device IDs are stored securely in the blockchain which promises transparency, integrity, and verifiability. The access restriction property of the blockchain does not allow unauthorized access to the device IDs while SPoF and SPA are avoided by the distributed and decentralized nature of the blockchain. However, most of the articles found in the literature use the public blockchain for storing these device IDs. Another approach found in the literature is Consortium blockchain [9], [10] which is a semi-decentralized approach that consists of multiple organizations (blockchains) with a predefined set of nodes and rules to achieve a common motive or requirement. A consortium blockchain can be used to build a permission-based private system that gives control over which blockchain data is made public or private and to whom.

The main drawback of the public blockchain is its slower transaction processing time. Because of its large size, the public blockchain may introduce a considerable amount of delay in the process [11], [12]. Thus, the frequent retrieval of the device ID from the public blockchain for repeated authentication may hinder the process. Especially when the device needs to be re-authenticated within a short span of time. Furthermore, if the application is time-critical, the delay introduced by the public blockchain during the frequent authentication may cause serious issues.

Considering the above-discussed points, the authors of this article have proposed a novel technique where both consortium and private blockchains are used for device authentication. In the proposed methodology, the device is authenticated using the consortium blockchain, and if the authentication is successful, its credentials are stored locally in the private blockchain. The private blockchain is preferred for re-authentication as it is small in size with limited nodes making it fast and flexible [13] [14]. Thus, when the same device should be authenticated next time, the credentials are retrieved from the private blockchain rather than from the consortium blockchain, which saves credentials retrieve time considerably. We have used the static random access memory-physical unclonable function (SRAM-PUF)-based device ID generation algorithm in the proposed authentication protocol which uses stable SRAM bits to generate a unique device ID. In addition, a repeated ID matching technique is also used to ensure the reliability and correctness of the generated device ID.

***The major contributions of this article are***

1) This article introduces a novel device authentication and re-authentication protocol for IoT devices using minimum cryptographic primitives.
2) A distinctive communication security protocol is also proposed in this work which secures the communication between the entities involved by achieving mutual authentication.
3) Along with the threat model, the proposed protocol is also formally analyzed and verified on a physical testbed for its correctness.

The rest of the article is organized as follows. Section II presents the related work and Section III explains the working scenario of the proposed PUF and blockchain based authentication protocol along with the algorithms for registration, authentication, and re-authentication phase. The theoretical proof of the proposed protocol as well as the Scyther results are also discussed in the same section. Section IV provides experimental results and the threat analysis followed by conclusions in Section V.

## II. Related Work

The authentication protocols present in the literature can be categorized as PUF based [15]–[25], blockchain based [26]–[32], and PUF and blockchain based [33]–[48] according to the state-of-the-art. In this section, only PUF and blockchain based authentication protocols are discussed which are undeniably robust, immutable and secured in terms of privacy, integrity, and confidentiality of both the device and data. In [33], authors have used SRAM PUF, bio-metrics, and digital signatures as device credentials for the authentication process. A PUF key is generated using SRAM PUF during the registration phase, and a fuzzy extractor is used as an error correction code to manage the unstable bits of SRAM. Authors have used ledger as a secured way of storing the device credentials. In [34], the authors have used a global blockchain to store the device-specific credentials. Each device embedded with SRAM PUF will have a public ID given by the manufacturer. The single challenge-response pair generated by PUF for each device is hashed to get a secret ID for the device. This secret ID along with the public ID of the device and manufacturer ID uniquely identifies the device and is stored for future reference in the blockchain. In [35], authors have proposed an authentication protocol for ownership transfer that uses consortium blockchain for storing the integrated circuit (IC) credentials securely. While in [36], [37], the authors have used PUF and Chinese remainder theorem for

the device authentication with the help of the blockchain. The authors have used Diffie-Hellman key agreement protocol for the communication between IoT devices and the blockchain. In [38], the authors have proposed a public blockchain-based ownership transfer protocol that aims at tracking and tracing of ICs in supply chain management (SCM) and in [39], a consortium blockchain-based authentication protocol has been proposed that consists of three types of nodes in a blockchain namely the initiator node, trusted node, and simple node each with different responsibilities at different capacities. In [40], the authors have proposed PUF-Chain, a private blockchain and PUF-based authentication protocol which uses both the secured database and the blockchain for the device authentication where the authors assumed that the content of the database is secured. In [41], the authors have proposed PUF-Chain 2.0 for the Internet of medical things (IoMT) device authentication in smart healthcare. PUF-Chain 2.0 uses MAC address of the device along with the PUF key for enhanced security of the devices.

In [42], the authors have proposed PUF and blockchain-based authentication protocol for the Internet of things-ambient intelligence (IoT-AmI) environment in healthcare. In [43], the authors have proposed a PUF and blockchain based authentication protocol for medical wireless sensor networks. The authors have used AVISPA tool for formal security analysis. Based on the cost analysis, the authors have concluded that the proposed protocol is computationally expensive only at GWN end. In [44], the authors have proposed PUF and blockchain based multi-server authentication protocol for cloud-Edge based IoT devices. In [45], the authors have proposed an authentication protocol which mitigates IC counterfeits and uniquely tracks ICs in the supply chain using PUF and consortium blockchain. In [46], the authors have proposed a decentralized authentication protocol using PUF and blockchain which uses CRPs and helper data, a string consisting of robust bits string and uniform random responses for a given challenge, to authenticate the device.

In [47], the authors have proposed blockchain and PUF based authentication protocol for industrial Internet of things (IIoT) devices. In [48], the authors have proposed blockchain and PUF based authentication protocol to safeguard the devices against the untoward activities in the supply chain. The proposed system uses radio frequency identifier (RFID) and blockhain network to authenticate the devices.

### A. Observations

The surveyed articles are summarized in Tables II, III, and IV while Table I shows the nomenclatures of different parameters considered for the comparison. Table II shows the PUF-based authentication protocols studied in the literature along with the cryptographic parameters considered and the type of PUF used in articles. The PUF-based authentication protocols aim at device security and anti-cloning techniques to secure devices from being cloned or duplicated. Table III on the other hand, showcases different blockchain-based authentication protocols with similar cryptographic parameters along with different types of blockchains. The blockchain-based protocols mainly aim to secure the device data or credentials that will be used for authentication. Finally, Table IV showcases protocols that combine both PUF and Blockchain techniques where both the device and data or credentials associated with the device are secured. Based on these comparisons, the following observations are made,

TABLE I
NOMENCLATURE.

| | |
|---|---|
| $C$ | Consortium blockchain |
| $DS$ | Digital signatures |
| $E$ | Error correction code |
| $H$ | Hash function |
| $M$ | Mathematical notions and key exchange |
| $PU$ | Public blockchain |
| $PR$ | Private blockchain |
| $S$ | Strong PUF |
| $W$ | Weak PUF |

1) To the best of authors knowledge, no prior work is found in the literature which uses a combination of consortium and private blockchains along with PUF for IoT device authentication and re-authentication.
2) To the best of our knowledge, only a few research attempts [33], [36], [42], [44], have been made that authenticates devices as well as secure the credentials. Most of the articles found in the literature concentrate on either device authentication or credentials security and not on both.
3) Some of the articles found in literature, [40], [41], [46] have used a secured database along with blockchain to store the authentication credentials of the device. There are high chances for an adversary to hack the database and manipulate or steal the data. If credentials are known to the adversary, then the entire process of authentication fails.

Considering all these observations, in this article, the authors have proposed a novel consortium and private blockchain-based authentication protocol for IoT devices. The proposed approach uses PUF based unique device ID and secret key $K$ to authenticate IoT devices. The consortium blockchain is used for registration and first-time authentication of device whereas the private blockchain is used for repeated authentication of the IoT device. The proposed approach secures the communication between the device and the authenticator node using the symmetric network key. Further, the encrypted form of the credentials is stored in the blockchain making the proposed protocol more secure and robust. For the evaluation, the authors have considered a physical test bed scenario consisting of Raspberry PI and Arduino mega devices. The proposed approach is analyzed against several attacks using a threat model and through the Scyther tool results, the proposed protocol is shown to be secured.

## III. Proposed PUF and Blockchain based Authentication Protocol

The proposed approach consists of an IoT device, a trusted manufacturer of IoT devices, consortium and private blockchain in a network. The network in turn consists of an authenticator node which runs the authentication protocol.

The proposed approach consists of two phases, namely a device registration (enrollment) phase and an authenti-

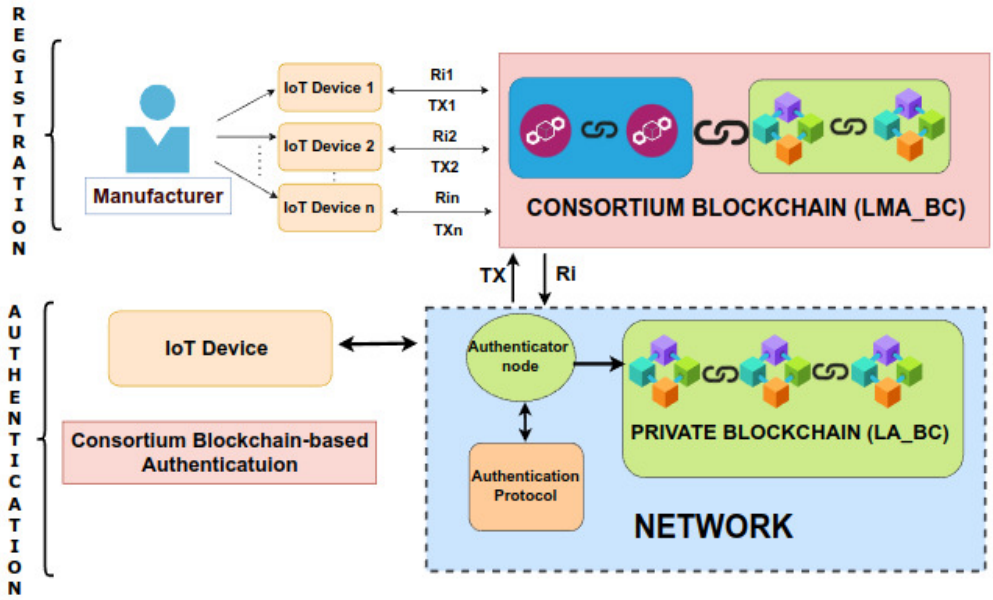


Fig. 2. Proposed registration and authentication approach.

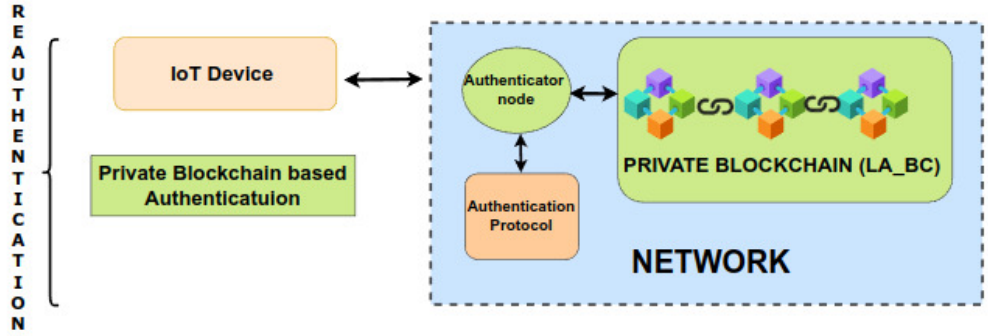


Fig. 3. Proposed re-authentication approach.

