2023

Blockchain and PUF-Based Secure Key Establishment Protocol for Cross-Domain Digital Twins In Industrial Internet of Things Architecture

Khalid Mahmood
University of Central Lancashire

Salman Shamshad
University of Lahore

Muhammad Asad Saleem
University of Sahiwal

Rupak Kharel
University of Central Lancashire

Ashok Kumar Das
International Institute of Information Technology

See next page for additional authors

Original Publication Citation

Mahmood, K., Shamshad, S., Saleem, M. A., Kharel, R., Das, A. K., Shetty, S., & Rodrigues, J. J. P. C. (2023). Blockchain and PUF-based secure key establishment protocol for cross-domain digital twins in Industrial Internet of Things architecture. *Journal of Advanced Research, 62*,155-163 . https://doi.org/10.1016/j.jare.2023.09.017

Journal of Advanced Research 62 (2024) 155–163

Contents lists available at ScienceDirect

Journal of Advanced Research

journal homepage: www.elsevier.com/locate/jare

Blockchain and PUF-based secure key establishment protocol for cross-domain digital twins in industrial Internet of Things architecture

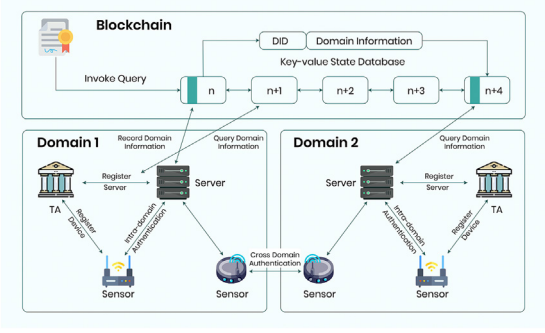
Khalid Mahmood [a,b,*], Salman Shamshad [c], Muhammad Asad Saleem [d], Rupak Kharel [a], Ashok Kumar Das [e,*], Sachin Shetty [f], Joel J.P.C. Rodrigues [g]

[a] School of Psychology and Computer Science, University of Central Lancashire, Preston, United Kingdom
[b] Graduate School of Intelligent Data Science, National Yunlin University of Science and Technology, Yunlin 64002, Taiwan, ROC
[c] Department of Software Engineering, The University of Lahore, Lahore 54590, Pakistan
[d] Department of Computer Science, University of Sahiwal, Sahiwal 57000, Punjab, Pakistan
[e] Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad 500 032, India
[f] Department of Modeling, Simulation and Visualization Engineering, Virginia Modeling, Analysis and Simulation Center, and Center for Cybersecurity Education and Research, Old Dominion University, Suffolk, VA 23435, USA
[g] COPELABS, Lusófona University, Campo Grande 376, 1749-024 Lisbon, Portugal

HIGHLIGHTS

- Design a blockchain-based secure key establishment protocol for cross-domain IoT architecture using Physically Unclonable Functions (PUFs).
- The developed protocol guarantees data transfer security across the domain and thwarts IoT devices from potential physical attacks.
- The proposed protocol employs a cross-domain trust-building method that helps the IoT devices derive keys from the multiple accumulator factors.
- We integrated cross-domain device authentication into the on-chain accumulator to resourcefully authenticate the unlinkable identities of IoT devices from distinct domains.
- We implemented the proof-of-concept prototype of the designed protocol.

GRAPHICAL ABSTRACT



ARTICLE INFO

Article history:
Received 14 May 2023
Revised 23 August 2023
Accepted 20 September 2023
Available online 29 September 2023

ABSTRACT

Introduction:: The Industrial Internet of Things (IIoT) is a technology that connects devices to collect data and conduct in-depth analysis to provide value-added services to industries. The integration of the physical and digital domains is crucial for unlocking the full potential of the IIoT, and digital twins can facilitate this integration by providing a virtual representation of real-world entities.

* Corresponding authors.
E-mail addresses: khalidm.research@gmail.com (K. Mahmood), salmanshamshad01@gmail.com, salman.shamshad@se.uol.edu.pk (S. Shamshad), masad@cuisahiwal.edu.pk (M.A. Saleem), rkharel1@uclan.ac.uk (R. Kharel), iitkgp.akdas@gmail.com, ashok.das@iiit.ac.in (A.K. Das), sshetty@odu.edu (S. Shetty), joeljr@ieee.org (J.J.P.C. Rodrigues).

https://doi.org/10.1016/j.jare.2023.09.017

Keywords:
Digital twins
Industrial Internet of Things (IIoT)
Mutual authentication
Key agreement
Physically Unclonable Functions (PUFs)

Objectives:: By combining digital twins with the IIoT, industries can simulate, predict, and control physical behaviors, enabling them to achieve broader value and support industry 4.0 and 5.0. Constituents of cooperative IIoT domains tend to interact and collaborate during their complicated operations.
Methods:: To secure such interaction and collaborations, we introduce a blockchain-based cross-domain authentication protocol for IIoT. The blockchain maintains only each domain's dynamic accumulator, which accumulates crucial materials derived from devices, decreasing the overhead. In addition, we use the on-chain accumulator to effectively validate the unlinkable identities of cross-domain IIoT devices.
Results:: The implementation of the concept reveals the fact that our protocol is efficient and reliable. This efficiency and reliability of our protocol is also substantiated through comparison with state-of-the-art literature. In contrast to related protocols, our protocol exhibits a minimum 22.67% increase in computation cost efficiency and a 16.35% rise in communication cost efficiency.
Conclusion:: The developed protocol guarantees data transfer security across the domain and thwarts IoT devices from potential physical attacks. Additionally, in order to protect privacy, anonymity and unlinkability are also guaranteed.

1. Introduction

Digital twins are a critical aspect of the Industrial Internet of Things (IIoT) and refer to virtual models that replicate the real-world behaviors of a physical entity or system. These virtual representations can be used to simulate and predict the physical behaviors of the corresponding real-world entity and can even be used to control the physical entity where applicable. The combination of digital twins and the IIoT can lead to the seamless integration of the physical and digital domains, which is necessary to derive maximum value from the IIoT. Digital twins enable engineers and operators to visualize, analyze, and optimize the performance of industrial systems, and even identify potential issues before they occur. The utilization of digital twins in combination with the IIoT can provide valuable insights into industrial operations, leading to more efficient and cost-effective industrial practices [1].

Recent developments have made the Industrial Internet of Things (IIoT) a viable device-oriented platform for industrial settings. Device-to-device linkages connect industrial assets, such as IIoT devices, resources, and systems, to facilitate the emergence of smart manufacturing. The operational procedures are linked with information technology to provide remote real-time access, flexible data collecting and exchange, and command on demand, among other things. In such a set-up, operating procedures and information technology enable remote real-time access, flexible data collection, and exchange. Consequently, several cutting-edge IIoT applications have emerged, including those for public security, factory management, healthcare services, and food supply chains. The cloud-based network enables IIoT devices from diverse cooperating areas to communicate and cooperate information flexibly to complete increasingly complex industrial tasks efficiently. Therefore, cross-domain cooperation in IIoT has the potential to considerably increase productivity and turn into a future paradigm of industrial production.

