Internet of Things 22 (2023) 100708

Contents lists available at ScienceDirect

Internet of Things

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

Research article

An access control scheme in IoT-enabled Smart-Grid systems using blockchain and PUF

Amina Zahoor [a], Khalid Mahmood [b], Salman Shamshad [c], Muhammad Asad Saleem [d], Muhammad Faizan Ayub [a], Mauro Conti [e], Ashok Kumar Das [f,*]

[a] Department of Computer Science, COMSATS University Islamabad, Sahiwal Campus 57000, Pakistan
[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, Pakistan
[e] Department of Mathematics, University of Padua, 35131 Padua, Italy
[f] Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad 500 032, India

ARTICLE INFO

Keywords:
Internet of Things (IoT)
Blockchain
Smart power grids
Authentication
Security
Key agreement

ABSTRACT

IoT-enabled Smart Grid (IoT-SG) is an emerging paradigm that enables the bi-direction communication of IoT devices and hardware to efficiently collect and transmit the consumer's information over the Internet. However, the underlying open communication network and resource-constrained capabilities of devices invite manifold challenges and threats in terms of security and privacy. To alleviate such issues, we designed a private blockchain-based access control protocol for IoT-SG using Physically Unclonable Function (PUF). Our protocol allows the Service Provider (SP) and smart meters to transfer the data in an efficient and secure manner. The participating SPs form the Peer-to-Peer (P2P) network, and each peer node is responsible for securely creating the blocks from the gathered data. Thereafter, all the peer nodes employ a voting-based consensus mechanism to verify and add the recently created block into the blockchain network. We have rigorously validated the security of our protocol under the Random or Real (RoR) model. The testbed results and security feature analysis show that our protocol is more efficient than competing ones while it also provides significant security properties.

1. Introduction

The development of Internet-of-Things (IoTs) infrastructure and the significant population expansion has dramatically increased the demand and popularity of IoT-enabled Smart Grid (IoT-SG). The IoT-SG is integrated with objects, actuators, and smart sensors to offer a reliable energy transmission and automation while improving the system's efficiency. It also enhances the delivery of quality services and economic benefits [1]. Despite of its various benefits, IoT-SG architecture still faces many challenges, including poor interoperability, transparency, centralized control and energy trading among untrusted networks [2]. Therefore, it is necessary to build a decentralized mechanism that can remedy all the limitations of a traditional centralized IoT-SG system. Furthermore, considerable work is required to increase efficiency, transparency, resilience, reliability, quality of services, and fraud prevention in IoT-SG [3]. In the recent years, the advent of blockchain technology in IoT-SG has overcome all the severe challenges of a

* Corresponding author.
E-mail addresses: aminazahoor890@gmail.com (A. Zahoor), khalid.mahmood@unipd.it (K. Mahmood), salmanshamshad01@gmail.com (S. Shamshad), masadsaleem123@gmail.com (M.A. Saleem), faizanayub9@gmail.com (M.F. Ayub), mauro.conti@unipd.it (M. Conti), ashok.das@iiit.ac.in (A.K. Das).

https://doi.org/10.1016/j.iot.2023.100708
Received 3 November 2022; Received in revised form 15 January 2023; Accepted 28 January 2023
Available online 1 February 2023
2542-6605/

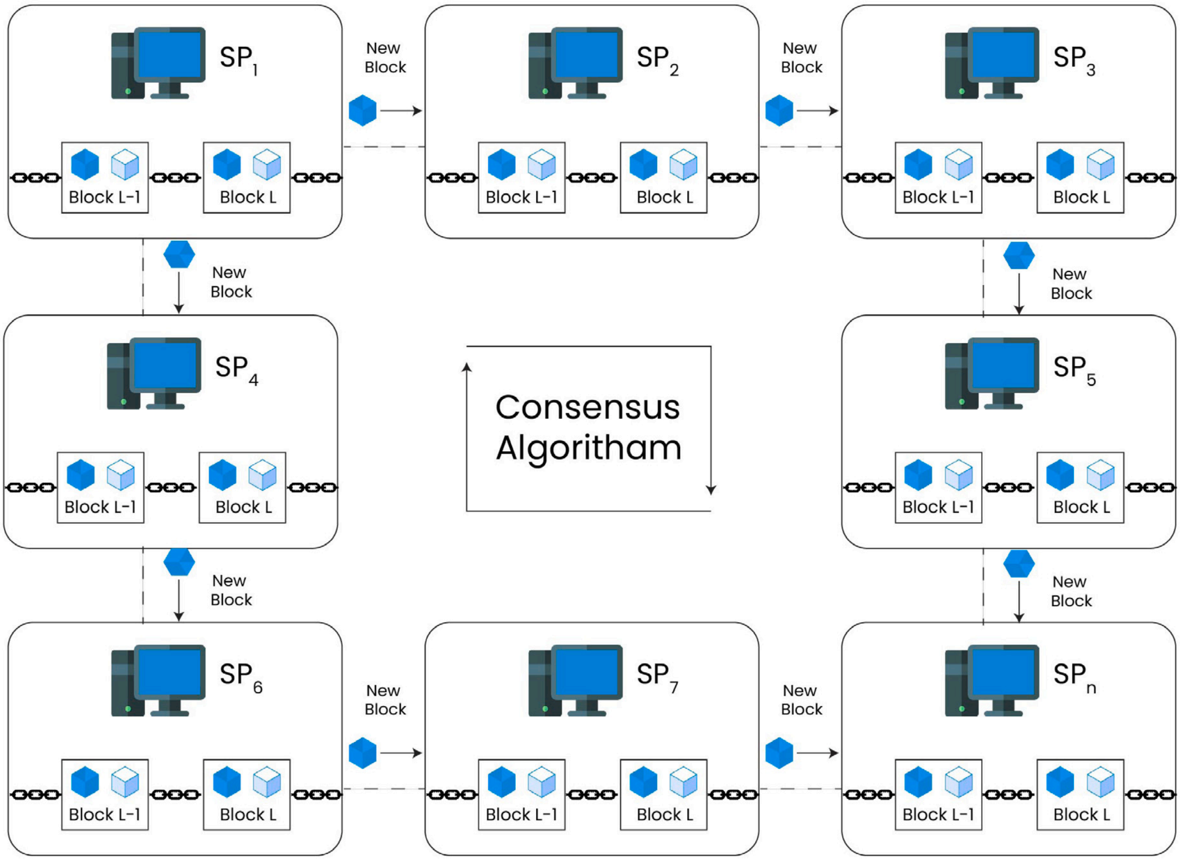

Fig. 1. Block distribution in blockchain network.

Table 1
A comparative summary of existing authentication schemes.