TABLE II
PUF BASED AUTHENTICATION PROTOCOLS.

| Paper | Techniques | | | | PUF Types | |
|---|---|---|---|---|---|---|
| | E | DS | H | M | S | W |
| [15] | | | ✓ | ✓ | | ✓ |
| [16] | | | | ✓ | | ✓ |
| [17] | ✓ | | | | | ✓ |
| [18] | | | | ✓ | | ✓ |
| [19] | | | | ✓ | ✓ | |
| [20] | | | ✓ | ✓ | | ✓ |
| [21] | ✓ | | ✓ | ✓ | | ✓ |
| [22] | | ✓ | | ✓ | | ✓ |
| [23] | | | | | ✓ | |
| [24] | | | | ✓ | ✓ | |
| [25] | | | ✓ | ✓ | ✓ | |

TABLE III
BLOCKCHAIN BASED AUTHENTICATION PROTOCOLS.

| Paper | Techniques | | | | Blockchain | | |
|---|---|---|---|---|---|---|---|
| | E | DS | H | M | PU | C | PR |
| [26] | | ✓ | | ✓ | | | ✓ |
| [27] | | ✓ | ✓ | ✓ | | | ✓ |
| [28] | | ✓ | ✓ | | ✓ | | |
| [29] | | ✓ | ✓ | ✓ | ✓ | | |
| [30] | | ✓ | | ✓ | | | ✓ |
| [31] | | ✓ | ✓ | ✓ | | | ✓ |
| [32] | | ✓ | ✓ | ✓ | | | ✓ |

cation phase, as shown in Fig. 2. The device registration is a one-time process in which the registered manufacturer adds a newly manufactured device to the consortium blockchain namely legitimate manufacturers and authenticators blockchain ($LMA\_BC$), with secret credentials (an encrypted device ID and secret key in the form of *Ri*) which are generated using *ID extraction* algorithm based on SRAM PUF [15]. After the successful registration of devices, the consortium blockchain gives a corresponding transaction hash (*TX*) through which the same *Ri* can be relocated in the consortium blockchain during the authentication phase. The authentication phase is initiated when a device tries to interact with the network, and this phase deals with the verification of the secret credentials (*KDID, K, PUF key, TX*) of the IoT devices that were collected during the registration phase. The

TABLE IV
PUF AND BLOCKCHAIN BASED AUTHENTICATION PROTOCOLS.

| Paper | Techniques | | | | PUF | | Blockchain | | |
|---|---|---|---|---|---|---|---|---|---|
| | E | DS | H | M | S | W | PU | C | PR |
| [33] | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ | | |
| [34] | ✓ | | ✓ | ✓ | | ✓ | ✓ | | |
| [35] | | ✓ | ✓ | ✓ | | ✓ | | ✓ | |
| [36] | | | ✓ | ✓ | ✓ | | ✓ | | |
| [37] | | | ✓ | ✓ | ✓ | | ✓ | | |
| [38] | | | | ✓ | | ✓ | ✓ | | |
| [39] | | ✓ | | ✓ | | ✓ | | | ✓ |
| [40] | | | ✓ | ✓ | ✓ | | | ✓ | |
| [41] | | | ✓ | ✓ | ✓ | | | ✓ | |
| [42] | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | | |
| [43] | ✓ | | | ✓ | ✓ | | | | ✓ |
| [44] | ✓ | | | | ✓ | | ✓ | | |
| [45] | | ✓ | ✓ | ✓ | ✓ | | | ✓ | |
| [46] | | | ✓ | ✓ | | ✓ | ✓ | | |
| [47] | | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | |
| [48] | | | | ✓ | ✓ | | | ✓ | |

authentication protocol is initiated by the authenticator node which acts as an authenticator or mediator between the device and the network. Upon successful authentication, the device credentials are encrypted and stored in the private blockchain namely legitimate authenticators's blockchain ($LA\_BC$) for future re-authentication purposes. The proposed protocol, uses only private blockchain ($LA\_BC$), avoiding the consortium blockchain ($LMA\_BC$) when the same device comes for re-authentication as shown in Fig. 3. The $LMA\_BC$ consists of two organizations namely legitimate manufacturers (LM) and legitimate authenticators (LA) and it is explicitly designed for manufacturers to register and authenticate their manufactured devices $IoD_i$ where $i = 1, 2, 3, \cdots, \mathbb{Z}$. Only the LMs are allowed to participate in consensus and add or endorse data (*Ri*) in $LMA\_BC$. Whereas LAs can only view or query the ledger data from $LMA\_BC$ for authentication of registered devices. On the other hand, the private blockchain of legitimate authenticators, $LA\_BC$, is used explicitly for re-authentication and its ledger data is restricted only to LAs. Since $LA\_BC$ contains credentials of already authenticated devices $IoD_j$ where $j = 1, 2, 3, \cdots, \mathbb{Z}^{'}$,and $\mathbb{Z}^{'} \subset \mathbb{Z}$, it speeds up the re-authentication process and reduces re-authentication time. Thus, both the consortium and private blockchains together increase the efficiency of the proposed protocol. In the following subsections, the registration and authentication, and re-authentication phases are discussed in detail. The terms used in the proposed authentication protocol are detailed in Table V.

The *ID extraction* algorithm using SRAM PUF [15] is used to generate unique device ID i.e., *PUFKeyE* and the random number generator is used to generate secret key *K* as secret credentials of the *IoD*. The same device ID will be regenerated when required and verified using a repeated ID matching technique.In order to generate a unique and cost-effective unclonable device ID for each IoT device, an on-chip SRAM present in the IoT device can be used. A small number of CRPs are sufficient to generate a unique device ID for each device, hence, a weak PUF can be preferred over a strong PUF. However, generating device ID using SRAM has many challenges. There is a possibility of the generation of different

TABLE V
TERMS USED.

| Variables list | |
|---|---|
| Variable | Meaning |
| $A_N$ | Authenticator node |
| $A_{NID}$ | Authenticator node ID |
| *Ai* | PUFKEY specific secret key |
| *ACK* | Acknowledgement |
| *DID* | Device specific ID |
| *H* | Hash function |
| $H_{DID}$ | Hash of DID |
| $H_{nd}$ | Hash of nonce value at the device end |
| $H_{na}$ | Hash of nonce value at authenticator end |
| *IoD* | IoT device |
| *K* | Secret key given by the manufacturer |
| *KDID* | Secret key based DID |
| *Ki* | K specific secret key |
| $M$ | Manufacturer of IoT device |
| *Mi* | Nonce and *KDID* specific secret key |
| $M_{ID}$ | Manufacturer ID |
| $N_{an}$ | Nonce |
| *NWK* | Network key |
| *LA* | Legitimate authenticators |
| *LA_BC* | Legitimate authenticator's blockchain (private) |
| *Li* | PUFKeyRA and KDID specific secret key |
| *LMA_BC* | Legitimate manufacturer's and authenticator's blockchain (consortium) |
| *LM* | Legitimate manufacturers |
| *Pi* | PUFKeyA and $H_{DID}$ specific secret key |
| *PUFKeyE* | PUFKey generated during registration phase |
| *PUFKeyA* | PUFKey re-generated during authentication phase |
| *PUFKeyRA* | PUFKey re-generated during re-authentication phase |
| *Ri* | Device specific public key generated by manufacturer |
| $SK_{IA}$ | Session key generated between $IoD$ and $A_N$ after authentication |
| $SKR_{IA}$ | Session key generated between $IoD$ and $A_N$ after re-authentication |
| *TX* | Transaction hash generated by consortium blockchain during registration of IoT device |
| *TXr* | Transaction hash generated by private blockchain during authentication of IoT device |

IDs for the same device due to instability in SRAM bits. This possibility can be avoided using an error correction code. But error correction codes require more computation and consume large memory. Thus, an error correction code may not be a feasible option for resource-constrained IoT devices. Instead, a repeated ID matching technique [15] can be used to avoid the instability of SRAM which does not require much space and computation.

### A. Registration Phase

The proposed registration phase is shown in Fig. 4. The steps involved in the registration phase are detailed below:

1) $M$ generates *PUFKeyE* and secret key *K* from an *IoD* using *ID extraction algorithm* and random number generator. From the *PUFKeyE* and *K*, the $M$ generates *Ri*, a device-specific key, for each device using the (1). $M$ then sends the device specific key, *Ri*, to consortium blockchain for the *IoD* registration. The $M$ should be a legitimate and registered identity in the blockchain to register the *IoD*.

$$Ri = PUFKeyE \oplus K \tag{1}$$

**Algorithm 1:** Device registration phase

*IoD* generates *PUFKeyE* using *ID extraction algorithm* and *K* using random number generator
*IoD* sends *PUFKeyE, K* to $M$
$M$ sends *Ri* to consortium blockchain
Consortium blockchain generates *TX* for $Ri$ and stores in blockchain
Consortium blockchain sends *TX* to $M$
$M$ computes *KDID*= $M_{ID} \oplus K$
$M$ sends *TX, KDID* to *IoD*
*IoD* stores *TX, KDID, K*

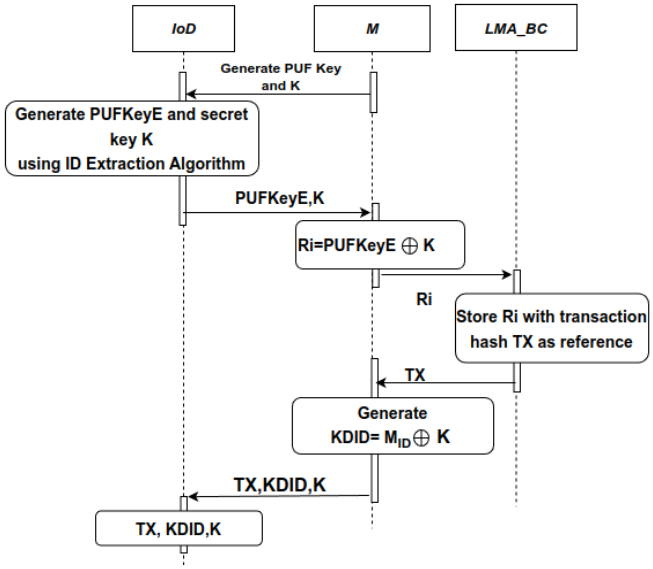


Fig. 4. Registration phase.