Integrating cross-domain cooperation in the IIoT is not a simple process due to significant security, privacy, and trust issues. IIoT devices installed in public places are susceptible to physical, cloning, and malicious impersonation threats. In addition, the public channel that carries the sensitive data is vulnerable to active or passive attacks, such as replay attacks or eavesdropping. Furthermore, building trust between various domains becomes a challenge since entities in each domain would only have confidence in the domain administrator. Therefore, it is clear that cross-domain device authentication plays a vital role in protecting cross-domain collaborations. It may be used to authenticate entities from other domains, foster confidence among them, and create a reliable session key to secure the public channel.

Over the past few years, the blockchain has appeared as an emerging technology that enables IoT devices to transfer data without controlling a central entity more securely and autonomously. Many researchers have presented state-of-the-art work focused on developing blockchain-based IoT ecosystems [2–5]. For instance, Dai et al. [2] in 2019 presented the concept of Blockchain of Things (BCoT). In their work, Dai et al. [2] emphasize the applications of BCoT in industrial processes. Additionally, they summarize the multiple challenges of traditional IoT architecture and their potential solution with blockchain. Ferrag et al. [3] also presented a comprehensive survey on the classification of threat models for BCoT and summed up the existing security and privacy literature for BCoT to demonstrate duture challenges. After that, Ferrag et al. [5] presented another remarkable review on the evaluation techniques for BCoT's security and consensus algorithms, which is a guide for other researchers to evaluate the blockchain-assisted privacy-preserving solutions for the IoT ecosystem. Kai et al. [4] presented their survey on the security challenges and their potential solutions using the smart contract in the BCoT. Motivated by these review papers, various researchers have contributed to improving the security of the IoT using the blockchain framework.

In 2020, Ali et al. [6] designed a blockchain-assisted authentication and authorization security solution for cross-domain IoT devices and users. The authors in [6] developed a hierarchy of global and local smart contracts to achieve access control and permission delegation that helps to preserve the privacy of the user and resist illegal delegation simultaneously. Guo et al. [7] inherited the hierarchical structure to develop a cross-domain authentication mechanism to enhance the data credibility and efficiency of the cross-domain IoT ecosystem. Moreover, Wang et al. [8] proposed an access control structure using the undirected graph to authenticate IIoT devices. They combined the digital accumulator and signature to accomplish the transitivity of signatures among distinct domains.

Tang et al. [9] introduced a passport-based security framework to achieve mutual authentication among cross-domain devices in an IoT architecture. They employed blockchain technology to record the signatures of devices involved in cross-domain communication. Collaborative devices use the signatures to build incentive mechanisms and achieve the authorization. Ma et al. [10] developed a blockchain-assisted security solution for privacy-oriented IoT systems based on a hierarchical access control mechanism. The authors in [10] employed multiple blockchain architectures to deal with the high-scalability and low-latency requirements of the IoT ecosystem.

In the recent literature, many researcher employ Identity-based Cryptography (IBC) techniques leveraged with blockchain technol-

ogy for privacy-oriented IoT scenarios. For instance, Jia et al. [11] developed a blockchain-assisted identity-based self-authentication mechanism to replace the conventional trusted certificate authority. Moreover, Shen et al., [12] introduced a blockchain-based access control structure to accomplish cross-domain device authentication in the IIoT architecture. Their scheme [12] exchanges the identity-based public keys among different IIoT domains with the help of a blockchain network. Chen et al. [13] also presented a decentralized identity management framework to evade the risk of single-point failure. Moreover, they utilized consensus algorithms to share the identity information across the domain without violating the privacy of the sender and receiver.

Ao et al. [14] introduced a blockchain-oriented dynamic key negotiation protocol for cross-domain autonomous vehicles. They utilized an optimized algorithm to maintain the transaction records, where the sender signs each transaction and shares it through the blockchain network. Similarly, Lin et al. [15] developed a privacy-preserving access control scheme using key a derivation method for blockchain-based Vehicular Adhoc Networks (VANETs). To enrich each transaction's efficiency and batch verification, the authors in [15] also used modified signature technique in the design of their protocol. Besides, Kang et al. [16] presented a blockchain-aided protocol leveraged with the reputation-based mechanism for secure and quality authorized data storage in edge-assisted VANET. Likewise, Cheng et al. [17] developed a blockchain-based key-management protocol for mobility devices in an edge computing environment. To overcome "incomplete cross-domain" problem of blockchain-based networks, Zhang et al. [18] designed a dynamic identity-based protocol that enables users from distinct domains to communicate using public identities. Quite recently, Zhang et al. [19] presented a multi-factor device access control and authorization protocol based on blockchain for cross-domain IIoT networks.

1.1. Motivation and contributions

From the literature, it is worth considering that not a single work has yet considered the on-chain storage overhead. Moreover, lack of unlikability and efficiency are other issues in these studies. Therefore, it is a dire need to reduce such cost from the protocol's efficiency aspects. Besides, the combining multifactor and blockchain technology in the design of access control mechanism can enrich cross-domain device collaborations and communications. The primary focus of this work is to address the secure transmission of cross-domain devices by designing a lightweight security protocol with the help of smart blockchain ledger. Moreover, we have considered the secure sharing of outsourced data for cross-domain IoT devices in the design of secure authentication mechanism. Following are the multifold contributions of our article:

- We design a blockchain-based secure key establishment protocol for cross-domain IoT architecture using Physically Unclonable Functions (PUFs).
- The developed protocol guarantees data transfer security across the domain and thwarts IoT devices from potential physical attacks.
- Our proposed protocol employs a cross-domain trust-building method that helps the IoT devices derive keys from the multiple accumulator factors. We uses the blockchain network to record the accumulator value of each domain, which lessens the on-chain storage overhead.
- In the design of our protocol, we integrate cross-domain device authentication into the on-chain accumulator to resourcefully authenticate the unlinkable identities of IoT devices from distinct domains. Consequently, privacy preservation and high efficiency are simultaneously satisfied in our protocol.
- To rigorously evaluate the performance of our protocol, we implemented the proof-of-concept prototype of the designed protocol.

We organized the rest of the paper as follows: Section 2 presents the preliminaries relevant to the theme of the paper. Section 3 discusses the proposed protocol. Security evaluation and performance analysis are observed in Sections 4 and 5. Section 6 presents the concluding remarks.

2. Preliminaries

This section demonstrates the system model and threat model. Moreover, the common notations that we use throughout the article is summarized in Table 1.