Scheme	Cryptographic primitives	Advantages	Disadvantages
Wang et al. [5]	* Hash function * ECC	* Provide mutual authentication * Provides key agreement * Provide blockchain solutions	* No anonymity * No voting based consensus algorithm for blockchain mining * Vulnerable to leader selection in $P2P$ network
Qi and Chen [6]	* Hash function * ECC	* Provide mutual authentication * Provide key agreement	* Unable to add dynamic nodes after deployment * Unable to support blockchain solutions
Chaudhry et al. [7]	* Hash function * ECC	* Provide mutual authentication * Provides session key agreement	* Unable to provide blockchain solutions
Bera et al. [8]	* Hash function * ECC	* Provide mutual authentication * Provides key establishment * Provides blockchain solution	* Unable to provide physical security * No implementation in the real world environment

conventional IoT-SG centralized infrastructures. The block-chain-based IoT-SG infrastructure can offer several advantages such as immutability, decentralization, confidentiality, trust, and transparency of shared data [4]. Generally, there are three categories of blockchain: (1) private blockchain, (2) consortium Blockchain, and (3) Public Blockchain. The private block-chain is considered as an entirely trustworthy network, where certain participant or a group of trusted participants has privilege to access the data. Additionally, the network owner primarily decides which participant will execute a particular task which is referred as permissioned blockchain. Generally, a permissioned blockchain has more ability to preserve privacy and security than a permissionless blockchain. Permissionless blockchains are also referred to as public or trustless blockchains. On the other hand, a consortium block-chain is a "combination of private and public block-chains". Consensus algorithms are required to gain consensus between the nodes involved in a peer-to-peer (P2P) block-chain network.

A $P2P$ network of SPs and their role in a smart electrical grid are shown in Fig. 1. A blockchain-based smart grid system consists of different entities, including service providers (SP_n), Trusted Authority (TA), smart meters (SM_m), and users who are connected to an SM_m. The communications among SMs and SPs must be protected against passive and active attacks [9]. In order to assure the privacy and security of consumers' secret information, it is crucial to develop an efficient and secure access control scheme between SPs and SMs. Secure information can be kept in form of blocks in a private block-chain using block-chain technology. Before new blocks are added to the block-chain using the consensus algorithm, the SPs in the peer 2 peer assisted SP system are responsible for authenticating them. In the recent years, designing of blockchain-assisted authentication mechanisms has become a hot research topic due to its transparency, data integrity and decentralization properties. For example, Musleh et al. [4] offered a survey that considers several mechanisms, benefits, and research hurdles when examining the feasibility of implementing a block-chain-based solution in a smart-grid system. In addition, they provided frameworks that are necessary for smart grid applications built on the block-chain. Furthermore, they highlighted that a block-chain is a viable option for integrating the smart grid's cyber–physical layer. Like the industrial sector, they predicted that the power grid would effectively use block-chain technology. Andoni et al. [3] looked at several academic and industrial sources to offer fundamentals of block-chain technologies, like "distributed consensus algorithms" and "network architectures", vital elements of performance for block-chain ecosystems. They emphasized several domains where innovation is crucial for stakeholders in the energy system and advanced industrial participants. Kim and Huh [10] introduced a smart grid design based on the block-chain concept and power trading system. Their technology makes it easier for P2P transactions to transfer power information from an existing smart grid network using transitory smart contract mechanisms. Wang et al. [11] introduced a distributed data integrity scheme based on blockchain that does not rely on TPA. The blockchain keeps the user's information, which is maintained by the cloud service providers. Moreover, Wang et al. [12] proposed an identity-based proxy re-encryption plus protocol to regulate safe social cloud data sharing. This process analyzed the security measurement by primitive bilinear map for the standard cryptographic techniques.

Chaudhry et al. [7] suggested a DRMAS mechanism for smart grid edge computing. Block-chain technology is also not supported in this particular scheme. After that, Tsai and Lo [9] offered anonymous key distribution. Their solution relays elliptic curve cryptography (ECC) and "bilinear pairings" to allow mutual authentication between SM and SP authentication. Mutual authentication creates a session key for confidential communication. Nevertheless, their approach is vulnerable to ESL threats and lacks credential privacy [13]. To overcome Tsai and Lo's [9] shortcomings, [13] suggests an alternative authentication approach. According to Wang et al. [5], mutual authentication is not attained in [14] because an SM does not verify utility control validity. Furthermore, Wang et al. [5] introduced a blockchain-based mutual authentication scheme. Due to use of block-chain, their scheme offers key management and conditional privacy. Gai et al. [15] designed a "permissioned block-chain edge in a smart power grid network". Their model uses block-chain and edge computing to protect privacy and energy. In addition, they utilized channel authorization and group signatures procedures to validate the users actively participating in smart power grid. Zhang et al. [16] developed a "signature key-less decentralized mechanism based on the consortium block-chain to provide a efficient system for the key management system". $P2P$ block-chain networks are used for data transmission. Their method involves the SMs sending queries and receiving responses to those requests. They proposed a decentralized consensus technique without the need for a TA. Zhou et al. [17] presented a power based access control scheme using block-chain technology. This mechanism would rely on "digital signature", "signcryption", and "identity-based combination encryption", respectively. Their method resolves the problem of "key escrow", which refers to the untrustworthy actions of third parties. Bera et al. [8] introduced a block-chain access control scheme in the smart grid systems. They asserted that their scheme offers protection against various threats. However, their scheme fails to provide physical protection for the smart meters.

Due to its uniqueness and decentralized architecture, blockchain-based solutions have recently emerged as one of the most promising technologies to provide privacy and security in the smart grid environment. Since all the communication between the users, SMs, and SPs through a public environment. Therefore, an attacker has the opportunity to tamper with the information and can easily perform different passive and active attacks. Various authentication schemes have been proposed in recent years, despite most of the presented schemes having high communication and computation costs. As a result, it is inappropriate for resource-constrained participants, such as smart meters or sensors. The vulnerability of hardware-based security solutions to Man-in-the-Middle threats is one of their main drawbacks. In these threats, AK can easily clone the devices when the hardware security modules are stolen. The Physical Unclonable Functions (PUFs) provide the solution to this problem, as mentioned earlier. PUF was introduced as cryptographic primitives [18] in 2001. This method was introduced by Frikken et al. [19], Delvaux et al. [20], and Resende et al. [21], contains two processes. (i) Enrolment, which a verifier carries out prior to the authentication, and (ii) verification which assures the authentication. A PUF is a random physical entity contained in a physical structure. It is simple to construct but practically impossible to predict, clone, or duplicate, even if the correct manufacturing process is composed again. The authentication of PUFs relies on the use of so-called challenge (C_{SM_m}) and response (R_{SM_m}) pairs rather than a private key connected to the device identity.

We introduced a new access control scheme in IoT-Enabled smart grid system to solve these problems using blockchain and PUF. In this scheme, information is securely gathered from SMs by their corresponding SPs before being formed into the blocks and added to the blockchain using a voting-based consensus algorithm in peer to peer assisted SP system. Because of the transparency and immutability features offered by the blockchain, a block cannot be changed by an attacker or even by a legitimate user of the smart grid system once it has been uploaded to the system, and everyone can view the data contained in the block. We mainly focus on private blockchain in this work since SPs' data collected from SMs by SPs is confidential and private. Table 1 summarizes various existing schemes and their cryptographic primitives, benefits, and drawbacks. The innovative contributions of our proposed scheme are listed below:

- We introduce an access control scheme in IoT-Enabled smart power grid system using blockchain and PUF to address the security issues of existing proposed schemes. To the best of our knowledge, it is the first blockchain-based solution that not only offers protection against well-known security attacks but also provides foolproof protection against tempering/cloning attacks.

Table 2
Symbols and their descriptions.