2) After verifying the credentials of the $M$, the consortium blockchain registers the *IoD* by adding *Ri* in to the blockchain. This registration process generates a transaction hash *TX*, which is specific to the registered *IoD*. As a reply, the consortium blockchain sends *TX* to the $M$.
3) $M$ will also have manufacturer specific built-in ID called $M_{ID}$. The *M* generates a secret key based device ID, known as *KDID* , using $M_{ID}$ and *K* as shown in (2). The *KDID* is generated to encrypt and protect secret key *K* from adversaries and provide a security environment to *K* in private blockhain. Finally, the *M* stores the secret key *K, KDID* and *TX*, in the *IoD* for future authentication.

$$KDID = M_{ID} \oplus K \tag{2}$$

The pseudo code for *IoD* registration phase is shown in *Algorithm 1*.

### B. Authentication Phase

The authentication phase is triggered when an IoD tries to join or communicate with the service provider network. The IoD first contacts the $A_N$ to initiate the authentication process. The $A_N$ generates a session and device-specific network key [21], *NWK*, and shares it with the *IoD* using secure channel in order to secure future communication. The authentication phase is shown in Fig. 5, and the steps involved are explained below:

1) The *IoD* encrypts *KDID* stored in its memory using the network key *NWK*, and sends it to the $A_N$ for the authentication.
2) The $A_N$ then decrypts the received encrypted message using *NWK* and extracts *KDID*. Now, KDID acts as an authenticating parameter for the IoD, and since storing *KDID* directly in the private blockchain (after successful authentication leads to vulnerability, we are creating another ID referred to as a device-specific ID (DID) based on *KDID*. The DID will be created by the $A_N$ using its built-in ID called $A_{NID}$ as shown in (3).

$$DID = KDID \oplus A_{NID} \tag{3}$$

3) In order to provide one more level of security, the DID is hashed to get $H_{DID}$ which will be actually stored in private blockchain (upon successful authentication) and the same will be used in the re-authentication phase. The generation of $H_{DID}$ is shown in (4).

$$H_{DID} = H(DID) \tag{4}$$

4) The $A_N$ uses a nonce $N_{an}$ to obtain a secret key *Mi* from *KDID* as shown in (5). The $N_{an}$ is basically used as a second factor of security to device credentials after *NWK*. The $N_{an}$ will be used to encrypt device credentials like *K* and *PUFKeyA*. Thus, securing the credentials during the communication at both $A_N$ and at *IoD* end.

$$Mi = KDID \oplus N_{an} \tag{5}$$

5) The *IoD* decrypts the received message using *NWK* and retrieves $N_{an}$ using (6).

$$N_{an} = Mi \oplus KDID \tag{6}$$

The retrieved $N_{an}$ is then hashed using hash function *H* to get $H_{nd}$. This process is shown in (7). The main reason for using the hash function is that if $N_{an}$ is retrieved by adversaries, then the device credentials can be decrypted using $N_{an}$, thus putting the entire system at risk. Hence, $N_{an}$ is hashed to protect device credentials from adversaries.

$$H_{nd} = H(N_{an}) \tag{7}$$

For the consortium blockchain authentication, the *IoD* regenerates PUF key i.e., *PUFkeyA* using credentials obtained during *ID extraction algorithm*. Then the *IoD* computes a secret key *Ai* using *PUFkeyA* and $H_{nd}$ and *Ki* using *K* and $H_{nd}$ as shown in (8).

$$Ai = H_{nd} \oplus PUFKeyA$$
$$Ki = H_{nd} \oplus K \tag{8}$$

The $Ai$ and $Ki$ are generated by $IoD$ to provide communication security and to safeguard $PUFKeyA$ and $K$ from adversaries. Then the *IoD* generates a secret session key $SK_{IA}$ using (9). The $SK_{IA}$ will also be genrated at the

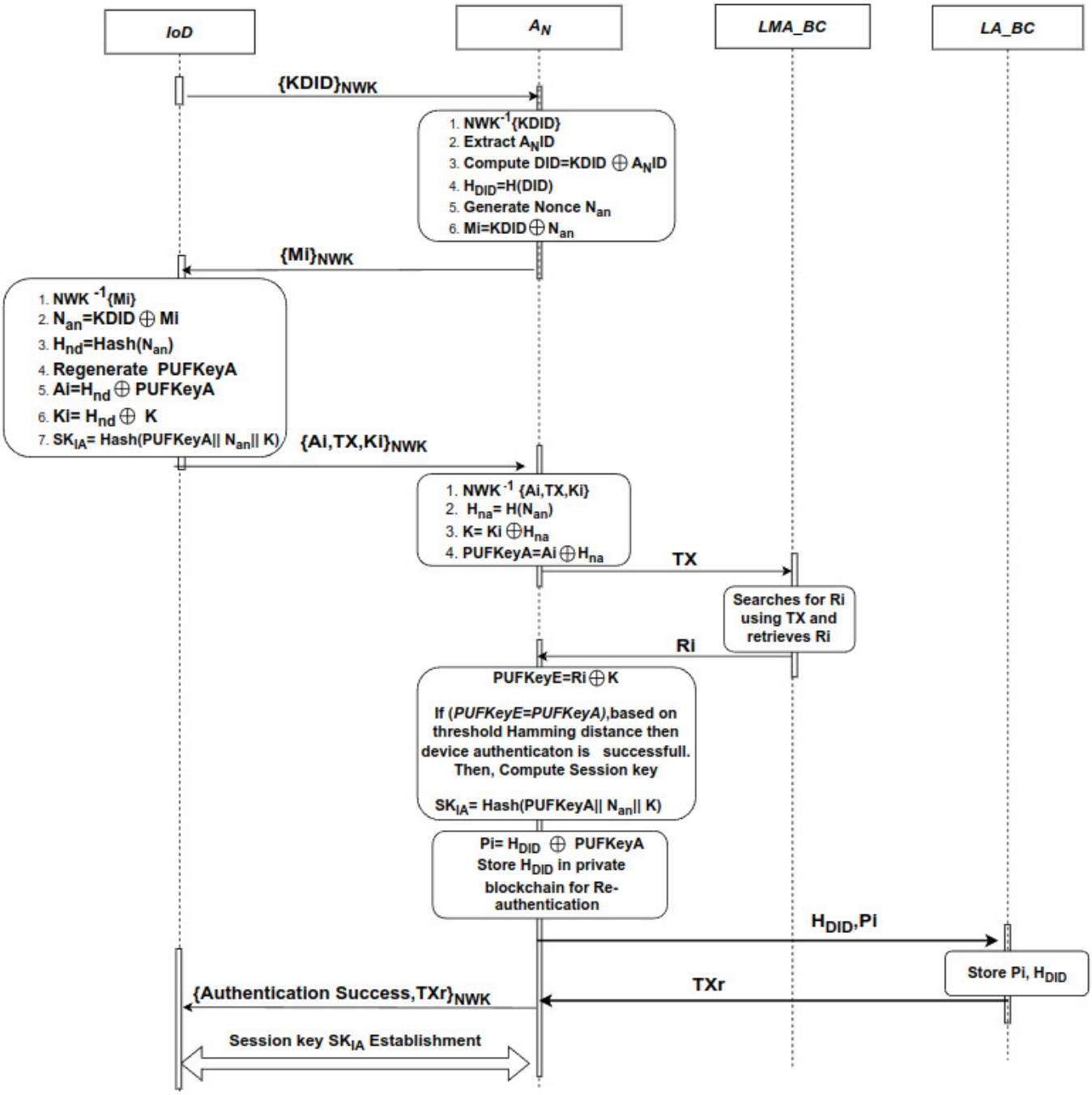


Fig. 5. Authentication phase.

$A_N$ end as it establishes mutual authentication between $IoD$ and $A_N$ and will be mainly used to communicate messages only after successful authentication.

$$SK_{IA} = Hash(PUFKeyA||N_{an}||K) \tag{9}$$

Then, the *IoD* encrypts the generated secret key Ai, Ki, and TX (which is stored in its memory) using NWK and sends it to the $A_N$ for consortium blockchain authentication.

6) The $A_N$ receives the tuple and decrypts it using NWK. The $nonce$ generated at $A_N$ end is then hashed using hash function *H* to get $H_{na}$ as shown in (10).

$$H_{na} = H(N_{an}) \tag{10}$$

The $A_N$ extracts *PUFKeyA* from *Ai* using (11).

$$PUFKeyA = H_{na} \oplus Ai \tag{11}$$

The $A_N$ extracts *K* from *Ki* using (12).

$$K = Ki \oplus H_{na} \tag{12}$$

Now, the $A_N$ has *PUFKeyA* , *K* and *TX* where the *TX* is unique transaction hash given by the consortium blockchain during the *IoD* registration phase. The $A_N$ then sends *TX* to the consortium blockchain to search or query for the corresponding *Ri* in its transaction history.

$$A_N \rightarrow LMA\_BC : TX \tag{13}$$

7) On successful search of *TX* specific *Ri* in previous transactions, the consortium blockchain sends the found *Ri* to the $A_N$ for verification. If *Ri* is not found, the consortium blockchain sends an unsuccessful search message to the $A_N$.

$$LMA\_BC \rightarrow A_N : Ri \tag{14}$$

8) The $A_N$ extracts *PUFkeyE* from *Ri* using secret key *K* as shown in (15).

$$PUFKeyE = Ri \oplus K \tag{15}$$

Now, the $A_N$ has two PUF keys namely *PUFKeyE* (PUF generated during registration) from the consortium blockchain and *PUFKeyA* from the *IoD* (re-genrated PUF during authentication). Both the PUF keys are then compared and based on threshold hamming distance [49]. If they match, the *IoD* will be stated as authenticated and allowed to interact with the network. Else, the *IoD* will be stated as unauthenticated and the *IoD* will be intimated accordingly. On successfull authentication, the $A_N$ computes session key $SK_{IA}$ similar to (9) followed by computation of $Pi$ using (16) and sends previously computed $H_{DID}$ and $Pi$ to the private blockchain for possible re-authentication of the same *IoD*.

$$Pi = H_{DID} \oplus PUFKeyA \tag{16}$$

$$A_N \rightarrow LA\_BC : \{H_{DID}, Pi\}$$

9) Private blockchain records $H_{DID}, Pi$ in its ledger. Private blockchain then sends corresponding transaction hash $TXr$ (related to private blockchain) of stored $H_{DID}$ to $A_N$ and $A_N$ sends $TXr$ along with authentication success acknowledgement to $IoD$. The mutual authentication and session key establishment will takes place between *Iod* and $A_N$ and $SK_{IA}$ will be used to further communicate messages or services requiered between *IoD* and $A_N$ untill device gets detached or session ends. The corresponding $TXr$ and $KDID$ will be used during re-authentication. For the subsequent IoD authentication, the private blockchain acts as an immediate authenticator bypassing the consortium blockchain.

The pseudo code for device authentication phase is shown in Algorithm 2.