2.1. System model

As illustrated in Fig. 1, the system model comprises multiple domains and a blockchain. Each domain has a cluster of IoT devices, a server, and a Trusted Authority (TA). The role of the constituents of each domain (i.e IoT device, server & Trusted Authority (TA)) and blockchain is expressed below:

- **IoT device:** The IoT devices are densely deployed and work in a particular domain to interact with users or accomplish their designated tasks. In this paper, the IoT devices are supposed to have enough processing resources. Therefore, it can efficiently execute all the cryptographic operations used in the design of our protocol. IoT devices are employed to observe, process and exchange data with each to accomplish the designated tasks. They can also communicate with both intra-domain or cross-domain devices and servers.
- **Server:** The servers of each domain have considerable resources to offer numerous services, including data analysis, real-time access, and data collection from the cluster of IoT devices. The server of each domain has the privilege to query the ledger of the blockchain to get the information of any specified domain.
- **Trusted authority (TA):** The TA is supposed to be an honest and trustworthy entity of each domain. It is responsible for enrolling the domain server and IoT devices. Additionally, TA oversees the smart contracts whenever a new entity enrolls in the system.
- **Blockchain:** The server and TA of every domain join the blockchain network to revoke, query, and update the information through the smart contract. The blockchain is assumed to be a public ledger that manages the information of each domain. Thus, there are twofold fundamental concerns about the blockchain: i) it should support smart contracts, and ii) it should be robust to offer security. To address such imperative requisites, the integration of the proposed protocol with blockchain plat-

Table 1
Notations and their meanings.

Symbol	Explanation
$\mathcal{TA}$	Trusted Party Agent
$\mathcal{S}_s$	The designated Server
SK_{sa}, PK_{sa}	Secret and Public keys of $\mathcal{S}_s$
$\mathcal{D}_i$	ith IoT Device
k_i, c_i	Parent Secret Keys of $\mathcal{D}_i$
$sk_{i,j}$	Child Secret Key of $\mathcal{D}_i$
$< \mathcal{C}, \mathcal{R} >$	Challenge Response Pairs
PUF	Physically Unlonable Function
SK	Session key
SID	Session Identity
$\mathbb{A}$	Adversary

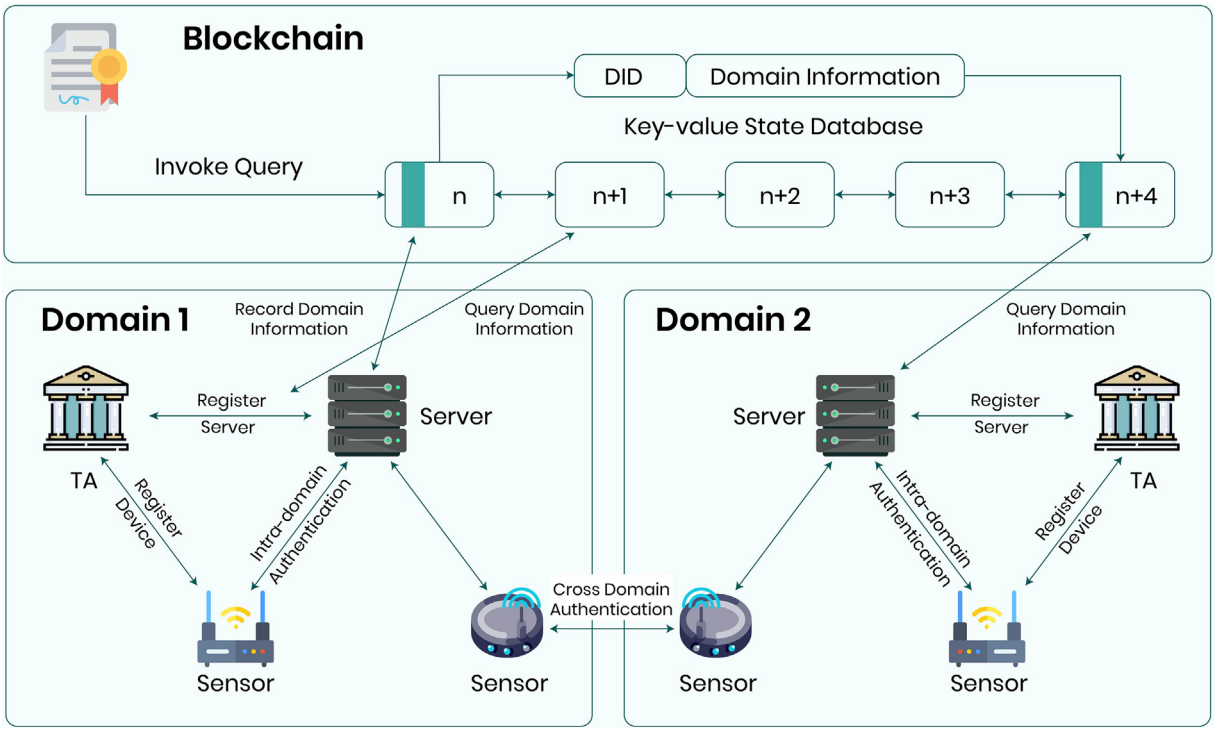


Fig. 1. Blockchain-enabled cross-domain IIoT architecture.

forms emerges as an effective solution. In this pursuit, we configure the server and TA of each domain as peer-nodes within the designated blockchain platform, which could encompass platforms such as Ethereum 2.0 or Hyperledger. This strategic alignment ensures that our protocol's aided features are seamlessly integrated into the blockchain's core framework.

The blockchain acts as a decentralized and dispersed ledger, where the information stored within the ledger is synchronized, tamper-proof and highly available among the peer in the form of smart contracts. Each peer node invokes a smart contract transforms the key-value into a digital ledger and records them into the blockchain repository. Here, the value represents the information of the designated domain, while the key is the blockchain address against the designated domain. The domain information *DID* includes public key of the domain, the version, the accumulator and the public parameters of IoT devices.

2.2. Threat model

This section introduces the capabilities of an adversary $\mathbb{A}$ under the well-known Canetti-Krawczyk (CK) and Dolev-Yao (DY) threat models. According to these models, $\mathbb{A}$ can interrupt, modify, and reply any public communication message. $\mathbb{A}$ can access the secret credentials stored within the IoT device through some power analysis approach. Moreover, these devices are a soft target for $\mathbb{A}$ to mount cloning or physical attacks. Each domain's TA and server are always trustworthy and semi-honest, respectively. The long-term secret values of the server are assumed inaccessible for $\mathcal{A}$. The transaction recorded in the blockchain is always available and tamper-proof. Therefore, if $\mathbb{A}$ tries to query the blockchain ledger, then even in that case, $\mathbb{A}$ cannot corrupt or purpose transaction to the blockchain.

3. Proposed protocol

This section demonstrates the discussion of the designed protocol in detail. Our protocol mainly comprises six phases: 1) system setup, 2) server registration, 3) device registration, 4) device revocation, 5) intra-domain authentication and 6) cross-domain authentication. Since the registration phase is completed in an offline manner, therefore, we assume all the messages are exchanged over private channel. Whereas the messages during the authentication and key-establishment phase are exchanged over the Internet, therefore they are assumed as a public channel exchange.

3.1. System setup

To initialize parameters of each domain, the trusted authority $\mathcal{TA}$ of each domain chooses a cyclic group $\mathcal{G}$ with the generator G having order r. At the same time, chooses one-way hash function $h(.)$ defined as: $h : \{1,0\}^* \rightarrow \{1,0\}^l$, where l symbolizes the output length of $h(.)$. Lastly, $\mathcal{TA}$ makes $PP_a = \{\mathcal{G}, G, h(.), DID\}$ public, where *DID* denotes the domain identity.