Symbols	Descriptions
$E_s(p,q)$	Non-singular elliptic curve: $y^2 = x^3 + px + q\ (mod\ s)$ with $4p^3 + 27q^2 \neq 0\ (mod\ s)$
PP	Public Point in $E_s(p,q)$
$k.PP$	Elliptic curve point multiplication: $k.PP = PP + PP + \dots + PP$ (k times)
$M+N$	Elliptic curve point addition: $M,N \in E_s(p,q)$
$u*v$	Ordinary modular multiplication in $GF(s)$
TA, ID_{TA}	Trusted Authority and its unique identity
PID_{TA}	Pseudo-identity of TA
Pub_{TA}, mk_{TA}	Public and Private keys of TA, respectively, $Pub_{TA} = mk_{TA}.PP$
SP_n	nth Service Provider
SPN	Peer to Peer assisted SP system of all the registered SPs
ID_{SP_n}, TID_{SP_n}	Real and Temporary-identities of SP_n, respectively
Pub_{SP_n}, K_{SP_n}	Public and Private keys of SP_n, respectively, $Pub_{SP_n} = K_{SP_n}.PP$
$f(x,y)$	Symmetric bivariate t-degree polynomial over the Galois field $GF(s)$: $f(x,y) = \sum_{m=0}^{t}\sum_{n=0}^{t} a_{mn}x^m y^n$ where $a_{mn} \in Z_s = \{0,1,2,\dots,s-1\}$
SM_m, ID_{SM_m}	mth smart meter and its real identity
TID_{SM_m}	Temporary identity of SM_m
TC_{SM_m}	Temporal credential 0f SM_m
$Cert_{SP_n}$, $Cert_{SM_m}$	Certificates issued by the TA to SP_n and SM_m, respectively
(C_{SM_m}, R_{SM_m})	Challenge and response pairs
$\|$, $\oplus$	Concatenation and XOR operations, respectively
AK	Attacker
$h(.)$	One-way cryptographic hash method
$E_{S_{SP_j}}$, $D_{S_{SP_j}}$	Encryption and decryption using SP_n shared key, respectively

- The PUF in our scheme deals with the hardware base security issues, which can cause the Man-in-the-Middle (MITM) attack. It is simple to construct but practically impossible to predict, clone, or duplicate, even if the correct manufacturing process is composed again.
- IoT-SG architecture faces many challenges, including poor interoperability, transparency, centralized control and energy trading among untrusted networks. Therefore, to remedy all the limitations of a traditional centralized IoT-SG system. We used blockchain in our scheme which overcome all the severe challenges of a conventional IoT-SG centralized infrastructures.
- We have rigorously validated the security of our scheme under the Random or Real (RoR) model. The detailed informal security analysis of our scheme has been presented to ensure the achievement of security and privacy goals.
- We compare our proposed scheme's performance with other related schemes in terms of computation overheads, communication overheads, and security attributes. The testbed results and security feature analysis shows that our scheme is more efficient than competing ones while our scheme also provides significant security properties. As a result, our contribution defines a scheme with an enhanced level of security and privacy.

The rest of our article is organized as follows: Section 2 specifies the cryptographic preliminaries of this paper. In Section 3 our designed scheme and description are present. Sections 4 and 5 present our scheme security analysis and performance analysis. The conclusion is presented in Section 6.

2. Preliminaries

This section provides a detailed description on the system model of the smart grid system. Furthermore, the threat model is also explained in this section. The symbols utilized in this article are illustrated in Table 2.

2.1. Threat model

We consider the well-known accepted "Dolev–Yao (DY)" threat model [22] for our proposed scheme. According to DY threat model, an attacker (AK) can add different data, remove, modify the communication messages and intercept messages between communicating entities in the smart power grid system. In addition, end-point communicating participants (i.e., users/ consumers, SM_m, and SP_n) are not considered trustworthy entities in the system. We also consider that AK can physically capture SM_m because SM_m cannot be monitored 24 × 7. Once SM_m is physically compromised, all the information in its memory can be obtained by AK using a power analysis attack. Furthermore, we have also considered a widely accepted Canetti and Krawczyk (CK) threat model [23] in the scrutiny of our scheme. In this model, AK not only can capture the communication messages but it can also compromise secret keys, session states, and secret credentials if the information is stored in the memory of SM_m and SP_n.

2.2. System model

In this network model, many consumers connect with SM_m, and many SMs are also associated with the SP_n. A group of SPs will create a $P2P$ assisted SP system, also known as $P2P$ assisted SP system. Offline registration of all installed SM_m and SP_n is the

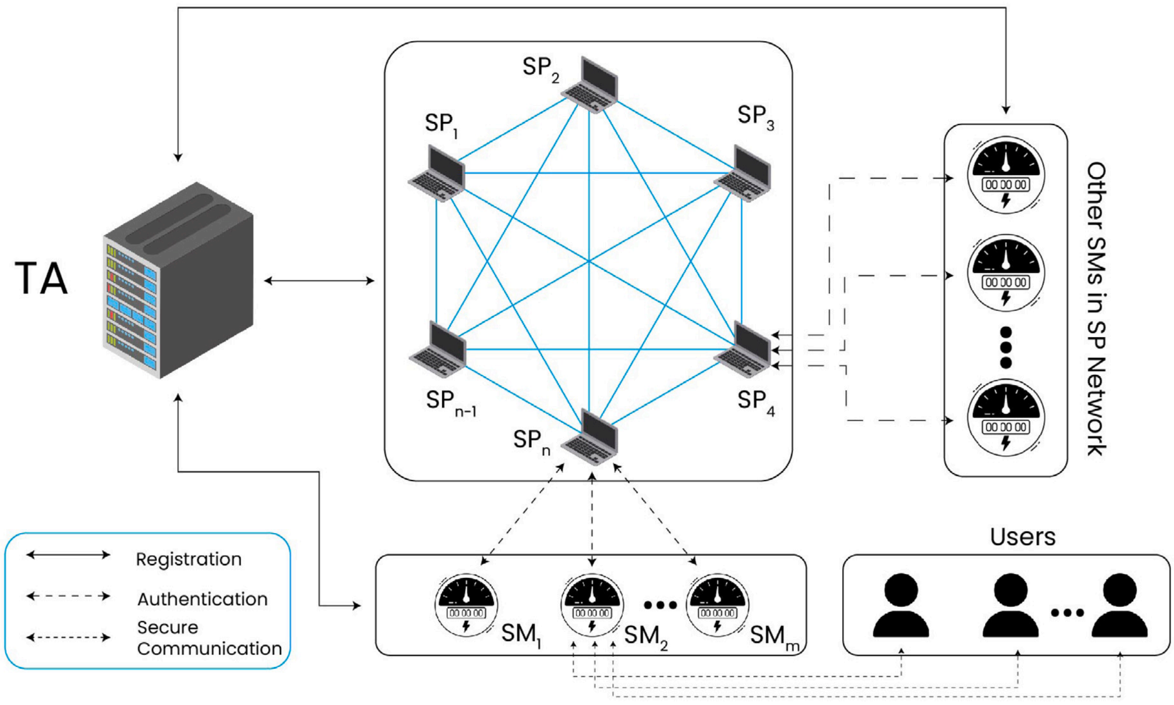

Fig. 2. Blockchain-based smart grid architecture without TTP.

responsibility of Trusted Authority (TA). TA executes the registration procedure securely. The consumers and SM_m communicate via secure communication. In contrast, SM_m and SP_n interact securely through the use of a shared session key and an access control mechanism. Furthermore, the SPs in SP system create private pairwise keys among themselves for the safe communication. According to this system model, SM_m first collects all the information privately from its connected consumers. Then, the gathered information is brought privately to SP_n, under which SM_m is registered with SP_n. With the help of collected information, SP_n then builds a block of transactions. Furthermore, the newly produced block can be uploaded to the current blockchain if the SPs in the SP system has reached a consensus. Once a block is added to the blockchain, it cannot be deleted or modified in order to keep the "immutability" property. The system model is depicted in Fig. 2.