**Algorithm 2:** Authentication phase

*IoD* encrypts the *KDID* using NWK and sends it to the $A_N$
$A_N$ decrypts *KDID*
$A_N$ computes $DID$=*KDID* $\oplus$ $A_{NID}$
$A_N$ computes $H_{DID}$=$H(DID)$
$A_N$ generates Nonce $N_{an}$
$A_N$ computes *Mi*=*KDID* $\oplus N_{an}$
$A_N$ encrypts $Mi$ using NWK and sends it to the *IoD*
The *IoD* computes $N_{an}$=*KDID*$\oplus$*Mi*
*IoD* computes $H_{nd}$=*Hash*($N_{an}$) // SHA-256
*IoD* reconstructs the PUF key i.e., *PUFKeyA* (Reconstruction phase)
*IoD* computes *Ai*=$H_{nd}\oplus$*PUFKeyA*
*IoD* computes *Ki*=$H_{nd}\oplus$*K*
*IoD* computes $SK_{IA} = Hash(PUFKeyA \parallel N_{an} \parallel K)$
*IoD* encrypts *Ai,Ki, TX* using NWK and sends to $A_N$
At authenticator end, the $A_N$ decrypts *Ai, TX* and computes $H_{na}$=*Hash*($N_{an}$)
$A_N$ Computes *PUFKeyA*= *Ai* $\oplus$ $H_{na}$
$A_N$ Computes *K*= *Ki* $\oplus$ $H_{na}$
$A_N$ sends $TX$ to consortium blockchain
  **if** *TX is found* **then**
    $A_N$ recieves Ri corresponding to TX from consortium blockchain
  **else**
    Device is not registered, "Authentication unsuccessful" error message will be sent back to device
  **end**
$A_N$ retrieves *PUFKeyE*=*Ri* $\oplus$ *K*
  **if** *PUFKeyE==PUFKeyA //based on hamming distance* **then**
    *IoD* is authenticated
    *IoD* computes $SK_{IA} = Hash(PUFKeyA \parallel N_A \parallel K)$
    *Pi*= $H_{DID}$ $\oplus$ *PUFKeyA*
    *Send* $H_{DID}$, *Pi to private blockchain*
  **else**
    *IoD* is rejected
  **end**
Send $TXr, ACK$ to *IoD*

### C. Re-authentication Phase

An IoD after the successful authentication by the consortium blockchain can interact with the network and it may log out or can get detached from the network in the due course. It may happen that the same IoD may again request authentication from the $A_N$ to interact with the network. In such cases, a re-authentication phase will be initiated where the private blockchain will authenticate the same *IoD* with less computation and less delay by skipping the consortium blockchain authentication procedure.

The sequence diagram of re-authentication phase is shown in Fig. 6 and is explained in the following steps.

1) The $A_N$ generates a session and *IoD*-specific NWK, upon re-authentication request, and shares it with the IoD for secure communication. The IoD then generates *Li* using (17) and encrypts the secret key *KDID, Li* and *TXr* stored in its memory using the *NWK*, and sends it to the $A_N$ for the re-authentication. Subsequently, the $IoD$ generates session key $SKR_{IA}$ using (18) which will be used to obtain mutual re-authentication and to secure communication after successful re-authentication.

$$Li = KDID \oplus PUFKeyRA \tag{17}$$

$$SKR_{IA} = PUFKeyRA||KDID \tag{18}$$

2) The $A_N$ decrypts the received message.

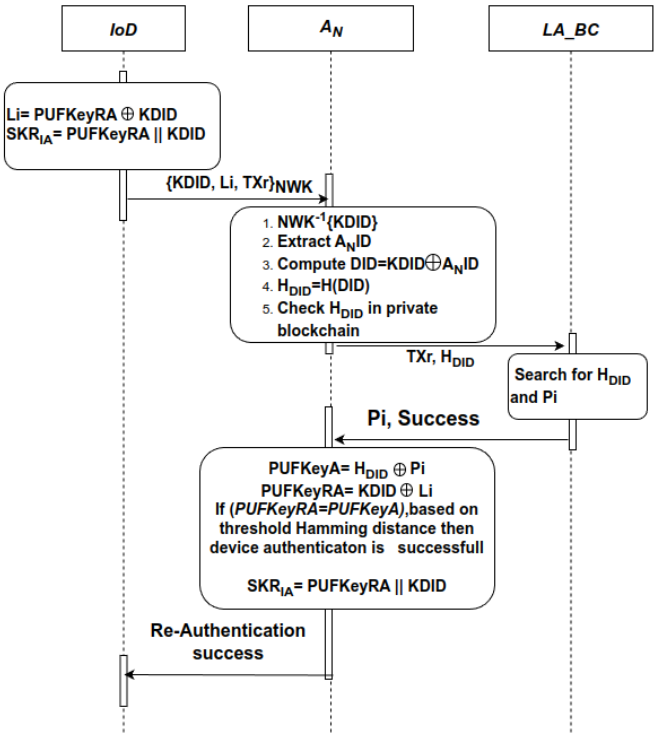


Fig. 6. Re-authentication phase.

3) The $A_{NID}$ will be used to generate *DID* which is further hashed to get $H_{DID}$ as shown in (19).

$$DID = KDID \oplus A_{NID}$$

$$H_{DID} = H(DID) \tag{19}$$

4) The $A_N$ then checks for $H_{DID}$ *and Pi* in the private blockchain using *TXr*. If the search is successful and if $H_{DID}$ matches with generated $H_{DID}$ by $A_N$, then private blockchain sends *Pi* to $A_N$.

$$A_N \rightarrow LA\_BC : TXr$$
$$LA\_BC \rightarrow A_N : Pi$$

5) $A_N$ retrieves *PUFKeyA* and *PUFKeyRA* using (20) and compares *PUFKeyA* and *PUFKeyRA* based on Hamming distance threshold.

$$PUFKeyA = H_{DID} \oplus Pi$$
$$PUFKeyRA = KDID \oplus Li \tag{20}$$

If the comparison satisfies the threshold, a success message will be sent by $A_N$ to the *IoD*. The IoD will be termed re-authenticated, and it is allowed to interact with the network. The session key after successful mutual re-authentication $SKR_{IA}$ will be generated at the $A_N$ end similar to (18) and will be used for future secure communication.

The pseudo code for the device re-authentication phase is shown in Algorithm 3.

**Algorithm 3:** Re-authentication phase

*IoD* computes *Li= KDID⊕ PUFKeyRA*
*IoD* computes $SKR_{IA}$= $PUFKeyRA \parallel KDID$
*IoD* encrypts the *KDID,Li,TXr* using NWK and send it to the $A_N$
$A_N$ decrypts *KDID*
$A_N$ computes *DID=KDID* $\oplus$ $A_{NID}$
$A_N$ computes $H_{DID}$ *=H(DID)*
$A_N$ checks for $H_{DID}$ *and* $Pi$ in private blockchain using $TXr$
  **if** $H_{DID}$ *match is found* **then**
    *IoD* sends *Pi* to $A_N$
  **end**
$A_N$ computes *PUFKeyRA= KDID⊕ Li*
$A_N$ computes *PUFKeyA=*$H_{DID}$ $\oplus$ *Pi*
  **if** *PUFKeyRA=PUFKeyA based on Hamming distance* **then**
    Send *ACK* as "success" to *IoD*
  **else**
    *IoD* not re-authenticated
  **end**
$A_N$ computes $SKR_{IA}$= PUFKeyRA $\parallel$ KDID

### D. Security Proof

A protocol is said to be secure if an adversary cannot interact with it and determine the confidential data such as secret key *K* or other sensitive information in the proposed protocol with total certainty. Thus, it is essential to analyse the proposed protocol in order to identify the protocol state that would endanger the device's and the network's confidentiality. In the following theorems, it is demonstrated that the proposed authentication protocol follows preimage resistance property and Kerckhoff principle and is secure against brute force attacks.

**Definition 1:** *(Brute force attck) Given an N-bit output of a cipher text y is z and if K =* $k_1, \cdots, k_n$ *is the key space of all possible keys* $k_i$*, it is computationally infeasible to calculate the input message x such that* $d_{ki}(z)$ *= x through brute-force attack for every* $k_i \in K$*.*

**Theorem 1:** ***The ciphertext Ri is secured from a brute-force adversary such that it is computationally infeasible to get secret key K or PUF key to decrypt ciphertext Ri, for all possible keys*** $ki \in K$**.**

**Proof:** Let (x,y) denote the pair of plain-text (PUF key) and cipher-text (Ri), and let K = $SK_1, \cdots, SK_n$ be the key space of all possible keys $k_i$. For every $k_i \in K$ a brute-force attack checks if $d_{ki}$(y) = *x*. If condition satisfies, a possible correct key $k_i$ is found. Consider a set of devices $D_1, D_2, \cdots, D_n$ which has PUF key $PUFKeyE_1$, $PUFKeyE_2, \cdots, PUFKeyE_n$. Consider secret key *K* for each device to be $SK_1, \cdots, SK_n$. Both secret keys and PUF keys are considered to be secured and out of the adversary's sight.

Equation (21) shows the extraction of $Ri$, the ciphertext of the device, from $PUFKeyE_n$ and $SK_n$ where *n = 1,2,*$\cdots$*,n* is a random device which has requested for the authentication.

$$Ri_n = PUFKeyE_n \oplus SK_n \tag{21}$$

Assume that the adversary *A* has cipher-text *Ri*. Now, the adversary has two options, get PUFKeyE and extract secret key *K* from *Ri* or guess secret key K and extract PUFKeyE from *Ri*.

***Case 1: Get PUFKeyE and extract secret key K from Ri***

The adversary getting hold of a PUFKeyE is a challenging task as it is generated on the fly and is not stored in the device explicitly. It is considered to be non-tampered, unpredictable and unclonable. Hence, PUFKeyE is assumed to be safe thereby stopping the adversary from getting the secret key *K*.

***Case 2: Guess secret key K and extract PUFKeyE from Ri.***

The possible combinations (PC) or population of secret key *K* are:

$$PC = Possible\ number\ of\ characters\ ^{Secretkeylength}. \tag{22}$$

Consider the length of SK as 20 digits and the possible number of characters in SK be 94 which is inclusive of numbers (0–9) 10, letters (A–Z and a–z) 52, and Special characters 32. Now the (22) becomes,

$$PC = 94^{20} = 2.9010624e + 39. \tag{23}$$

If a supercomputer can generate three trillion keys per second, then it would take,

2.90106241e+39 ÷ 3,000,000,000,000
= 9.67020804e+26 seconds.

An adversary may take 9.67020804e+26 seconds, which is approximately $3.064 \times 10^7$ trillion years, to predict *K* and decrypt *Ri*.

Hence the brute force attack for decrypting $d_{ki}$(Ri) will be computationally high or may take years to predict if Definition 1 holds good.

**Definition 2:** *(Preimage resistance) Given a secure hash function H with an N-bit output z, the computation required to determine the input message x such that H(x) = z is computationally infeasible.* [50].

**Definition 3:** *(Kerckhoffs principle) A cryptosystem must be secure even if the attacker is aware of all of its details other than the secret key. Particularly, even if the attacker is aware of the encryption and decryption procedures, the system should be secure.* [50].