3.2. Server Registration

Initially, $\mathcal{TA}$ selects private and public key pair as SK_{sa}, PK_{sa} and shares with the server $\mathcal{S}_{sa}$. Thereafter, $\mathcal{TA}$ creates a smart contract in order to store the information of the domain $(PK_{sa}, PP_a, h(DID), version)$ into the ledger of the blockchain. If any new transaction is recorded, $\mathcal{TA}$ updates the values in the blockchain's ledger.

3.3. Device registration

Firstly, the device $\mathcal{D}_i$ picks unique and temporary identities ID_i and Tid_i. $\mathcal{D}_i$ also chooses i as its working mode. Afterwards, $\mathcal{D}_i$ shares both ID_i and Tid_i with $\mathcal{TA}$.

On receiving both identities, $\mathcal{TA}$ picks $< \mathcal{C}_i, \mathcal{R}_i >$ and computes: $X_i = h(ID_i \| SK_{sa}), (k_i, c_i) = DerPsk(ID_i, i, r, X_i), sk_{i,j} = DerCsk(k_i, c_i, \mathcal{R}_i)$ and $\mathcal{E}_i = sk_{i,k} + ID_i sk_{sa}$. Thereafter, $\mathcal{TA}$ maps the elements $K_i = k_i.G, C_i = c_i.G$ and generate a set of pseudonyms $PID_i = pid_{i1}, pid_{i2}, \ldots, pid_n$. $\mathcal{TA}$ then selects an accumulator ACC_i

and builds a mapping table $\mathbb{MT} = \{PID_i, C_i, K_i, ID_a, ACC_a\}$. At the end, $\mathcal{TA}$ locally records $\mathbb{MT}$ to the server's database and protects it with servers private key. At the same time, $\mathcal{TA}$ sends $\{PID_i, C_i, \mathcal{E}_i, ACC_a, PUF(.)\}$ towards $\mathcal{D}_i$.

On receiving the registration credentials from the $\mathcal{TA}, \mathcal{D}_i$ records the parameters in its memory for later use.

3.4. Device revocation phase

To revoke the status of any device, $\mathcal{TA}$ computes $\Delta_{new} = \Delta(ACC_i + \frac{1}{2})$ and deletes ACC_i from the accumulator. Thereafter, $\mathcal{TA}$ creates the ledger and updates the domain's information into the blockchain. At the same time $\mathcal{TA}$ distributes the updated values to $\mathcal{S}_{sa}$ and $\mathcal{D}_i$, respectively.

3.5. Intra-domain authentication

If a device $\mathcal{D}_{ia}$ wants to communicate with its domain's server $\mathcal{S}_{sa}$, then $\mathcal{D}_{ia}$ performs the trailing steps to establish mutual intra-domain authentication.

1. Firstly, $\mathcal{D}_{ia}$ inputs ID_{ia} and generates r_1.Thereafter, $\mathcal{D}_{ia}$ picks $< \mathcal{C}^*_{ia}, \mathcal{R}^*_{ia} >$ and computes:$N_1 = r_1.P, RN_i = (N_1 \| \mathcal{C}^*_{ia}) \oplus ID_{ia}$ and $\gamma_{ia} = h(ID_{ia} \| N_1 \| \mathcal{R}^*_{ia} \| X_{ia})$.Afterwards, $\mathcal{D}_{ia}$ sends sign request message $MSG_{ia} = \{RN_i, \gamma_{ia}, Tid_i\}$ to its domain server $\mathcal{S}_{sa}$.
2. On receiving $MSG_{ia}, \mathcal{S}_{sa}$ obtains $\{ID_{ia}, \mathcal{E}_{ia}, K_{ia}, C_{ia}\}$ by Tid_i and computes:$PK_{ia} = DerCpk(C_i, K_i, j)$ and $(N_1 \| \mathcal{C}^*_{ia}) = RN_i \oplus ID_{ia}$. Afterwards, $\mathcal{S}_{sa}$ gets $\mathcal{R}^*$ against $\mathcal{C}^*$ and computes:$X_{ia} = h(ID_{ia} \| SK_{sa})$, and checks $\gamma_{ia} \stackrel{?}{=} h(ID_{ia} \| N_1 \| \mathcal{R}^*_{ia} \| X_{ia})$.In case of equality, $\mathcal{S}_{sa}$ generates:n_2 and picks $< \mathcal{C}\prime_{ia}, \mathcal{R}\prime_{ia} >$.Further $\mathcal{S}_{sa}$ computes:$\theta_i = n_2.(PK_{ia} + ID_{ia}.PK_{sb}), RN_{sb} = n_2.N_1, N_2 = n_2.P, NR = N_1 \oplus (N_2 \| \mathcal{C}\prime_{ia}), SK = h(ID_{ia} \| N_2 \| N_1), SID = h(N_1 \| N_2)$ and $\gamma_b = h(ID_{ia} \| \mathcal{R}\prime_{ia} \| SID \| SK)$.Finally, $\mathcal{S}_{sa}$ transmits$MSG_{sa} = \{NR, \gamma_{sa}\}$ to $\mathcal{D}_{ia}$.
3. Whenever, $\mathcal{D}_{ia}$ receives MSG_{sa} from $\mathcal{S}_{sa}, \mathcal{D}_{ia}$ computes: $(N_2 \| \mathcal{C}\prime_{ia}) = NR \oplus N_1$. Thereafter, $\mathcal{D}_{ia}$ gets $\mathcal{R}\prime_{ia}$ against $\mathcal{C}\prime_{ia}$ and computes: $\theta_i = N_2 \mathcal{E}_{ia}$. $\mathcal{D}_{ia}$ then computes session key and session identity $SK = h(ID_{ia} \| N_2 \| N_1)$ and $SID = h(N_1 \| N_2)$, respectively, for the particular session. At the end, $\mathcal{D}_{ia}$ authenticates $\mathcal{S}_{sa}$ verifying the check: $\gamma_b \stackrel{?}{=} h(ID_{ia} \| \mathcal{R}\prime_{ia} \| SID \| SK \| \theta_i)$. If it does not match, $\mathcal{D}_{ia}$ instantly terminates the session. Elseways, $\mathcal{D}_{ia}$ accepts both SID and SK for secure communication with the intra-domain server.

3.6. Cross-domain authentication

In this phase, we describe the details of mutual cross domain authentication of an IoT device with the cross domain server. If an IoT device $\mathcal{D}_{ia}$ from a particular domain (e.g., domain a) wants to communicate with the IoT device $\mathcal{D}_{jb}$ of another domain (e.g., domain b), then $\mathcal{D}_{ia}$ needs to send authorization request towards the cross domain server (i.e., $\mathcal{S}_{sb}$). On receiving the request from $\mathcal{D}_{ia}, \mathcal{S}_{sb}$ queries the ledger information to verify whether the accumulator information of $\mathcal{D}_{ia}$ exists in the blockchain or not. In case of successful verification, the unilateral cross-domain authentication is accomplished.