3. Proposed scheme

On the basis of the architecture shown in Fig. 2, a new scheme is designed in this section. The proposed scheme consists of five phases: (A) system setup phase, (B) registration phase, (C) access control phase, (D) block formation and addition, and (E) dynamic node update phase.

3.1. System setup phase

The Trusted Authority (TA) selects the system parameters. TA performs the following steps:

SSP 1: TA picks a non-singular elliptic curve $y^2 = x^3 + px + q \ (mod \ s)$, where s is a large prime and $4p^3 + 27q^2 \neq 0 \ (mod \ s)$ with $\mathcal{O}$ as point at the infinity. Additionally, TA chooses a public/base point $PP \in E_s(p,q)$ whose order is as large as s, let us say s, or $n.PP = \mathcal{O}$.

SSP 2: Further, TA picks an identity ID_{TA}, and selects a master key mk_{TA} as the secret/private key and its respective public key as $Pub_{TA} = mk_{TA}.PP$, and also calculate its pseudo-identity $PID_{TA} = h(ID_{TA} \| mk_{TA})$.

SSP 3: Next, TA then selects a cryptographic hash method $h : \{0,1\}^* \to \{0,1\}^{lh}$, which generates lh bits fixed length output string, $h(x) \in \{0,1\}^{lh}$ with an arbitrary length input string $x \in \{0,1\}^*$. Furthermore, TA chooses a "elliptic curve digital signature algorithm (ECDSA)" to sign a message, which contains the verification and signature generation algorithm.

SSP 4: Finally, TA stores mk_{TA} as its secret key and publishes other information $\{E_s(p,q), h(.), PP, \ Pub_{TA}\}$ which are openly accessible to all the participants in the system.

3.2. Registration phase

TA executes this phase offline to register all the deployed smart meters (SM_m), ($m = 1, 2, \dots, m_{sm}$) and the service providers (SP_n), ($n = 1, 2, \dots, n_{sp}$).

3.2.1. Smart meter registration

The procedures listed below are vital to complete the registration phase of each installed SM_m.

SMR 1: For each SM_m, TA selects a real identity ID_{SM_m}, random number RN_m, and produces challenge and response pairs (C_{SM_m}, R_{SM_m}). TA selects a random private/secret key K_{SM_m} and computes its public key $Pub_{SM_m} = K_{SM_m}.PP$. Next, TA compute the temporary credentials for every SM_m as $TC_{SM_m} = h(ID_{SM_m} \| mk_{TA} \| K_{SM_m} \| RN_m \| C_{SM_m} \| R_{SM_m})$, $TID_{SM_m = E_{S_{SP_j}}(ID_{SM_m} \| TC_{SM_m} \| RN_m)}$, and certificate $Cert_{SM_m} = K_{SM_m} + h(PID_{TA} \| Pub_{TA} \| \ ID_{SM_m} \| C_{SM_m} \| R_{SM_m}) * mk_{TA}$ (mod s).

SMR 2: Finally, TA loads all the credential $\{TID_{SM_m}, TC_{SM_m}, PID_{TA}, ID_{SM_m}, Cert_{SM_m}, (C_{SM_m},\ R_{SM_m})\}$ in the SM_m memory. It also declares all Pub_{SM_m} as public. It is also important to note that each deployed SM_m over the network has distinct information, including $\{TID_{SM_m}, TC_{SM_m}, PID_{TA}, ID_{SM_m}, Cert_{SM_m}, (C_{SM_m}, R_{SM_m})\}$. Next, TA removes generated private key for every registered SM_m.

3.2.2. Service provider registration

TA completes the registration of each deployed SP_n, under which the SM_m, ($m = 1, 2, \dots, n_{sm}$) will be functional. TA performs the following steps:

SPR 1: For each SP_n, TA selects a real identity ID_{SP_n}, random number RN_n, and selects shared key S_{SP_j} for each SP_n. Next, TA compute $TID_{SP_n} = h(ID_{SP_n} \| mk_{TA} \| RN_n)$.

SPR 2: Next, TA pick a random private key K_{SP_n} and computes its public key $Pub_{SP_n} = K_{SP_n}.PP$. Afterward, TA computes $Cert_{SP_n} = K_{SP_n} + h(PID_{TA} \| Pub_{SP_n}) * mk_{TA}$ (mod s).

SPR 3: Finally, TA keeps all the credentials $\{TID_{SP_n}, PID_{TA}, Cert_{SP_n}, S_{SP_j}, f(TID_{SP_n}, y), \{TID_{SM_m} | m = 1, 2, \dots, n_{sm}\}\}$ in SP_n, remove generated private keys K_{SP_n} for every registered SP_n. Next, TA declares all the public keys Pub_{SP_n} as public. Furthermore, TA also keeps $\{(TID_{SP_l}),\ l \neq n,\ n = 1, 2, \dots, n_{sp}\}$ in SP_n corresponding to all other SP_l.

3.3. Access control phase

In this phase, SM_m and SP_n mutually authenticate each other before establishing a shared session key for future secure communication. The access control phase of the proposed scheme is shown in Fig. 3. The following steps are performed between SM_m and SP_n, as shown below:

ACP 1: SM_m produces a random number $r_m \in Z_s^*$ and then computes $Y_1 = h(r_m \| ID_{SM_m}).\ PP$ and $Y_2 = h(ID_{SM_m} \| PID_{TA} \| Y_1 \| Cert_{SM_m}.PP)$. Next, SM_m forwards the authentication request message $M_1 = \{Y_1, Y_2, TID_{SM_m}, Cert_{SM_m}\}$ to SP_n via a public channel.

ACP 2: SP_n after receiving the message M_1, first calculate $(ID_{SM_m} \| TC_{SM_m} \| RN_m) = D_{S_{SP_j}}(TID_{SM_m})$ and $Cert_{SM_m}.PP = Pub_{SM_m} + h(PID_{TA} \| Pub_{TA} \| ID_{SM_m} \| C_{SM_m} \| R_{SM_m}).Pub_{TA}$?, and check whether $Y_2' = h(ID_{SM_m} \| PID_{TA} \| Y_1 \| Cert_{SM_m}.PP)$ or not. If the verification fails, SP_n terminates the connection.

ACP 3: Otherwise, SP_n chooses a pair of $(C_{SM_m}^1, R_{SM_m}^1)$ and produces random secret $r_n \in Z_s^*$. Further, SP_n calculates $X_1 = h(r_n \| TID_{SP_n}).\ PP$, $DK_{mn} = h(r_n \| TID_{SP_n}).Y_1$, $SK_{mn} = h(DK_{mn} \| Cert_{SM_m} \| Cert_{SP_n} \| TC_{SM_m})$, and $X_2 = h(ID_{SP_n} \| PID_{TA} \| X_1 \| Cert_{SP_n}.PP \| R_{SM_m}^1)$. Next, SP_n generate a new random number r_n^{new} and compute $TID_{SM_m}^{new} = E_{S_{SP_j}}(ID_{SM_m} \| TC_{SM_m} \| r_n^{new})$ and $X_3 = (TID_{SM_m}^{new} \| r_n^{new} \| ID_{SP_n}) \oplus h(TID_{SM_m} \| TC_{SM_m})$. Next, SP_n forward authentication response message $M_2 = \{X_1, X_2, X_3, Cert_{SP_n}, C_{SM_m}^1\}$ via public channel.