**Theorem 2:** ***The proposed authentication protocol $\alpha$ is protected from eavesdropping adversaries if the Kerckhoffs principle and preimage resistance definitions hold good.***

**Proof:** Consider a nonce $N_{an} \in N$, where $N$ is set of all possible nonce, is encrypted using secret key *KDID* to get cipher-text *Mi* using (24).

$$Mi = KDID \oplus N_{an} \tag{24}$$

Assume that the *KDID* key length is known to the adversary. Consider the adversary constructs a more efficient algorithm $\alpha^{'}$ which can compute $KDID'$ using hash(KDID) based on key length using which an adversary can get hold of encrypted nonce as shown in (25).

$$N_{an} = Mi \oplus KDID = KDID' \oplus N_{an} \oplus KDID \tag{25}$$

Similar to (25), using the nonce, an adversary can get hold of PUF key or secret key as shown in (26).

$$\begin{aligned} H_a(N_{an}) \oplus Ai &= H_a(N_{an}) \oplus PUFkeyA \oplus H(N_{an}) \\ &= PUFkeyA \end{aligned} \tag{26}$$

Where $H_a$ is the hash function of adversary and Ai is response from the device end.

With (26), all upcoming authentications can be impersonated. The same nonce can be obtained using the secret key *KDID*, even if the protocol produces and selects a different nonce for subsequent authentication. Hence, revealing *K, KDID or PUFKey* puts the protocol at serious risk because any subsequent IoD authentication can be passed. To carry out this attack, the algorithm $\alpha^{'}$ must locate *KDID* from its hashed value, which infers a direct contradiction to both Kerckhoffs principle (Definition 3) and the preimage resistance property of a secure hash (Definition 2). Hence, the protocol $\alpha$ is termed to be secured if Kerckhoffs principle and the preimage resistance property hold good.

TABLE VI
TERMS USED IN ROR MODEL.

| Variables List | |
|---|---|
| Variable | Meaning |
| $q_H$ | Total number of active *Hash* queries |
| $q_P$ | Total number of active *PUF* queries |
| $q_s$ | Total number of active *Send* oracle queries |
| $q_e$ | Total number of *Execute* oracle queries in execution |
| $l_r$ | String length of random number |
| $C^{'}.s^{'}$ | Zipf parameters [51] |
| $L_H$ | Lists which stores hash oracle queries output |
| $L_A$ | Lists which stores random oracle output |
| $L_T$ | Lists that records message transcripts between $IoD^i$ and ${A_n}^j$ |

### *E. Formal Security Analysis using Real-or-Random (ROR) Model*

This section uses the ROR model [25], [44] to formally show the security of the proposed protocol. The terms used in ROR model are shown in Table VI. We begin by describing the participants $\tau$ in our protocol *P*, which consists of one IoT device *IoD* and authenticator node $A_n$. Each participant can have several instances that act as oracles. Three different types of results (accept, reject, terminate) can be obtained from an Oracle depending on the supplied data. The Oracle's current state will be accepted if the proper input is received, and the state will be rejected if the input is not appropriate. The oracle outputs the terminated state if there is no reply. As stated in the widely recognized Dolev–Yao (DY) threat model [52], we assume in this article that the adversary possesses the capabilities to completely seize control of the communication channel, meaning that they can replay, intercept, eavesdrop, and alter messages sent over the open channel. Next, we add Canetti and Krawczyk's model (CK-adversary model) [53] to

our model, in which the adversary can access the session keys or session states [44] in addition to the secret credentials. In addition, IoDs placed in public spaces would be vulnerable to physical attack. Through power analysis attacks, the adversary can obtain the private keys of IoDs [54]. Based on the capabilities listed in the adversary model, adversary A launches various attacks to compromise the security of our proposed protocol. The following is a simulation of these attacks using Oracle queries to the instances:

1) Execute($IoD^i, {A_n}^j$): This query enables *A* to eavesdrop on messages between the $IoD^i$ and ${A_n}^j$ and launch a passive attack.
2) Send($\tau, m$): This query enables an *A* to send request message *m* to instance $\tau$ and $\tau$ replies back with actual result resulting in active attack.
3) Reveal($\tau^k$): This query enables *A* to simulate Session key $SK_{IA}$ generated between $\tau^k$ and $A_n^j$.
4) Corrupt($IoD^i$): This query enables *A* to simulate physical attack on *IoD* and get the private key of $IoD^i$.
5) Test($\tau^k$): This query deals with testing the Semantic security of the session key, where *A* requests *P* for $SK_{IA}$ and *P* replies probabilistically on the outcome of flipped coin *b*.

Some of the basic concepts of ROR model are as follows:

*Freshness*: If both of the following conditions hold true at the same time: a) the session is in an approved state; b) no *Reveal* query has been run on the instance $\tau^k$and its partner. In such cases, the session is referred to as fresh. *Sementic security*: The indistinguishability of $SK_{IA}$ from a random number by an Adversary *A* based on the *Test* query is known as semantic security of protocol *P* .

**Definition 4:** *(Semantic Security) The proposed protocol P is semantically secure if the advantage function $ADV_A^{HP}(A)$ is only negligibly larger than $C^{'}.q_s^{s'}$*

**Definition 5:** *Let $ADV_A^{HP}(A)$ denote the advantage of adversary A running in polynomial time in breaking the semantic security of proposed protocol P then, $ADV_A^{HP}(A) = |2Pr|b^{'} = b| - 1|$,where b' is the guessed bit.*

**Definition 6:** *(Collision resistant one way hash function) A hash function $H\{0,1\}^* \rightarrow \{0,1\}^m$ is collision-resistant if it is computationally infeasible to find two distinct inputs that produce the same hash output of m bits and also for given hash value it is computationally infeasible to find any input that produces the same hash value (one-way). The advantage of an adversary A to find the hash collision is $ADV_A^{HP}(t) \leq Pr[(x, x^{'}) \in_R A{:}x \neq x^{'}, H(x)=H(x^{'})]$.If $ADV_A^{HP}(t) \leq \epsilon$, then hash function is collision-resistant, where $\epsilon$ is small positive number, and t is the maximum execution time.*

**Definition 7:** *(Secure PUF function) A PUF function is secure, if the variation of a PUF function's response to two arbitarary challenges, C1 and C2, is at least d1 and the variation to the same challenge for any two PUFs is at least d2 where d1 and d2 are security parameters. If PUF1(.), PUF2(.) are two PUF functions for the input C1,C2 $\in \{0,1\}^k$, PUF is secure if the variation is Pr[HD((PUF1(C1),PUF2(C2))> d]= $1-\epsilon$, where HD is Hamming distance and d is error tolerance threshold.*

**Theroem 3: If $ADV_A^{HP}(A)$ denotes the advantage function of an adversary A running in polynomial time *t* in breaking the semantic security of the proposed scheme *P*, then advantage is estimated by,**

$$ADV_A^{HP}(A) \leq \frac{q_H^2}{|Hash|} + \frac{q_P^2}{|PUF|} + \frac{(q_s + q_e)^2}{2^{lr}} + 2C^{'}.q_s^{s'}.$$

**Poof:** We follow the similar proof as applied in [55]–[58]. We define a sequence of four games, namely $GM_i$ for *i = 0, 1, 2, 3*. In a game $GM_i$, adversary *A* tries to guess a correct bit *b* through the Test query. This event is defined by $S_i$ and its corresponding probability is denoted by Pr[$S_i$].

$Game0$ ($Gm_0$) The initial game $Gm_0$ is considered to be identical with the actual protocol executing under the ROR model. Hence, we have,

$$ADV_A^{HP}(A) = 2|Pr[S_0] - 1|. \tag{27}$$

$Game1$ ($Gm_1$): This game considers simulation of *Send, Test, Execute, Reveal* and *Corrupt* queries with respect to the proposed scheme and considers lists $L_H, L_A, L_T$ for storing results of various oracle queries related to hash and transcripts. To derive $SK_{IA}$ , adversary requires both temporal secrets $N_A$ and K, and also the PUF key (PUFKEYA and PUFKeyRA) which is computationally infeasible to predict as per Defination 7. Hence, winning the game $Gm_1$ by A is not increased by eavesdropping,send and reveal attacks. Due to indistinguishability of $Gm_0$ and $Gm_1$, we obtain

$$Pr[S_1] = Pr[S_0]. \tag{28}$$

$Game2$ ($Gm_2$): Three sorts of collision scenarios, including hash (H) queries, PUF queries and random numbers are considered in this game for all the communicated messages between IoD and $A_n$. In message *Mi, Ai,* and *Ki*, the random number *K*, Hash function ($H_{nd}$), and PUF key are used. Based on the birthday paradox, the collision probability of the hash function and PUF function is $\frac{q_H^2}{2|Hash|}$ and $\frac{q_P^2}{2|PUF|}$ as per Definitions 6 and 7. Moreover, the messages also consist of a random number in the form of nonce and Secret key *K* and the probability of the random number collision is at most $\frac{(q_s+q_e)^2}{2^{lr+1}}$. Thus the obtained probability difference between $Gm_1$ and $Gm_2$ is

$$|Pr[S_2] - Pr[S_1]| \leq \frac{q_H^2}{2|Hash|} + \frac{q_P^2}{2|PUF|} + \frac{(q_s + q_e)^2}{2^{lr+1}}. \tag{29}$$

$Game3$ ($Gm_3$): In the final game, corrupt query simulation is considered where the adversary gets the credentials using the physical attack on the device. Using this attack, the adversary tries to guess *KDID, K,* and *TX* of the device. But to authenticate itself, the adversary should know the PUF key to calculate *Ai* during the authentication phase and *Li* during the re-authentication phase. The PUF function is considered to be secure in Definition 2, hence it is computationally difficult for adversaries to guess the PUF key through corrupt and send query.

Hence by using Zipf's law on passwords or secret keys simulation, it follows that

$$|Pr[S_3] - Pr[S_2]| \leq C^{'}.q_s^{s'}, \tag{30}$$

where $C^{'}$ and $s^{'}$ are Zipf's parameters [51].

Since all the games are executed, it is remained for adversary to guess the correct bit c. Thus, we have,

$$Pr[S_3] = \frac{1}{2}. \tag{31}$$

From (27) and (28),

$$\begin{aligned} \frac{1}{2}ADV_A^{HP}(A) &= |Pr[S_0] - \frac{1}{2}| \\ &= |Pr[S_1] - \frac{1}{2}|. \end{aligned} \tag{32}$$

From (31) and (32)

$$\frac{1}{2}ADV_A^{HP}(A) = |Pr[S_1] - Pr[S_3]|. \tag{33}$$

Using triangular inequality for (29) and (30), we obtain

$$\begin{aligned} &|Pr[S_1] - Pr[S_3]| \\ &\quad \leq |Pr[S_1] - Pr[S_2]| + |Pr[S_2] - Pr[S_3]| \\ &\quad \leq \frac{q_H^2}{2|Hash|} + \frac{q_P^2}{2|PUF|} + \frac{(q_s + q_e)^2}{2^{lr+1}} + C^{'}.q_s^{s'}. \end{aligned} \tag{34}$$

From (33) and (34)

$$\frac{1}{2}ADV_A^{HP}(A) \leq \frac{q_H^2}{2|Hash|} + \frac{q_P^2}{2|PUF|} + \frac{(q_s + q_e)^2}{2^{lr+1}} + C^{'}.q_s^{s'}. \tag{35}$$

Finally by multiplying both sides of (35) by factor of 2, we obtain

$$ADV_A^{HP}(A) \leq \frac{q_H^2}{|Hash|} + \frac{q_P^2}{|PUF|} + \frac{(q_s + q_e)^2}{2^{lr}} + 2(C^{'}.q_s^{s'}). \tag{36}$$

Hence, the theorem 3 is proved.