1. Firstly, $\mathcal{D}_{ia}$ enters ID_{ia} and computes: $(k_{ia}, c_{ia}) = DerPsk(ID_{ia}, i, r_{ia}, X_{ia})$. After that,$\mathcal{D}_{ia}$ derives $\mathcal{R}_{ia}$ against $\mathcal{C}_{ia}$ and calculates $(sk_{i,j}) = DerCsk(k_{ia} \| c_{ia} \| \mathcal{R}_{ia})$. Further, $\mathcal{D}_{ia}$ generates n_3, picks a pair $< \mathcal{C}\prime_{ia}, \mathcal{R}\prime_{ia} >$ and computes: $N_3 = n_3.P$, $D_3 = n_3.PK_{sb}, \beta_{ia} = Enc_{(D_3)}(ID_{ia} \| ID_{jb} \| N_3 \| \mathcal{C}\prime_{ia} \| Tid_{ia})$, and $\gamma_{ia} = h(ID_{ia} \| D_3 \| N_3 \| \mathcal{R}\prime_{ia} \| sk_{i,j}.P)$. $\mathcal{D}_{ia}$ then invokes a login request message $MSG_1 \leftarrow \{\beta_{ia}, \gamma_{ia}, N_3\}$
2. On receiving the cross-domain authentication request message MSG_1 from $\mathcal{D}_{ia}, \mathcal{S}_{sb}$ computes: $(ID_{ia} \| ID_{jb} \| N_3 \| \mathcal{C}\prime_{ia} \| Tid_{ia}) = Dec_{(N_3.sk_{sb})}(\beta_{ia})$ and queries domain identity DID_a to obtain the information from the blockchain ledger: $QueryDomainInfo(DID_a)$ and $getsLedger(K_{ia} \| C_{ia} \| ACC_{ia} \| \mathcal{E}_{ia})$ by Tid_{ia}. After acquiring the desired information, $\mathcal{S}_{sb}$ further computes $PK_{i,j} = DerCpk(K_{ia} \| C_{ia}, j)$, and verifies $\gamma_{ia} \stackrel{?}{=} h(ID_{ia} \| D_3 \| N_3 \| \mathcal{R}\prime_{ia} \| PK_{i,j})$. If it holds, $\mathcal{S}_{sb}$ generates n_4 and $Picks < \mathcal{C}\prime_{jb}, \mathcal{R}\prime_{jb} >$, and calculates: $N_4 = n_4.N_3, \beta_{jb} = Enc_{(ID_{jb})}(ID_{ia} \| N_4 \| D_3 \| N_3 \| \mathcal{C}\prime_{jb})$ and $\gamma_{jb} = h(ID_{jb} \| \mathcal{C}\prime_{jb} \| \mathcal{R}\prime_{jb} \| N_4)$. Lastly, $\mathcal{S}_{sb}$ sends $MSG_2 \leftarrow \{\beta_{jb}, \gamma_{jb}\}$ to $\mathcal{D}_{ib}$.
3. Upon receiving $MSG_2, \mathcal{D}_{ib}$ calculates: $(ID_{ia} \| N_4 \| D_3 \| N_3 \| \mathcal{C}\prime_{jb}) = Dec_{(ID_{jb})}(\beta_b)$, and gets $\mathcal{R}\prime_{jb}$ *against* $\mathcal{C}\prime_{jb}$. Thereafter, $\mathcal{D}_{ib}$ verifies $\gamma_{jb} \stackrel{?}{=} h(ID_{jb} \| \mathcal{C}\prime_{jb} \| \mathcal{R}\prime_{jb} \| N_4)$. If it is verified, $\mathcal{D}_{ib}$ generates $n_5, < \mathcal{C}^*_{jb}, \mathcal{R}^*_{jb} >$, and computes: $N_5 = n_5.N_4, N_6 = n_5.N_3$. $\mathcal{D}_{ib}$ then queries the blockchain ledger $getsLedger\{PK_{sb}, PK_{ia}\}$ to acquire $\{PK_{sb}, PK_{ia}\}$. Next, $\mathcal{D}_{ib}$ calculates: $\theta = n_4(PK_{ia} + ID_a PK_{sb}), SID = h(N_5 \| N_6), SK = h(ID_{ia} \| ID_{jb} \| SID \| \theta), \eta = Enc_{(D_3)}(N_5 \| N_6 \| \mathcal{C}^*_{jb} \| n_4)$ and $\tau_b = h(ID_{jb} \| SID \| SK \| N_5 \| \mathcal{R}^*_{jb})$. At the end, $\mathcal{D}_{ib}$ sends $MSG_3 \leftarrow \{\eta_b, \tau_b\}$ towards $\mathcal{D}_{ia}$.
4. Whenever $\mathcal{D}_{ib}$ receives MSG_3 from $\mathcal{D}_{ib}$, then $\mathcal{D}_{ia}$ computes: $(N_5 \| N_6 \| \mathcal{C}^*_{jb} \| n_4) = Dec_{(D_3)}(\eta)$ and gets $\mathcal{R}^*_{jb}$ *against* $\mathcal{C}^*_{jb}$. Thereafter, $\mathcal{D}_{ia}$ determines $\theta = n_4 \mathcal{E}_{ia} P$ and calculates session identity and session key $SID = h(N_5 \| N_6)$ and $SK = h(ID_{ia} \| ID_{jb} \| SID \| \theta)$, respectively. Finally, $\mathcal{D}_{ia}$ authenticates $\mathcal{D}_{ib}$ by checking $\tau_b = h(ID_{jb} \| SID \| SK \| N_5 \| \mathcal{R}^*_{jb})$. In case of successful verification, $\mathcal{D}_{ia}$ believes that $\mathcal{D}_{ib}$ is legitimate and will use the session identity and session key to protect its communication over public channel for the particular session.

The proposed protocol is also summarized in Fig. 2.

4. Security evaluation

In this section, we discuss how our designed protocol prevents known security threats. We also present a detail discussion about the key security features of our protocol.

4.1. Intra-domain mutual-authentication

In our protocol, IoT device $\mathcal{D}_{ia}$ and server of $\mathcal{S}_{sa}$ of any designated domain a mutually authenticate each other before establishing the session identity and session key, respectively. Whenever, $\mathcal{D}_{ia}$ transmits $MSG_{ia} = \{RN_i, \gamma_i, Tid_i\}$ toward $\mathcal{S}_{sa}, \mathcal{S}_{sa}$ verifies the legitimacy of $\mathcal{D}_{ia}$ on checking whether $\gamma_{ia} \stackrel{?}{=} h(ID_{ia} \| N_1 \| \mathcal{R}^*_{ia} \| X_{ia})$. If this check holds the desired value, it means $\mathcal{D}_{ia}$ has passed the authentication. It is to be noted that only legal $\mathcal{S}_{sa}$ can verifies this check since the computation of γ_{ia} requires the secret key SK_{sa} of $\mathcal{S}_{sa}$. On the other hand, $\mathcal{D}_{ia}$ authenticates $\mathcal{S}_{sa}$ on $\gamma_b \stackrel{?}{=} h(ID_{ia} \| \mathcal{R}\prime_{ia} \| SID \| SK \| \theta_i)$. $\mathcal{D}_{ia}$ accepts session key only if it has successfully verified $\mathcal{S}_{sa}$. Hence, our protocol ensures the intra-domain mutual authentication between $\mathcal{D}_{ia}$ and $\mathcal{S}_{sa}$.