ACP 4: SM_m after receiving the message M_2, first retrieve the related $R_{SM_m}^1$, based on $C_{SM_m}^1$, and then check certificate if $Cert_{SP_n}.PP = Pub_{SP_n} + h(PID_{TA} \| Pub_{SP_n}).Pub_{TA}$?. Next, SM_m compute $(TID_{SM_m}^{new} \| r_n^{new} \| ID_{SP_n}) = X_3 \oplus h(TID_{SM_m} \| TC_{SM_m})$ and check whether $X_2' = h(ID_{SP_n} \| PID_{TA} \| X_1 \| Cert_{SP_n}.PP \| R_{SM_m}^1)$ or not. If the verification fails, SM_m terminates the connection. Otherwise, SM_m computes $DK_{nm} = h(r_m \| TID_{SM_m}).X_1$ and $SK_{nm} = h(DK_{nm} \| Cert_{SM_m} \| Cert_{SP_n} \| TC_{SM_m})$. It updates TID_{SM_m} with new $TID_{SM_m}^{new}$ in its database. In the end, SM_m shares the common session key $SK_{mn} (= SK_{nm})$ for secure communication in the future.

3.4. Block formation and addition phase

The access control process described in Section 3-C, it is important to note that an SP_n and its associated SM_m generate a session key $SK_{mn} (= SK_{nm})$. Now applying this SK_{mn}, SP_n will gather all the encrypted data of the form (ET_i, $Sign_{ET_i}$) from its SM_m, where $Sign_{ET_i}$ is the $ECDSA$ signature generation algorithm [24]. So, SP_n decrypts all the encrypted data by applying the same SK_{mn}. Next, the encryption transaction is then created by SP_n by encrypting the decrypted data applying its own (Pub_{SP_n})

Table 3
Formation of a block $Block_i$ on encrypted transactions by SP_n in proposed scheme.

Block header	
Block Version (BV_i)	Unique number of block version
Previous Block Hash (PBH_i)	Hash value of the previous block $Block_{i-1}$
Merkle Tree Root ($MTRoot_i$)	$MTRoot_i$ on the encrypted transactions
Timestamp (TS_i)	Creation time of a block
Owner (O_i) of $Block_i$	Service provider (SP_n)
Public key of owner O_i	Pub_{SP_n}
Block payload (Encrypted Transactions)	
Encrypted Transactions ET_i	$EP_{Pub_{SP_n}}$ (ET_i, $Sign_{ET_i}$) ($i = 1, 2, \dots, n_z$)
Current Block Hash (CBH_i)	Hash value of the current block $Block_i$
$ECDSA$ signature on (CBH_i)	$Sign_{CBH_i}$

as $EP_{Pub_{SP_n}}$ $[ET_i, Sign_{ET_i}]$, and then putting it into the global transaction pool (GT_P), which will be accessible in $P2P$ assisted SP system. Suppose that GT_P is completely filled by the list of n_z encryption transaction, say $\{EP_{Pub_{SP_n}}$ $(ET_1, Sign_{ET_1}), EP_{Pub_{SP_n}}$ $(ET_2, Sign_{ET_2}), \dots, EP_{Pub_{SP_n}}$ $(ET_{n_z}, Sign_{ET_{n_z}})\}$. Now, when GT_P reaches to transaction threshold (T_{tresh}) the less number of the transactions (n_z) to be kept in $Block_i$, i.e., $T_{tresh} = n_z$, a leader (LE) will be chosen by the algorithm from $P2P$ assisted SP network applying the similar approach mentioned in [16]. Further, LE will create the block $Block_i$ as discussed in Table 3.

Once $Block_i$ is created by LE, a voting-based consensus applying $PBFT$ algorithm [25] will be accomplished in proposed scheme for block addition in the blockchain. Firstly, LE forwards $Block_i$ along with different random secrets to other SP_n in the $P2P$ assisted SP network to verify the block for consensus purpose. If SP_n successfully authenticates the block with an existing GT_P, it securely sends its verification status (V_{Status}) to LE. The authentic block verification status V_{Status} is part of the global commitment message pool (GCM_p), which is kept up to date by LE and accessible to all peer nodes. Next, LE keeps the global commitment message pool (GCM_p), which consists of authentic block verification status V_{Status} and is available to all P2P nodes. Next, the leader LE increments its counter (VB_{count}) depending on valid V_{Status}, where VB_{count} is the amount of authentic votes in (GCM_p), initially is set to 0. $VB_{count} > 2m_f + 1$, LE forwards commit messages to all responded SP_n and add the $Block_i$ in blockchain. Meantime, all other peer nodes in $P2P$ assisted SP system will add the block to their distributed ledger. Note that any SP_n and other participants involved in the system can authenticate the added block. However, using the public key-based ECC decryption algorithm, only SP_n (LE), who created $Block_i$ is able to decrypt the encrypted transactions contained in that block. SP_n owns the identical secret key K_{SP_n} that corresponds to the open/public key $Pub_{SP_n} = K_{SP_n}.PP$.

3.5. Dynamic node update phase

Sometimes, some SP_s may become defective nodes, or an attacker (AK) may physically compromise some SM_s. As a result, adding new SM_s or SP_s to the existing IoT smart power grid network becomes essential. Suppose a new smart meter (SM_m^{new}) demands to be installed under the existing SP_n, and a new service providers (SP_n^{new}) needs to be installed in existing peer to peer assisted SP system. To accomplish this task, TA selects for SM_m^{new} a unique identity $ID_{SM_m}^{new}$, random number RN_m^{new}, generates challenge and response pairs ($C_{SM_m}^{new}, R_{SM_m}^{new}$), random secret key $K_{SM_m}^{new}$, and computes its public/open key $Pub_{SM_m}^{new} = K_{SM_m}^{new}.PP$. Further, TA computes $TC_{SM_m}^{new} = h(ID_{SM_m}^{new} \| mk_{TA} \| K_{SM_m}^{new} \| RN_m^{new} \| C_{SM_m}^{new} \| R_{SM_m}^{new}), TID_{SM_m}^{new} = E_{S_{SP_j}}(ID_{SM_m}^{new} \| TC_{SM_m}^{new} \| RN_m^{new})$, and $Cert_{SM_m}^{new} = K_{SM_m}^{new} + h(PID_{TA} \| Pub_{TA} \| ID_{SM_m}^{new} \| C_{SM_m}^{new} \| R_{SM_m}^{new}) * mk_{TA}$ $(mod\ s)$. Next, TA loads all the credentials $\{TID_{SM_m}^{new}, ID_{SM_m}^{new}, TC_{SM_m}^{new}, PID_{TA}, Cert_{SM_m}^{new}, (C_{SM_m}^{new}, R_{SM_m}^{new})\}$ in $SM_m^{new'}s$ memory. Next, TA declares the public key $Pub_{SM_m}^{new}$ as public. Moreover, TA also keeps the information $TID_{SM_m}^{new}$ in the database of SP_n. Similarly, (SP_n^{new}) will be registered by TA as discussed in Section 3.2.2. before its formation.