### F. Formal Security Verification using Scyther

Scyther tool is widely used in the verification of formal security analysis of communication protocols. This tool analyses all possible attacks on the protocol and gives descriptions in the form of flowcharts. The Scyther tool is scripted in security protocol description language (SPDL) by the roles of individual entities, that comprise computations, communications, and claims. The written script is executed in Scyther tool and is verified and termed as secure from all forms of attacks as shown in Fig. 7. The claims "Nisynch," "Niagree," "Alive," and "Weakagree" validated the authenticity of the entities while the confidentiality of credentials was verified by claim "Secret". ( *IoD*: Dev, Anode: $A_N$, *HyB*: consortium blockchain, PrivB: private blockchain). We have used Scyther-W32-v1.1.3, Python 2.7 and wxPython 2.8 in the computing system that has windows OS (64-bit).

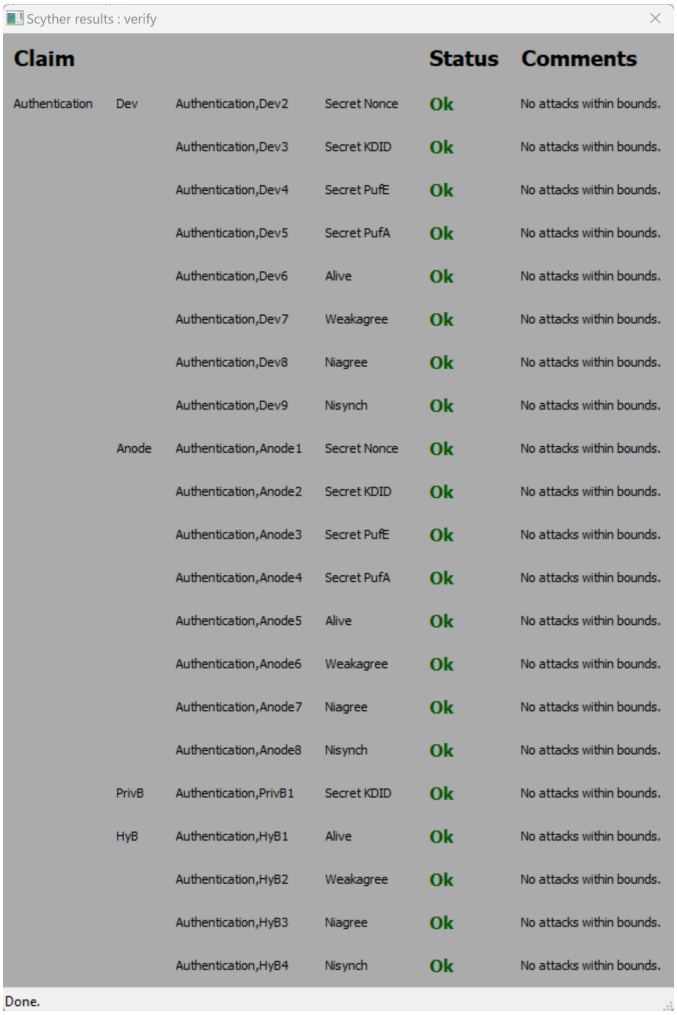


Fig. 7. Scyther tool results of proposed protocol.

## IV. Experimental Results and Discussion

The proposed system consists of an $A_N$, IoD and blockchain. The $A_N$ plays a major role in authenticating the *IoD* and acts as a mediator between network, blockchain, and the *IoD*. It has access to both consortium and private blockchains. The proposed system is implemented on a physical test bed as shown in Fig. 8. Raspberry Pi-4 is used as an $A_N$ with private blockchain and Arduino mega is used as an IoD. The computing system (laptop) is used as a consortium blockchain. The Raspberry PI-4 model B (4 GB) has a 64-bit quad core cortex-A72 (ARM V8) processor having 1.5 GHz clock speed and 4 GB SDRAM with the Raspbian operating system. The Arduino mega has ATMega 2560 microcontroller with 16 MHz clock speed and 8 KB SRAM and 256 KB flash memory. The computing system consists of a 2.90 GHz clock speed with 16 GB RAM, 1 TB of secondary storage and an Ubuntu 20.04 operating system.

We have used hyper-ledger fabric running in the laptop as consortium blockchain (*LMA_BC*) and private network in Raspberry PI as peer in private blockchain (*LA_BC*). The data in the consortium blokchain, $R_i$, can be viewed and verified by $A_n$ using a query, but cannot be modified. On the other hand, the private blockchain is only accessible to $A_N$ to store

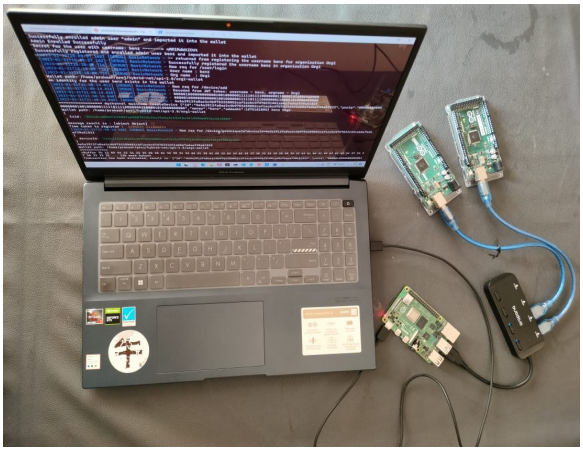

Fig. 8. The testbed.

TABLE VII
EXECUTION TIME OF PROPOSED PROTOCOL TEST-BED RESULTS.

| Primitive | Time in milliseconds (ms) |
|---|---|
| $E_{TR}$ | 4685.63 ms |
| $E_{TA}$ | 854.66 ms |
| $E_{TRA}$ | 393.52 ms |

$E_{TR}$: Time taken to register a device, $E_{TA}$: Time taken to authenticate a device, $E_{TRA}$: Time taken to re-authenticate a device.

$H_{DID}$,*Pi* upon successful authentication and retrieve the same during re-authentication phase. The blockchain is built on Hyper-ledger fabric V2.1 with NPM V12.22 and Go V1.13 is used for chain code. For the SRAM-PUF based device ID generation, the test-bed uses Arduino IDE v1.8.13 with C++ and Python 3 for extracting device ID. The physical testbed results are shown in Table VII which shows the time taken to authenticate an *IoD* using both the consortium blockchain and the private blockchain.

The results show that the time taken for the device authentication by the consortium blockchain ($E_{TA}$) is considerably more when compared to the time taken to re-authenticate a device by the private blockchain ($E_{TRA}$). Thus, the proposed re-authentication protocol is more feasible and economical for resource-constrained IoDs which undergo frequent authentication.

### A. Computation and Communication Cost Analysis

The computation cost of the proposed protocol against other state-of-the-art protocols namely [25], [36], [42]–[44], are shown in Table VIII. The computation cost is computed using cryptographic primitives, namely the total number of hash functions, PUF functions, bitwise XORs, random number generations, and scalar multiplications used in the protocols. The state-of-the-art protocols also consist of registration and authentication phases with a series of key exchanges between the user device and an authenticator node using cryptographic primitives. The total computation cost is the total cryptographic primitives used in protocols.

The computation cost of the proposed protocol and the state-of-the-art protocols for the registration phase and authentication phase is shown in Figs. 9 and 10. When compared to state-of-the-art protocols, our proposed protocol is computationally economical as it uses a minimum number of cryptography primitives (hash, PUF, XOR, and random number).

TABLE VIII
COMPARISON OF COMPUTATION COST OF PROPOSED PROTOCOL VS. STATE-OF-THE-ART APPROACHES.

| Articles | Computation cost |
|---|---|
| **P1** [25] | $H_{22t} + PUF_{5t} + XOR_{16t} + R_{3t} + B_{1t}$ |
| **P2** [36] | $H_{13t} + PUF_{2t} + XOR_{10t} + R_{3t}$ |
| **P3** [42] | $H_{7t} + PUF_{2t} + XOR_{20t} + R_{7t} + B_{1t}$ |
| **P4** [43] | $H_{14t} + PUF_{2t} + XOR_{16t} + R_{4t} + B_{1t}$ |
| **P5** [44] | $H_{25t} + PUF_{3t} + XOR_{6t} + R_{4t} + SM_{8t}$ |
| **Proposed** | $H_{3t} + PUF_{2t} + XOR_{10t} + R_{4t}$ |

H: Hash funtion, PUF: PUF function, XOR: Bit-wise XOR, R: Random number, B: Bio-metric, SM : Scalar multiplication, $N_t$: Number of times.

TABLE IX
COMPARISON OF COMMUNICATION COST OF PROPOSED PROTOCOL VS. STATE-OF-THE-ART APPROACHES.

| A | RP | AP | TC in bits | TC in bytes |
|---|---|---|---|---|
| [25] | 672 bits | 6720 bits | 7392 | 924 |
| [36] | 672 bits | 3648 bits | 4320 | 540 |
| [42] | 512 bits | 2912 bits | 3424 | 428 |
| [43] | 1696 bits | 3040 bits | 4736 | 592 |
| [44] | 1696 bits | 6112 bits | 7808 | 976 |
| **PR** | 576 bits | 1344 bits | 1920 | 240 |

A: Articles, RP: Registration phase, AP: Authentication phase, TC: Total cost, PR: Proposed.

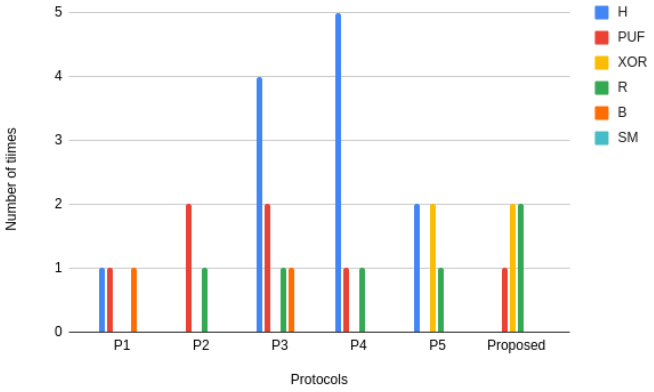


Fig. 9. Computation cost of proposed protocol vs. state-of-the-art protocols at registration phase.