4.2. Cross-domain authentication

Our protocol allows an IoT $\mathcal{D}_{ia}$ from any particular domain (i.e., a) to communicate with cross-domain IoT device $\mathcal{D}_{ib}$. Such communication requires authentication and establishment of session key among the devices. The cross-domain authentication among $\mathcal{D}_{ia}, \mathcal{S}_{sb}$ and $\mathcal{D}_{ib}$ takes place in the following manner:

Fig. 2. Summary of authentication phases.

- $\mathcal{D}_{ia} \rightarrow \mathcal{S}_{sb} : MSG_1 \leftarrow \{\beta_{ia}, \gamma_{ia}, N_3\}$: Here, $\mathcal{S}_{sb}$ checks $\gamma_{ia} \stackrel{?}{=} h(ID_{ia}\|D_3\|N_3\|\mathcal{R}\prime_{ia}\|sk_{i,j}.P)$ to authenticate $\mathcal{D}_{ia}$.
- $\mathcal{S}_{sb} \rightarrow \mathcal{D}_{jb} : MSG_2 \leftarrow \{\beta_{jb}, \gamma_{jb}\}$: Here, $\mathcal{D}_{jb}$ checks $\gamma_{jb} \stackrel{?}{=} h(ID_{jb}\|\mathcal{C}\prime_{jb}\|\mathcal{R}\prime_{jb}\|N_4)$ to authenticate $\mathcal{S}_{sb}$.
- $\mathcal{S}_{sb} \rightarrow \mathcal{D}_{jb} : MSG_3 \leftarrow \{\eta_b, \tau_b\}$: Here, $\mathcal{D}_{ia}$ checks $\tau_b \stackrel{?}{=} h(ID_{jb}\|SID\|SK\|N_5\|\mathcal{R}^*_{jb})$ to authenticate $\mathcal{D}_{ia}$.

4.3. Anonymity and unlinkability

Our designed protocol preserves the identities of IoT devices. During the establishment of each session (whether it is intra-domain or cross-domain), rather than sharing real identity $ID_i, \mathcal{D}_i$ sends its temporary identity Tid_i toward $\mathcal{S}_s$. Moreover, not even a single secret credential can be acquired from the ledger of the blockchain and Tid_i is unique in each session. Therefore, $\mathbb{A}$ will remain unable to link messages different messages of two session to the same $\mathcal{D}_i$. Hence, our protocol preserves anonymity and support unlinkability.

4.4. Strong forward secrecy

The session key *SK* and session identity *SID* in both intra-domain and cross-domain authentications are freshly generated in each session and ephemeral in nature. *SK* and *SID* in intra-domain authentication are computed as $SK = h(ID_{ia}\|N_2\|N_1)$ and $SID = h(N_1\|N_2)$, respectively. On the other hand, during the cross-domain authentication, these values are generated as $SID = h(N_5\|N_6), SK = h(ID_{ia}\|ID_{jb}\|SID\|\theta)$, respectively. From these

cases, it is evident that *SK* and *SID* are composed of both independent secrets (i.e., identities of devices) and ephemeral nonces. Moreover, $\mathbb{A}$ can not break the security of ECDLP hard problem. Therefore, it is infeasible for $\mathbb{A}$ to generate *SK* and *SID* of any session. Hence, our protocol offers strong forward secrecy.

4.5. Other security functionalists

Here, we demonstrate how our protocol prevents $\mathbb{A}$ to mount several potential attacks.

4.5.1. Impersonation attack resistance

To act as real IoT device $\mathcal{D}_i$ or server $\mathcal{S}_s$, $\mathbb{A}$ needs to pass all the authentication checks that are demonstrated in Sections 4.1 and 4.2. Since we have already proved the security and robustness of mutual authentication process in the earlier sections. Therefore, $\mathbb{A}$ can never succeed in making attempt to impersonate any entity.

4.5.2. Physical and cloning attack resistance

In the design of our protocol, we attached PUF function to each IoT device. Due to the independent, secure and distinct nature, it is impossible for $\mathbb{A}$ to predict or make a clone of real response message $\mathcal{R}$. If $\mathbb{A}$ attempts to physically capture any IoT device aiming to corrupt the PUF, then it will fluctuate the inherent desire output of PUF. Consequently, $\mathbb{A}$ can not reproduce real $\mathcal{R}$ from the corrupted IoT device. Ultimately, our protocol preserves the security against physical and cloning attacks.

4.5.3. Replay attack resistance

The involvement of ephemeral credentials in the computation of each message for every session, no one can resuse the same message once the session has been expired. Therefore, our protocol restricts $\mathbb{A}$ from replaying any previous message to pass the verification checks. Hence, our protocol withstands the replay attack.

5. Performance analysis

In this section, we conduct the experimental results to determine the performance of the proposed protocol. The performance is measured using the following metrics: i) computation overhead, ii) communication overhead, iii) storage overhead and iv) security features comparison. The experimental results of these metrics are briefly discussed in subsequent subsections..

Table 2
Unit running time of key cryptographic operations for IIoT domains (in milliseconds).

Operation	Description	Running Time	
		Arduino	System
$T_{enc/dec}$	Execution time of enc/dec	3.10 ms	0.40 ms
T_{Pm}	Execution time of point multiplication	3.80 ms	0.55 ms
T_{Sm}	Execution time of scalar multiplication	4.90 ms	0.80 ms
T_{Pa}	Execution time of point addition	3.20 ms	0.50 ms
T_H	Execution time of hash function	2.90 ms	0.25 ms
T_{Ex}	Execution time of exponentiation	3.50 ms	0.35 ms

5.1. Experimental setup

In this experiment, we simulated two IIoT domains. Each domain comprises of a trusted authority, a server and an IIoT device. Trusted authorities and server of both the domains are implemented on a system with following specifications: Intel Core i7, 2.9 GHz processor and 32 GB RAM. On the other hand, the cryptographic operations of IIoT devices for both domains are implemented on Arduino device. The key cryptographic operations and their running time are listed down in Table 2.

5.2. Analysis of computation overhead

To compute the computation overhead of the proposed protocol, we count the most time consuming cryptographic operations discussed in Table 2. Later on, the computation overhead of proposed protocol is compared with the related protocols [19,20,12]. It is worth mentioning here that the computation overhead of cross domain authentication process of proposed and related protocols is computed only. The both devices of proposed protocol takes $(4T_H, 2T_{Pm}, 2T_{enc/dec})$ and $(2T_H, 1T_{Pm}, 2T_{enc/dec})$ cryptographic operations, respectively. Similarly, the server side of proposed protocol takes $(2T_H, 1T_{Pm}, 2T_{enc/dec})$ operations to complete the authentication process. To compute the overall computation overhead of all these cryptographic operations, we consider the running time mentioned in Table 2 with respect to the implementation devices. The overall computation overhead of the proposed protocol is 42.45 ms. The computation overhead of related protocols [19,20,12,21] is computed in same way and presented in Table 3 and Fig. 3. The comparison shown in Table 3 clearly indicates the supremacy of proposed protocol over related protocols in terms of computation overhead.

5.3. Analysis of communication overhead

This section solicits the communication overhead comparison of the proposed and related protocols. The communication overhead refers to the required number of bits to transmit the messages among the participating entities of an authentication protocol to accomplish the login and authentication phase. Table 4 represents the assumptions taken to calculate the communication overhead of proposed and related protocols.