4. Security analysis

This section provides both formal security and informal security analysis to prove that our scheme guards against all well known attacks, which are elaborated below:

4.1. Formal analysis

To demonstrate the security of the proposed scheme against an AK in deriving the session key among a smart meter (SM_m) and an (SP_n), we use the well-known ROR oracle model [26]. To achieve this, we first briefly introduce the ROR model with the semantic security idea, then explain the session key integrity of proposed scheme in Theorem 1. The ROR model is related to the following different components.

(i) ***Participants***: In the access control procedure, there are two participants involved: a SM_m and a SP_n. The TA is only involved in the registration as well as in dynamic node addition phases of the process, therefore it is not present during the access control phase. We demonstrate the I_1, I_2 instances of SM_m and SP_n using $\Pi_{SM_m}^{I_1}$, $\Pi_{SP_n}^{I_2}$ respectively. These are known as random oracles.

Smart Meter (SM_m)	**Service Provider** (SP_n)
Generate random secret $r_m \in Z_s^*$ Compute $Y_1 = h(r_m \parallel ID_{SM_m}).\ PP$ $Y_2 = h(ID_{SM_m} \parallel PID_{TA} \parallel Y_1 \parallel Cert_{SM_m}.PP)$	
$\xrightarrow{M_1=\{Y_1, Y_2, TID_{SM_m}, Cert_{SM_m}\}}$ (Public Channel)	
	$(ID_{SM_m} \parallel TC_{SM_m} \parallel RN_m) = D_{S_{SP_j}}(TID_{SM_m})$ Check certificate: if $Cert_{SM_m}.PP = Pub_{SM_m} + h(PID_{TA} \parallel Pub_{TA} \parallel ID_{SM_m} \parallel C_{SM_m} \parallel R_{SM_m}).Pub_{TA}$? Verify if $Y_2' = h(ID_{SM_m} \parallel PID_{TA} \parallel Y_1 \parallel Cert_{SM_m}.PP)$ Generate random secret $r_n \in Z_s^*$ Chooses a pair of $(C^1_{SM_m}, R^1_{SM_m})$ Compute $X_1 = h(r_n \parallel TID_{SP_n})\ .\ PP$ $DK_{mn} = h(r_n \parallel TID_{SP_n}).Y_1$ $SK_{mn} = h(DK_{mn} \parallel Cert_{SM_m} \parallel Cert_{SP_n} \parallel TC_{SM_m})$ $X_2 = h(ID_{SP_n} \parallel PID_{TA} \parallel X_1 \parallel Cert_{SP_n}.PP \parallel R^1_{SM_m})$ Generate new random number r_n^{new} Calculate $TID^{new}_{SM_m} = E_{S_{SP_j}}(ID_{SM_m} \parallel TC_{SM_m} \parallel r_n^{new})$ $X_3 = (TID^{new}_{SM_m} \parallel r_n^{new} \parallel ID_{SP_n}) \oplus h(TID_{SM_m} \parallel TC_{SM_m})$
$\xleftarrow{M_2=\{X_1, X_2, X_3, Cert_{SP_n}, C^1_{SM_m}\}}$ (Public Channel)	
Retrieve the related $R^1_{SM_m}$, on the basis of $C^1_{SM_m}$ Verify Certificate: if $Cert_{SP_n}.PP = Pub_{SP_n} + h(PID_{TA} \parallel Pub_{SP_n}).Pub_{TA}$? $(TID^{new}_{SM_m} \parallel r_n^{new} \parallel ID_{SP_n}) = X_3 \oplus h(TID_{SM_m} \parallel TC_{SM_m})$ Verify if $X_2' = h(ID_{SP_n} \parallel PID_{TA} \parallel X_1 \parallel Cert_{SP_n}.PP \parallel R^1_{SM_m})$ $DK_{nm} = h(r_m \parallel TID_{SM_m}).X_1$ $SK_{nm} = h(DK_{nm} \parallel Cert_{SM_m} \parallel Cert_{SP_n} \parallel TC_{SM_m})$ Update TID_{SM_m} with new $TID^{new}_{SM_m}$.	Update TID_{SM_m} with new $TID^{new}_{SM_m}$
Both SM_m and SP_n store the shared common session key $SK_{mn}(= SK_{nm})$.	

Fig. 3. Access control phase of proposed scheme.

(ii) ***Accepted State***: The accepted state of an instance Π^I will be entered after it reaches the accept state and receives its final valid scheme message. This will cause the instance to enter its final state. Afterward, it is possible to sort in sequence all of the messages that have been sent and received, and this results in what is known as the session identification sid of Π^I for the current session.

(iii) ***Partnering***: The two instances, Π^{I_1}, and Π^{I_2} are partners if all three of the following conditions are met:

- Π^{I_1}, and Π^{I_2} must be in accept state.
- Π^{I_1}, and Π^{I_2} needs to mutually authenticate each other and must have the same sid.
- Π^{I_1}, and Π^{I_2} are the mutual partners of each other.

(iv) ***Freshness***: If the Reveal Π^I query does not reveal the generated session key $SK_{mn}(= SK_{nm})$ shared among SM_m and SP_n to AK, then we say that the instance $\Pi^{I_1}_{SM_m}$ or $\Pi^{I_2}_{SP_n}$ for SM_m or SP_n is fresh. Further, we elaborate various queries that AK can determined.

- ***Execute(*** $\Pi^{I_1}_{SM_m}$ ***or*** $\Pi^{I_2}_{SP_n}$***)***: Through the use of execute query, the attacker is able to overhear the conversation between SM_m and SP_n.
- ***CorruptSmartMeter (*** $\Pi^{I_1}_{SM_m}$***)***: In this query, AK is allowed to extract stored secret credentials of lost or stolen SM_m.
- ***Reveal (*** Π^I***)***: Using this reveal query, the session key $SK_{mn}(= SK_{nm})$ shared among Π^I and its related entities revealed to AK.
- ***Test (*** Π^I***)***: This query provides a flipped unbiased coin for random outcomes, as Cc and allows AK to get Π^I for checking $SK_{mn}(= SK_{nm})$.

Now, before we proceed on to proving Theorem 1, let us first describe the semantic security of our proposed scheme in Definition 1.

Definition 1. If we denote $Adv^{DBC}_{AK}(T_{pol})$ as a benefit that an AK run in a polynomial time (T_{pol}) has in busting the security and privacy of the presented scheme for calculating $SK_{mn} = SK_{nm}$ between SM_m and SP_n, then $Adv^{DBC}_{AK}(T_{pol}) = |2P_r[Cc' = Cc] - 1|$. Furthermore, Cc' and Cc are guessed and correct bits, respectively.