For communication cost analysis, we have considered the length of the random number as 160 bits, the hash function (SHA – 256) as 256 bits, and PUF key as 256 bits. When compared to the hash function, the computation cost for bit-wise XOR operations, scalar multiplication and bio-metrics are infinitesimal. Hence, we have only considered hash, random number, and PUF function to calculate the communication cost. As shown in Table IX, the overall communication cost of our proposed protocol is 1920 bits, which is lightweight when compared to state-of-the-art schemes. The communication cost of our proposed protocol vs. state-of-the-art protocols at both registration and authentication phase is shown in Fig. 11.

The comparative analysis of various protocol features between the proposed protocol and other state-of-the-art protocols is shown in Table X. The proposed protocol fulfills all the

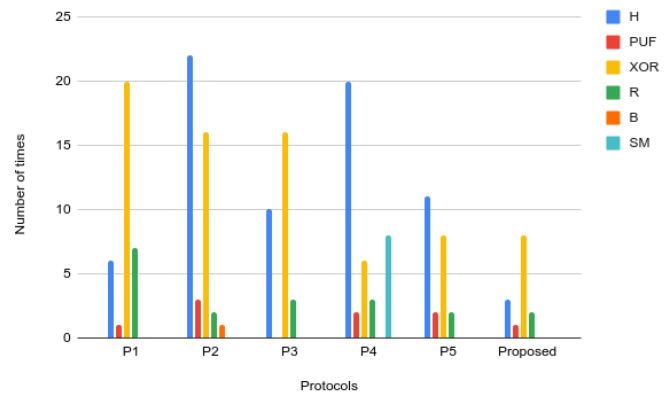


Fig. 10. Computation cost of proposed protocol vs. state-of-the-art protocols at authentication phase.

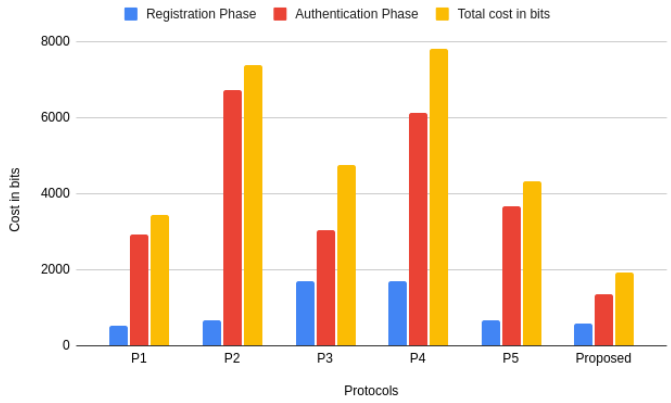


Fig. 11. Communication cost of proposed protocol vs. state-of-the-art protocols at registration phase and authentication phase.

TABLE X
COMPARISON WITH OTHER AUTHENTICATION PROTOCOLS.

| PF | C1 | C2 | C3 | C4 | C5 | C6 | C7 |
|---|---|---|---|---|---|---|---|
| [33] | * | ✓ | ✓ | ✓ | ✗ | ✓ | ✗ |
| [34] | ✓ | ✓ | ✗ | ✓ | ✗ | ✓ | * |
| [35] | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | * |
| [36] | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✗ |
| [37] | ✗ | ✗ | ✗ | ✓ | ✗ | ✓ | ✗ |
| [38] | * | * | * | ✓ | ✗ | ✗ | * |
| [39] | ✓ | ✓ | ✓ | ✗ | ✗ | ✓ | ✗ |
| [40] | * | * | * | ✓ | ✓ | ✗ | * |
| [41] | * | * | * | ✓ | ✗ | ✗ | * |
| [42] | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ | ✓ |
| [43] | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ |
| [44] | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ |
| [45] | ✗ | ✗ | ✓ | ✓ | ✓ | ✗ | * |
| [46] | ✓ | ✓ | ✗ | ✓ | ✓ | ✗ | ✓ |
| [47] | ✗ | ✓ | ✓ | ✓ | ✗ | ✓ | ✗ |
| [48] | ✗ | ✗ | ✓ | ✓ | ✓ | ✗ | * |
| **Proposed** | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

PF: Protocol Features, *: not applicable, C1: Dos attack protection, C2: Replay attack Protection, C3: MITM attack Protection, C4: Confidentiality, C5 : Formal security analysis, C6: Mutual authentication, C7: Low computation and communication overhead

protocol features and is prone to be secure, cost-effective, and efficient when compared to other state-of-the-art protocols.

### B. *Informal Security Analysis*

In this section, the various possible attacks or threats [59]–[61], to our proposed protocol have been discussed.

1) **Denial of service (DoS) attack**
A DoS attack occurs when an adversary purposefully causes a genuine IoD's authentication to fail in order to reject its service request. In order to stage failed authentication, the attacker should be able to manipulate credentials and send fake requests or ciphers to the $A_N$. As the communication is secured with network key *NWK*, this attack is eradicated so that an attacker cannot eavesdrop on the communication and only a legitimate *IoD* with a priorly shared network key can communicate with the $A_N$. This feature is tested in Scyther to be true.

2) **Man in the middle attack**
An attacker intervenes the communication between sender and receiver either to eavesdrop or impersonate either sender or receiver without their knowledge is known as man in the middle attack. The proposed protocol withstands man in the middle attack as each communication between the device and the authenticator is secured via network key and only legitimate device can interact with authenticator. Hence, only after verifying the device identity, communication takes place between the authenticator and device to eradicate communication attacks including man in the middle attack.

3) **Replay attack**
An attacker uses prior communications to change or impersonate credentials in an effort to authenticate an IoD. This is known as a replay attack. Consider an attacker receiving *Ai* from the device end. The original *Ai* is shown in (37).

$$Ai = H_{nd} \oplus PUFKeyA \tag{37}$$

Consider the attacker computes $Ai'$ as per (38) using advserary efficient algorithm.

$$Ai' = H'_{nd} \oplus PUFKeyA' \tag{38}$$

where $H'_{nd}$ and $PUFKeyA'$ are adversary altered credentials. Now the $A_N$ will have two ciphers namely $Ai$ and $Ai'$. Authenticator if receives $Ai'$ then there are two possibilities. Either $A_N$ will not be able decrypt $Ai'$ due to mismatch in $H_{nd}$ and reject the device or authenticator will retrieve adversary based $PUFKeyA'$ and it will certainly mismatch with $PUFKeyE$ and device will be rejected. In both the cases, device will be rejected. Hence to eradicate this , we have proposed network key for each transmission of *Ai* and *KDID,Ki* from device end to $A_N$ and vice versa, to send or receive legitimate ciphers at $A_N$ end and device end.

4) **IoD impersonation attack**
An adversary must have precise knowledge of $IoD's$ PUF key if it wants to pose as an $IoD$. Section III-E, however, proves that the advantage of an adversary to guess the PUF key is negligible. Thus, our proposed system can stop *IoD* impersonation attacks since the adversary cannot generate authentication request messages without precise PUF key.

5) **Physical attacks**

The entire authentication protocol relies on the security of device credentials. The adversary has to take critical, sophisticated measures (invasive and non-invasive physical attacks) to break memory to get hold of the elements stored in it. During this course, these sophisticated measures will affect the stability of SRAM memory causing a change in the PUF key. Hence, the resulting PUFKeyA will not match the PUFKeyE during authentication. Even if the adversary manages to simulate and guess credentials using a corrupt query (Section III-E) or physical attack, the adversary won't be able to authenticate itself because of the unclonable PUF key. Any attempt by an adversary to tamper with the device physically and to hack credentials will change the PUF key, as PUF relies on physical variations. An adversary without PUF key, wont be able to pass the authentication and re-authentication phases. Hence, the proposed approach is physical attack-resistant. Further, to enhance device memory security, tamper-proof memory, hardened firmware, tamper-evident seals, and secure packaging can also be used as future alternatives.

## V. Conclusion

The security of IoT devices in the current digital era is a challenging task as the number of devices is increasing exponentially day by day. Protecting the authenticity of the device is a mandatory requirement in today's era. In this article, PUF and blockchain-based authentication protocol is proposed for IoT devices that use both private and consortium blockchains where PUF provides device security and blockchain provides the security of credentials. The proposed protocol creates a device-centric PUF key and a secret key specific to each device. These two parameters are verified during the authentication phase to prove the device's authenticity. Further, the re-authentication protocol is also proposed in this work to speed up the authentication of devices that undergo frequent authentication. The protocol uses a network key and session key to secure the overall communication and thus provides authenticity, integrity, and confidentiality to the device and the device's credentials. The proposed protocol is theoretically proved by the ROR model and is proven to be stable against possible attacks, and the same is also proved using the Scyther security analysis tool. Further, the proposed protocol is tested on a physical testbed for its correctness, along with a performance analysis of the computation and communication cost of the proposed protocol against state-of-the-art protocols. The results are encouraging.

## References


[1] G.-J. Schrijen and C. Garlati, "Physical unclonable functions to the rescue," *Proceedings of the Embedded World*, 2018.

[2] I. Makhdoom, M. Abolhasan, J. Lipman, R. P. Liu, and W. Ni, "Anatomy of threats to the Internet of things," *IEEE Commun. Surveys Tuts*, vol. 21, no. 2, pp. 1636–1675, 2018.

[3] V. Rao and K. Prema, "Lightweight authentication and data encryption scheme for IoT applications," in *Proc. IEEE DISCOVER*, 2020.

[4] C. Maurya, A. Gupta, and V. K. Chaurasiya, "Efficient mutual authentication scheme for IoT-based healthcare system," in *Proc. IEEE CICT*, 2022.

[5] B. D. Deebak *et al.,* "Tab-sapp: A trust-aware blockchain-based seamless authentication for massive IoT-enabled industrial applications," *IEEE Trans. Ind. Inf.*, vol. 19, no. 1, pp. 243–250, 2023.

[6] 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.

[7] U. Marjit and P. Kumar, "Towards a decentralized and distributed framework for open educational resources based on ipfs and blockchain," in *Proc. IEEE ICCSEA*, 2020.

[8] E. Balistri, F. Casellato, C. Giannelli, and C. Stefanelli, "Blockhealth: Blockchain-based secure and peer-to-peer health information sharing with data protection and right to be forgotten," *ICT Express*, vol. 7, no. 3, pp. 308–315, 2021.

[9] K. Qian *et al.,* "Hpcchain: A consortium blockchain system based on CPU-FPGA hybrid-PUF for industrial Internet of things," *IEEE Trans. Ind. Inf.*, vol. 19, no. 11, pp. 11205–11215, 2023.

[10] A. Tomar, N. Gupta, D. Rani, and S. Tripathi, "Blockchain-assisted authenticated key agreement scheme for IoT-based healthcare system," *Internet of Things*, p. 100849, 2023.

[11] V. Aleksieva, H. Valchanov, and A. Huliyan, "Smart contracts based on private and public blockchains for the purpose of insurance services," in *Proc. IEEE ICAI*, 2020.

[12] P. Abhishek, D. Narayan, H. Altaf, and P. Somashekar, "Performance evaluation of Ethereum and hyperledger fabric blockchain platforms," in *Proc. IEEE ICCCNT*, 2022.