The communication overhead of proposed protocol is calculated in the following way: In our protocol, the device $\mathcal{D}_{ia}$ transmits a message $MSG_1 \leftarrow \{\beta_{ia}, \gamma_{ia}, N_3\}$ towards $\mathcal{S}_{sb}$. In MSG_1, $\beta_{ia} = Enc_{(D_3)}(ID_{ia}, ID_{jb}, D_3, N_3, \mathcal{C}\prime_{ia}, Tid_{ia})$, $\gamma_{ia} = h(ID_{ia}\|D_3\|N_3\|\mathcal{R}\prime_{ia}\|sk_{i,j}.P)$ and $N_3 = n_3.P$ takes 128, 256 and 160 bits, respectively. So, the communication overhead of MSG_1 is 128 + 256 + 160 = 544 bits. Similarly, $\mathcal{S}_{sb}$ sends a message $MSG_2 \leftarrow \{\beta_{jb}, \gamma_{jb}\}$ towards $\mathcal{D}_{jb}$, where $\beta_{jb} = Enc_{(ID_{jb})}(ID_{ia}, N_4, N_3, \mathcal{C}\prime_{jb})$ and $\gamma_{jb} = h(ID_{jb}\|\mathcal{C}\prime_{jb}\|\mathcal{R}\prime_{jb}\|N_4)$. Therefore, the transmission of MSG_2 takes 128 + 256 = 384 bits. In the end, device $\mathcal{D}_{jb}$ transmits a message $MSG_3 \leftarrow \{\eta_b, \tau_b\}$ towards $\mathcal{S}_{sb}$, where $\eta = Enc_{(D_3)}(N_5, N_6, \mathcal{C}^*_{jb}, n_4)$ and $\tau_b = h(ID_{jb}\|SID\|SK\|\ N_5\|\mathcal{R}^*_{jb})$. Hence, the required number of bits for the transmision of MSG_3 are 128 + 256 = 384. The overall communication overhead for

Table 3
Comparison of computation overheads.

Protocols	Device $\mathcal{D}_{ia}$	Device $\mathcal{D}_{jb}$	Server $\mathcal{S}_{sb}$	Overall Computation Overhead
Our	$4T_H + 2T_{Pm} + 2T_{enc/dec}$	$2T_H + 1T_{Pm} + 2T_{enc/dec}$	$2T_H + 1T_{Pm} + 2T_{enc/dec}$	42.45 ms
Zhang et al. [19]	$4T_H + 6T_{Sm} + 2T_{Ex} + 2T_{enc/dec}$	$6T_{Sm} + 2T_{Ex}$	$3T_H + 2T_{enc/dec}$	92.15 ms
Li et al. [20]	$2T_H + 3T_{Sm} + 2T_{enc/dec}$	$3T_H + 3T_{Sm} + 2T_{enc/dec}$	$3T_H + 2T_{Sm} + 2T_{enc/dec}$	59.65 ms
Shen et al. [12]	$2T_H + 3T_{Sm} + 1T_{Ex}$	$3T_H + 3T_{Sm} + 1T_{Ex}$	$3T_H + 4T_{Sm}$	54.85 ms
Zhang et al. [21]	$7T_H + 2T_{enc/dec}$	$5T_H + 2T_{enc/dec}$	$5T_H + 2T_{enc/dec}$	49.25 ms

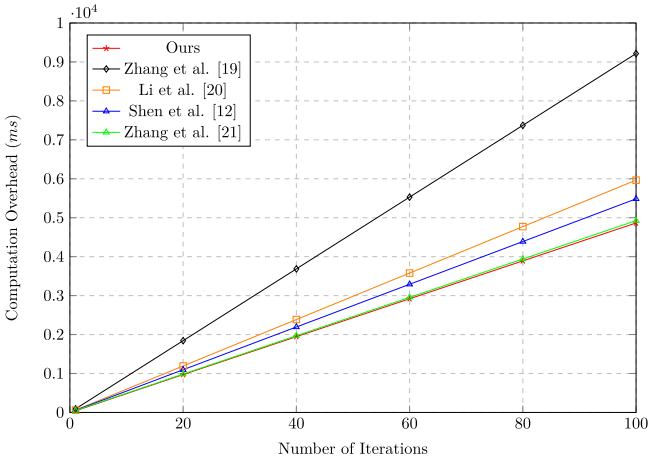


Fig. 3. Computation overheads graphical comparison.

Table 4
Assumptions for communication overheads.

Attribute	Required Bits
Identity	160
XoR	160
Time stamp	160
Point multiplication	160
Concatenation	160
Hash function	256
Signature	256
Symmetric encryption/decryption	128
Bi-linear Pairing	160

Table 5
Communication overheads comparison.

Protocol	Overall Communication Overhead
Our	1312 bits
[19]	3616 bits
[20]	1568 bits
[12]	1952 bits
[21]	2080 bits

the cross domain authentication process of proposed protocol is (544 + 384 + 384)= 1312 bits. The communication overhead of related protocols [19,20,12,21] is also determined in the same way and shown in Table 5 and Fig. 4. It is obvious from this comparison that the proposed protocol takes least number of bits for communication as compared to related protocols.

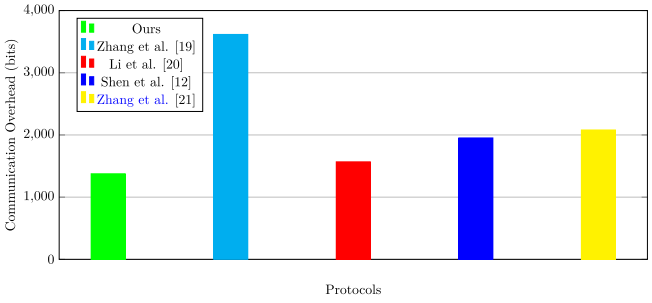


Fig. 4. Communication overheads graphical comparison.

5.4. Storage overhead comparison

As far as the storage overhead on the blockchain for the proposed protocol is concerned, we have presented the comparison by considering 1000 public keys for each domain. In this scenario, the proposed protocol takes 0.172 KB of storage resource for a single domain. On the other hand, the related protocols [19,20,12,21] take 0.452, 0.196, 0.244 and 0.320 KBs, respectively as storage resources. Fig. 5 shows the supremacy of the proposed protocol over related protocols regarding storage resources according to the growth of the number of domains. The benefit of the proposed protocol can be justified using the trust-building method used in our protocol. Instead of increasing the number of public keys, our method stores the accumulator in the blockchain. Therefore, the proposed protocol requires fewer storage resources as compared to the related protocols.

5.5. Security features comparison

In this section, we discuss the security features comparison between the proposed and related protocols [19,20,12,21]. The comparison is listed down in Table 6, which shows that the related protocols do not provide resistance against desynchronization attack. Furthermore, all of the related protocols do not claim the resilience against impersonation attacks. While, the proposed protocol not only resists the major security threats but also takes less on chain storage overhead as compared to all related protocols. Hence, it can be claimed that proposed protocol secure as well as ensures aided security features as compared to related protocols.