Theorem 1. *Let us assume that there is a AK that operates T_{pol} to compute $SK_{mn} = SK_{nm}$ that is formed between SM_m and SP_n in the presented scheme. If q_{hsh}, $Adv^{DBC}_{AK}(T_{pol})$, and $|Hash|$ each stand for the number of hash queries, the advantage of breaking the elliptic curve decisional Diffie–Hellman problem (ECDDHP), and the range space of a one-way collision-resistant hash function h(.). Then, $Adv^{DBC}_{AK}(T_{pol}) \leq \frac{q^2_{hsh}}{|Hash|} + 2Adv^{DBC}_{AK}(T_{pol})$.*

Proof. To demonstrate that this theorem is valid, we will proceed in the same manner as in [1]. There are three games, used as $Game^{AK}_i$ for AK, where i = 1, 2, and 3, and we will define $Suc^{AK}_{Game_i}$ as the occurrence of AK successfully guessing the random bit Cc

in the $Game_i^{AK}$. The advantage that AK has in winning the $Game_i^{AK}$ in DBC is denoted by the notation $Adv^{DBC}_{AK,Game_i} = P_r[Suc^{AK}_{Game_i}]$. The following is a detailed breakdown of each individual game.

- **Game$_1^{AK}$**: Under the ROR model, the AK performed possible attack against the proposed DBC. These are always corresponds to the start $Game_1^{AK}$. Before starting $Game_1^{AK}$, the value of the bit Cc must first be chosen at random by AK. The following are the results of applying the semantic security stated in Definition 1:

$$Adv^{DBC}_{AK}(T_{pol}) = |2.Adv^{DBC}_{AK,Game_1} - 1|. \tag{1}$$

- **Game$_2^{AK}$**: This game is same as eavesdropping game, in which the AK uses the defined Execute query. With the help of this query, AK will be able to read all of the messages that are being transmitted as $M_1 = \{Y_1, Y_2, TID_{SM_m}, Cert_{SM_m}\}$ and $M_2 = \{X_1, X_2, X_3, Cert_{SP_n}, C^1_{SM_m}\}$ and tries to get the session key. Next, AK need to perform Reveal and Test queries to determine if the derived session key is a correct one or simply a random key. This will allow AK to determine whether the derived session key is correct or not. It is important to note that $SK_{nm} = h(DK_{nm}\|Cert_{SM_m}\|Cert_{SP_n}\|TC_{SM_m})$ $= h(DK_{mn}\|Cert_{SM_m}\|Cert_{SP_n}\|TC_{SM_m}) = SK_{mn}$, where $DK_{nm} = h(r_m\|TID_{SM_m}).X_1 = (h(r_m\|TID_{SM_m}) * h(r_n\|TID_{SP_n})).PP =$ $h(r_n\|TID_{SP_n}).(h(r_m\|TID_{SM_m}).PP) = h(r_n\|TID_{SP_n}).Y_1 = DK_{mn}$. Now the success probability of obtaining the session key $SK_{mn} = SK_{nm}$ will no longer be increased by message interception m_n (where n = 1, 2). This is because all of the temporary and long term private credentials are secured by h(.). As a result of eavesdropping assault, it is no longer possible to differentiate between the games $Game_1^{AK}$ and $Game_2^{AK}$. Therefore, the following is what we find:

$$Adv^{DBC}_{AK,Game_2} = Adv^{DBC}_{AK,Game_1} \tag{2}$$

- **Game$_3^{AK}$**: This game will be analogous to an active attack, and it will feature simulations of CorruptSmartMeter and Hash queries, in addition to testing players' ability to solve ECDDHP. In order to determine the session key $SK_{mn} = SK_{nm}$, the AK must first determine the value of $DK_{mn} = DK_{nm}$, where $DK_{mn} = h(r_n\|TID_{SP_n}).Y_1$ and $DK_{nm} = h(r_m\|TID_{SM_m}).X_1$. Suppose that AK already possesses the intercepted communications M_n (n = 1, 2), he is aware of the values of $Y_1 =$ $h(r_m\|ID_{SM_m})$. PP and $X_1 = h(r_n\|TID_{SP_n})$. PP. Since $DK_{nm} = h(r_m\|TID_{SM_m}).X_1 = (h(r_m\|TID_{SM_m}) * h(r_n\|TID_{SP_n})).PP =$ $h(r_n\|TID_{SP_n}).(h(r_m\|TID_{SM_m}).PP) = h(r_n\|TID_{SP_n}).Y_1 = DK_{mn}$, the AK has to figure out how to solve the computational ECDDHP to get $DK_{mn} = DK_{nm}$. In addition, additional secrets, such as TC_{SM_m} and TC_{SP_n}, are embedded within the h(.) hash function. Additionally, AK can have the credentials $\{TID_{SM_m}, TC_{SM_m}, PID_{TA}, Cert_{SM_m}\}$ if the CorruptSmartMeter query is used. AK will then be able to extract the session key $SK_{mn} = SK_{nm}$ since they will have knowledge of additional secrets such as r_m, r_n, TID_{SP_n}, and TC_{SP_n}. We find that the games $Game_2^{AK}$ and $Game_3^{AK}$ are indistinguishable from one another if we do not have a simulation of the CorruptSmartMeter and Hash queries, and we also find that ECDDHP is not a difficult problem. With the use of the outcomes of the birthday paradox for locating the hash collision, as well as the benefits of solving ECDDHP, we are able to derive the following relation:

$$|Adv^{DBC}_{AK,Game_2} - Adv^{DBC}_{AK,Game_3}| \leq \frac{q^2_{hsh}}{2|Hash|} + Adv^{DBC}_{AK}(T_{pol}) \tag{3}$$

It is important to point out that AK is the one who generates all of the queries, and all that is required for AK to win the game $Game_2^{AK}$ is for him to make a few accurate guesses. Because of this, we have:

$$Adv^{DBC}_{AK,Game_3} = \frac{1}{2} \tag{4}$$

Now Eq. (1) gives:

$$\frac{1}{2}Adv^{DBC}_{AK}(T_{pol}) = |Adv^{DBC}_{AK,Game_1} - \frac{1}{2}| \tag{5}$$

Using triangular inequality of Eqs. (2)–(4), the results of Eq. (5) is as

$$\begin{aligned}\frac{1}{2}Adv^{DBC}_{AK}(T_{pol}) &= |Adv^{DBC}_{AK,Game_1} - Adv^{DBC}_{AK,Game_3}| \\ &= |Adv^{DBC}_{AK,Game_2} - Adv^{DBC}_{AK,Game_3}| \\ &\leq \frac{q^2_{hsh}}{2|Hash|} + Adv^{DBC}_{AK}(T_{pol}).\end{aligned} \tag{6}$$

Finally, multiplying both sides of (6) by 2, we get the final result as:

$$Adv^{DBC}_{AK}(T_{pol}) \leq \frac{q^2_{hsh}}{|Hash|} + 2Adv^{ECDDHP}_{AK}(T_{pol}) \quad \square$$

4.2. Informal analysis

The informal security analysis demonstrates that the proposed scheme prevents different threats. The details of the prevention of the threats are explained in the subsections:

4.2.1. Smart meter impersonation attack

Suppose an attacker (AK) behaves as a register SM_m and wants to communicate with SP_n with the message $M_1 = \{Y_1, Y_2, TID_{SM_m}, Cert_{SM_m}\}$. In that case, AK can choose a random secret r_m to calculate $Y_1 = h(r_m \| ID_{SM_m})$. PP and $Y_2 = h(ID_{SM_m} \| PID_{TA} \| Y_1 \| Cert_{SM_m}.PP)$. However, AK needs to know ID_{SM_m}, which is the real identity of SM_m. Therefore, AK cannot find the ID_{SM_m} of an SM_m because ID_{SM_m} is not forwards in a plain message across the network. Therefore, our scheme is secure against smart meter impersonation attack because AK unable to scam the correct authentication request message.