[13] H.-T. Wu and C.-W. Tsai, "An intelligent agriculture network security system based on private blockchains," *J. Commun. Netw.*, vol. 21, no. 5, pp. 503–508, 2019.

[14] I. Fedorov, A. Pimenov, G. Panin, and S. Bezzateev, "Blockchain in 5g networks: Perfomance evaluation of private blockchain," in *Proc. IEEE WECONF*, 2021.

[15] M. J. a. Mahmod and U. Guin, "A robust, low-cost and secure authentication scheme for IoT applications," *Cryptography*, vol. 4, no. 1, p. 8, 2020.

[16] B. Ray, M. Howdhury, J. Abawajy, and M. Jesmin, "Secure object tracking protocol for networked rfid systems," in *Proc. IEEE/ACIS SNPD*, 2015.

[17] S. Intrinsic-ID, "PUF: the secure silicon fingerprint," *White Paper*, 2016.

[18] Y. Yilmaz, S. R. Gunn, and B. Halak, "Lightweight PUF-based authentication protocol for IoT devices," in *Proc. IEEE IVSW*, 2018.

[19] A. Braeken, "PUF based authentication protocol for IoT," *Symmetry*, vol. 10, no. 8, p. 352, 2018.

[20] V. P. Yanambaka, S. P. Mohanty, E. Kougianos, and D. Puthal, "Pmsec: Physical unclonable function-based robust and lightweight authentication in the Internet of medical things," *IEEE Trans. Consumer Electron.*, vol. 65, no. 3, pp. 388–397, 2019.

[21] F. Farha, H. Ning, K. Ali, L. Chen, and C. Nugent, "SRAM-PUF-based entities authentication scheme for resource-constrained IoT devices," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5904–5913, 2020.

[22] L. Li, Z. Huang, Y. Chen, M. Du, Q. Wang, and J. Liu, "A PUF-based scheme for identify trusted mobile phone accessories," in *Proc. IEEE NaNA*, 2020.

[23] M. Ebrahimabadi, M. Younis, and N. Karimi, "A PUF-based modeling-attack resilient authentication protocol for IoT devices," *IEEE Internet Things J.*, vol. 9, no. 5, pp. 3684–3703, 2021.

[24] T. A. Idriss, H. A. Idriss, and M. A. Bayoumi, "A lightweight PUF-based authentication protocol using secret pattern recognition for constrained IoT devices," *IEEE Access*, vol. 9, pp. 80546–80558, 2021.

[25] P. Gope, A. K. Das, N. Kumar, and Y. Cheng, "Lightweight and physically secure anonymous mutual authentication protocol for real-time data access in industrial wireless sensor networks," *IEEE Trans. Ind. Inf.*, vol. 15, no. 9, pp. 4957–4968, 2019.

[26] C. H. Lau, K.-H. Y. Alan, and F. Yan, "Blockchain-based authentication in IoT networks," in *Proc. IEEE DSC*, 2018.

[27] K. Rahim, H. Tahir, and N. Ikram, "Sensor based PUF IoT authentication model for a smart home with private blockchain," in *Proc. IEEE ICAEM*, 2018.

[28] D. Puthal, S. P. Mohanty, P. Nanda, E. Kougianos, and G. Das, "Proof-of-authentication for scalable blockchain in resource-constrained distributed systems," in *Proc. IEEE ICCE*, 2019.

[29] Z. Haddad, M. M. Fouda, M. Mahmoud, and M. Abdallah, "Blockchain-based authentication for 5G networks," in *Proc. IEEE ICIoT*, 2020.

[30] H. Honar Pajooh, M. Rashid, F. Alam, and S. Demidenko, "Hyperledger fabric blockchain for securing the edge Internet of things," *Sensors*, vol. 21, no. 2, p. 359, 2021.

[31] B. Bera, A. K. Das, and A. K. Sutrala, "Private blockchain-based access control mechanism for unauthorized UAV detection and mitigation in Internet of drones environment," *Comput. Commun.*, vol. 166, pp. 91–109, 2021.
[32] M. Indushree *et al.,* "Mobile-chain: Secure blockchain based decentralized authentication system for global roaming in mobility networks," *Comput. Commun.*, 2022.
[33] Guartime and I. ID, "Internet of things authentication: A blockchain solution using SRAM physical unclonable functions," 2017.
[34] U. Guin, P. Cui, and A. Skjellum, "Ensuring proof-of-authenticity of IoT edge devices using blockchain technology," in *Proc. IEEE iThings, IEEE GreenCom, IEEE CPSCom, and IEEE SmartData*, 2018.
[35] M. N. Islam and S. Kundu, "Enabling IC traceability via blockchain pegged to embedded PUF," *ACM Trans. Design Auto. Electron. Syst.* , vol. 24, no. 3, pp. 1–23, 2019.
[36] A. S. Patil *et al.,* "Efficient privacy-preserving authentication protocol using pufs with blockchain smart contracts," *Comput. Security*, vol. 97, p. 101958, 2020.
[37] A. S. Patil, R. Hamza, H. Yan, A. Hassan, and J. Li, "Blockchain-PUF-based secure authentication protocol for Internet of things," in *Proc. ICA3PP*, 2019.
[38] L. Negka *et al.,* "Employing blockchain and physical unclonable functions for counterfeit IoT devices detection," in *Proc. IEEE COINS*, 2019.
[39] O. N. Diedhiou and C. Diallo, "An IoT mutual authentication scheme based on PUF and blockchain," in *Proc. IEEE CSCI*, 2020.
[40] S. P. Mohanty, V. P. Yanambaka, E. Kougianos, and D. Puthal, "PUFchain: A hardware-assisted blockchain for sustainable simultaneous device and data security in the Internet of everything (IoE)," *IEEE Consumer Electron. Mag.*, vol. 9, no. 2, pp. 8–16, 2020.
[41] V. K. Bathalapalli *et al.,* "PUFchain 2.0: Hardware-assisted robust blockchain for sustainable simultaneous device and data security in smart healthcare," *SN Computer Science*, vol. 3, no. 5, pp. 1–19, 2022.
[42] M. Masud, G. S. Gaba, P. Kumar, and A. Gurtov, "A user-centric privacy-preserving authentication protocol for IoT-ami environments," *Comput. Commun.*, vol. 196, pp. 45–54, 2022.
[43] W. Wang *et al.,* "Blockchain and PUF-based lightweight authentication protocol for wireless medical sensor networks," *IEEE Internet Things J.*, vol. 9, no. 11, pp. 8883–8891, 2021.
[44] Y. Zhang, B. Li, B. Liu, Y. Hu, and H. Zheng, "A privacy-aware pufs-based multiserver authentication protocol in cloud-edge IoT systems using blockchain," *IEEE Internet Things J.*, vol. 8, no. 18, pp. 13958–13974, 2021.
[45] L. Aniello *et al.,* "Anti-bluff: towards counterfeit mitigation in IC supply chains using blockchain and PUF," *Int. J. Inf. Security*, vol. 20, no. 3, pp. 445–460, 2021.
[46] K. P. Satamraju and B. Malarkodi, "A decentralized framework for device authentication and data security in the next generation Internet of medical things," *Comput. Commun.*, vol. 180, pp. 146–160, 2021.
[47] D. Li *et al.,* "Blockchain-based authentication for IIoT devices with PUF," *J. Syst. Architecture*, vol. 130, p. 102638, 2022.
[48] M. D. Islam, H. Shen, and S. Badsha, "Integrating blockchain into supply chain safeguarded by PUF-enabled rfid," *Internet of Things*, vol. 18, p. 100505, 2022.
[49] M. Platonov, J. Hlavác, and R. Lórencz, "Using power-up SRAM state of atmel atmega1284p microcontrollers as physical unclonable function for key generation and chip identification," *Inf. Security J.: A Global Perspective*, vol. 22, no. 5-6, pp. 244–250, 2013.
[50] C. Paar and J. Pelzl, *Understanding cryptography: A textbook for students and practitioners*. Springer Science & Business Media, 2009.
[51] D. Wang, H. Cheng, P. Wang, X. Huang, and G. Jian, "Zipf's law in passwords," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 11, pp. 2776–2791, 2017.
[52] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. 29, no. 2, pp. 198–208, 1983.
[53] R. Canetti and H. Krawczyk, "Universally composable notions of key exchange and secure channels," in *Proc. EUROCRYPT*, 2002.
[54] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Trans. Comput.*, vol. 51, no. 5, pp. 541–552, 2002.
[55] K. Park, Y. Park, Y. Park, A. G. Reddy, and A. K. Das, "Provably secure and efficient authentication protocol for roaming service in global mobility networks," *IEEE Access*, vol. 5, pp. 25110–25125, 2017.
[56] F. Wang, G. Xu, and G. Xu, "A provably secure anonymous biometrics-based authentication scheme for wireless sensor networks using chaotic map," *IEEE Access*, vol. 7, pp. 101596–101608, 2019.
[57] J. Zhao *et al.,* "A secure biometrics and pufs-based authentication scheme with key agreement for multi-server environments," *IEEE Access*, vol. 8, pp. 45292–45303, 2020.
[58] S. Roy *et al.,* "On the design of provably secure lightweight remote user authentication scheme for mobile cloud computing services," *IEEE Access*, vol. 5, pp. 25808–25825, 2017.
[59] K. M., B. Ray, J. Hassan, A. Kashyap, and V. Chandrappa, "Blockchain based secure ownership transfer protocol for smart objects in the Internet of things," *Internet of Things*, p. 101002, 2023.
[60] W. Wang *et al.,* "Blockchain-based reliable and efficient certificateless signature for IIoT devices," *IEEE Trans. Ind. Inf.*, vol. 18, no. 10, pp. 7059–7067, 2022.
[61] W. Wang *et al.,* "Ultra super fast authentication protocol for electric vehicle charging using extended chaotic maps," *IEEE Trans. Ind. Applicat.*, vol. 58, no. 5, pp. 5616–5623, 2022.

**Tyson Baptist D Cunha** received the M.Tech. degree in Computer Science and Engineering from NMAMIT, Nitte University, India, in 2019. He is currently pursuing a Ph.D. degree in the Department of Information Technology, National Institute of Technology Karnataka, Surathkal, India. He has one year of teaching experience. His current research interest is IoT security using PUF and blockchain.

**Kiran Manjappa** (Member, IEEE) is an Assistant Professor in the Department of Information Technology, National Institute of Technology Karnataka (NITK), Surathkal, India. He holds a Ph.D. from the National Institute of Technology Karnataka (NITK), Surathkal, India. He has contributed to the literature through many international conferences and journal publications under IEEE, IET, and Springer publishers. He is serving as a Reviewer for reputed journals and served as Chair and Technical Program Committee member for international conferences. Dr. Kiran is the Technical Expert Committee Member for the Smart City projects "E-Library" and "Centralized Command Center" in Karnataka State, India. His research interests include blockchain technology, IoT, and 5G networks.