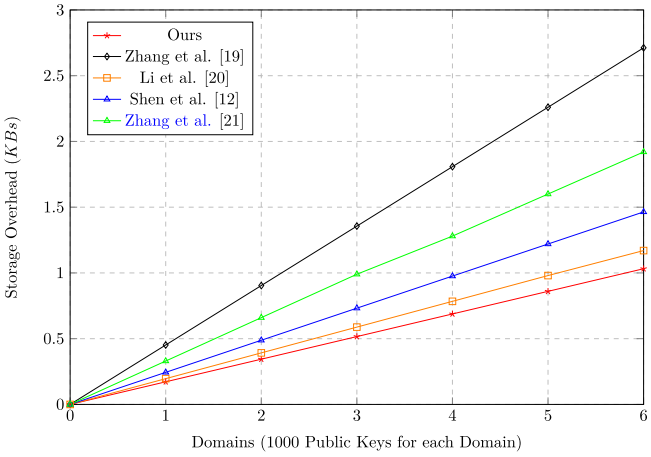


Fig. 5. Storage overheads graphical comparison.

Table 6
Security features comparison.

Protocols → / Security Features ↓	Ours	[19]	[20]	[12]	[21]
Anonymity and Privacy	✓	✓	✓	✓	✓
Physical Attack Resilience	✓	✗	✗	✗	✗
Desynchronization Attack Resilience	✓	✗	✗	✗	✗
Impersonation Attack Resilience	✓	✗	✗	✗	✓
Mutual Authentication	✓	✓	✓	✓	✓
Perfect Forward Secrecy	✓	✗	✓	✓	✓
Lower Storage Overhead	✓	✗	✗	✗	✗

Note: ✓ Claims/provides; ✗ Doesn't claim/provide.

6. Conclusion

To secure cross-domain IIoT device cooperation, we have designed an effective privacy-preserving device authentication protocol leveraging blockchain technology. The security analysis demonstrates that our protocol offers database protection and resilience to potential security threats. Additionally, in order to protect privacy, anonymity and unlinkability are also guaranteed. Performance analysis demonstrates the effectiveness and viability of our protocol. We aim to further explore the protocol's scalability by evaluating its performance under a variety of network conditions and with increasing traffic loads in the future. There is potential for this ongoing study to provide valuable insights that will guide future research endeavors and optimize the protocol's effectiveness.

Data Availability

No data was used for the research described in the article.

Compliance with ethics requirements

This article does not contain any studies with human or animal subjects.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work is partially funded by Brazilian National Council for Scientific and Technological Development - CNPq, via Grant No. 313036/2020-9. This work was also supported by the DoD Center of Excellence in AI and Machine Learning (CoE-AIML) under Contract Number W911NF-20-2-0277 with the U.S. Army Research Laboratory.

References

[1] Glaessgen E, Stargel D. The digital twin paradigm for future nasa and us air force vehicles. In: 53rd AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference 20th AIAA/ASME/AHS adaptive structures conference 14th AIAA; 2012. p. 1818.
[2] Dai HN, Zheng Z, Zhang Y. Blockchain for internet of things: a survey. IEEE Internet of Things J 2019;6(5):8076–94.
[3] Ferrag MA, Derdour M, Mukherjee M, Derhab A, Maglaras L, Janicke H. Blockchain technologies for the internet of things: Research issues and challenges. IEEE Internet of Things J 2018;6(2):2188–204.
[4] Peng K, Li M, Huang H, Wang C, Wan S, Choo KKR. Security challenges and opportunities for smart contracts in internet of things: A survey. IEEE Internet of Things J 2021;8(15):12004–20.
[5] Ferrag MA, Shu L. The performance evaluation of blockchain-based security and privacy systems for the internet of things: A tutorial. IEEE Internet of Things J 2021;8(24):17236–60.
[6] Ali G, Ahmad N, Cao Y, Khan S, Cruickshank H, Qazi EA, et al. xdbauth: Blockchain based cross domain authentication and authorization framework for internet of things. IEEE Access 2020;8:58800–16.
[7] Guo S, Wang F, Zhang N, Qi F, Qiu X. Master-slave chain based trusted cross-domain authentication mechanism in iot. Journal of Network and Computer Applications 2020;172:102812.
[8] Wang L, Tian Y, Zhang D. Toward cross-domain dynamic accumulator authentication based on blockchain in internet of things. IEEE Trans. Industr. Inf. 2021;18(4):2858–67.
[9] Tang, B., Kang, H., Fan, J., Li, Q., Sandhu, R. Iot passport: A blockchain-based trust framework for collaborative internet-of-things. In: Proceedings of the 24th ACM symposium on access control models and technologies. 2019, p. 83–92.
[10] Ma M, Shi G, Li F. Privacy-oriented blockchain-based distributed key management architecture for hierarchical access control in the iot scenario. IEEE access 2019;7:34045–59.
[11] Jia X, Hu N, Su S, Yin S, Zhao Y, Cheng X, et al. Irba: an identity-based cross-domain authentication scheme for the internet of things. Electronics 2020;9(4):634.
[12] Shen M, Liu H, Zhu L, Xu K, Yu H, Du X, et al. Blockchain-assisted secure device authentication for cross-domain industrial iot. IEEE J. Sel. Areas Commun. 2020;38(5):942–54.
[13] Chen R, Shu F, Huang S, Huang L, Liu H, Liu J, et al. Bidm: a blockchain-enabled cross-domain identity management system. Journal of Communications and Information Networks 2021;6(1):44–58.
[14] Lei A, Cruickshank H, Cao Y, Asuquo P, Ogah CPA, Sun Z. Blockchain-based dynamic key management for heterogeneous intelligent transportation systems. IEEE Internet of Things J 2017;4(6):1832–43.
[15] Lin C, He D, Huang X, Kumar N, Choo KKR. Bcppa: A blockchain-based conditional privacy-preserving authentication protocol for vehicular ad hoc networks. IEEE Trans. Intell. Transp. Syst. 2020;22(12):7408–20.
[16] Kang J, Yu R, Huang X, Wu M, Maharjan S, Xie S, et al. Blockchain for secure and efficient data sharing in vehicular edge computing and networks. IEEE Internet of Things J 2018;6(3):4660–70.
[17] Cheng G, Chen Y, Deng S, Gao H, Yin J. A blockchain-based mutual authentication scheme for collaborative edge computing. IEEE Trans Comput Social Syst 2021;9(1):146–58.
[18] Zhang H, Chen X, Lan X, Jin H, Cao Q. Btcas: A blockchain-based thoroughly cross-domain authentication scheme. Journal of Information Security and Applications 2020;55:102538.
[19] Zhang Y, Li B, Wu J, Liu B, Chen R, Chang J. Efficient and privacy-preserving blockchain-based multi-factor device authentication protocol for cross-domain iiot. IEEE Internet of Things J 2022.
[20] Li G, Wang Y, Zhang B, Lu S. Smart contract-based cross-domain authentication and key agreement system for heterogeneous wireless networks. Mobile Information Systems 2020;2020.
[21] Zhang S, Du X, Liu X. A novel and quantum-resistant handover authentication protocol in iot environment. Wireless Netw. 2023:1–18.