4.2.2. Service provider impersonation attack

Suppose an AK tries to impersonate a legitimate SP_n, AK should scam the response message. However, it is challenging for AK to forge the response message because AK unable to know the PUF challenge/response pairs ($C^1_{SM_m}, R^1_{SM_m}$). Therefore, our scheme is secure against service provider impersonation attack because AK unable to scam the correct response message.

4.2.3. Smart meter physical capture attack

Suppose that SM_m are physically captured by AK and AK then extracts all the secret parameters $\{TID_{SM_m}, TC_{SM_m}, PID_{TA}, ID_{SM_m}, Cert_{SM_m}, (C_{SM_m}, R_{SM_m})\}$ in the memory. However, AK does not compute the login request message without knowing the secret credential $Cert_{SM_m}$, TID_{SM_m}, ID_{SM_m}, and PID_{TA} of smart meter. Furthermore, there are secure, independent, and distinct for all deployed SM_m because PUF challenge/response pairs (C_{SM_m}, R_{SM_m}) are randomly produced. Therefore, AK is unable to calculate the login request message. The outputs of the PUF method are dependent on the inherent physical variations in the IC chip. Consequently, our scheme prevents physical capture attack.

4.2.4. Man-in-the-middle attack

Suppose an AK may capture the login request message $M_1 = \{Y_1, Y_2, TID_{SM_m}, Cert_{SM_m}\}$ from public/insecure channel. Further, AK generate another authentic message M'_1 on the fly so that SP_n as the recipient cannot find it as an modified one. However, AK cannot produce the values of Y_1 and Y_2 to generate the legal message M'_1 due to preloaded private credentials ID_{SM_m} and TID_{SM_m}. Similarly, AK also unable to produce valid response message for the intercepted message $M_2 = \{X_1, X_2, X_3, Cert_{SP_n}, C^1_{SM_m}\}$ due to preloaded private credentials TID_{SP_n} and S_{SP_j}. Further, the common session key (SK_{mn}) is required for this purpose. Thus, our scheme prevents man-in-the-middle attack.

4.2.5. Ephemeral secret leakage attack

In access control phase, SP_n computes SK_{mn} shared with SM_m as $SK_{mn} = h(DK_{mn} \| Cert_{SM_m} \| \; Cert_{SP_n} \| TC_{SM_m})$, where $DK_{mn} = h(r_n \| TID_{SP_n}).Y_1$, and $X_1 = h(r_n \| TID_{SP_n})$. PP. However, SM_m also computes SK_{nm} shared with SP_n as $SK_{nm} = h(DK_{nm} \| Cert_{SM_m} \| Cert_{SP_n} \| TC_{SM_m})$, where $DK_{nm} = h(r_m \| TID_{SM_m}).X_1$, and $Y_1 = h(r_m \| ID_{SM_m})$. PP. Since $DK_{nm} = DK_{mn}$, both SM_m and SP_n share the common $SK_{mn} (= SK_{nm})$, which is also proved in Theorem 1. SK_{mn} is the combination of both "long term secrets" such as random secrets and "short term secrets" such as temporary identities and various secret parameters. Additionally, the computation of shared session keys among different SM_m and SP_n over various sessions produces unique session keys among SM_m and SP_n due to the use of random secrets. Even if a session key is exposed for a particular session, the usage of short and long-term secrets prevents the calculation of session keys over other sessions. Consequently, our scheme prevents ephemeral secret leakage attack.

4.2.6. Prevents anonymity and untraceability

Assume an AK snoop the communication messages M_1 and M_2. Since each message does not have the real identity ID_{SM_m} of SM_m and the real identity ID_{SP_n} of SP_n directly. Therefore, AK unable to find the real identity of SM_m and SP_n. As a result, "anonymity" of both SP_n and SM_m is preserved in proposed scheme. Due to the use of random numbers, the parameters in the different messages M_1 and M_2 are all completely dynamic. In the access control phase, no two key exchange sessions are same. As a result, AK is unable to determine whether or not the messages delivered and received by entities over two consecutive sessions belong to the same entity. Hence, our scheme preserves "untraceability as well".

4.2.7. No online TA

In our scheme, TA excludes during the access control phase. All data are kept on the blockchain. The revocation and verification are done by requesting blockchain transactions.

4.2.8. Prevents stolen verifier attack

Since the blockchain ledger contains all the verification data, the information can get by invoking transactions. Therefore, there is no need to keep a verifier table maintained. Hence, our scheme prevents stolen verifier attack.

4.2.9. Provides mutual authentication

In our scheme, all participants perform mutual authentication. After receiving the request message $M_1 = \{Y_1, Y_2, TID_{SM_m}, Cert_{SM_m}\}$, SP_n verify whether $Y'_2 = h(ID_{SM_m} \; \| \; PID_{TA} \; \| \; Y_1 \; \| \; Cert_{SM_m}.PP)$ or not. If the authentication is true, SP_n is considered SM_m as legitimate participant. After receiving the request message $M_2 = \{X_1, X_2, X_3, Cert_{SP_n}, C^1_{SM_m}\}$, SM_m verify whether $X'_2 = h(ID_{SP_n} \| PID_{TA} \| X_1 \| Cert_{SP_n}.PP \| R^1_{SM_m})$ or not. If the authentication is true, SM_m is considered SP_n as legitimate participant. As a result, all participants are mutually authenticated, and AK cannot generate authentication responses or request messages.

Table 4
Specifications.

Items	Arduino device	System
Platform	–	Ubuntu
IDE	Arduino IDE	PyCharm
Processor	Microcontroller : (ATmega 328)	intel Core i7
RAM	2 kB (ATmega 328)	16 (GB)
Clock speed	16 (MHz)	2.9 (GHz)

Table 5
Execution time cryptographic operations.

Operations	Execution time	
	Arduino device	System
T_h	1.870 ms	0.00948 ms
T_{pm}	0.765 ms	0.00532 ms
T_{pa}	0.645 ms	0.00468 ms
$T_{e/d}$	0.880 ms	0.00658 ms

4.2.10. Prevents desynchronization attack

The desynchronization attack can launch when the participants involved in the access control phase update a few credentials during the execution of every access control phase to assure anonymous communication. AK has the ability to block any message in the network, or there may be few packet losses. In these conditions, SM_m cannot obtain new $TID^{new}_{SM_m}$. However, in our scheme, SM_m is a legal smart meter whose unique identity is already kept in SP_n. As a result, the legal SM_m can continue the session with SP_n by using SM_m's stored identity TID_{SM_m}. Thus, our scheme prevents desynchronization attack.

5. Performance analysis

This section explains the comparison of the performance between design scheme and related schemes, including Bera et al. [8], Badshah et al. [27], Park et al. [28], and Tomar et al. [29].

5.1. Experimental setup

The implementation of our scheme and related schemes involves two participants, including (i) Smart Meter (SM_m) and (ii) Service Provider (SP_n). As we already know, registration phase is a one-time activity, whereas the dynamic node update phase is carried out at the request of SM_m. Therefore, we have omitted these two processes. When calculating the associated communication and computation costs, we only consider the messages exchanged during the access control process.

We denote T_h, T_{pm}, T_{pa}, and $T_{e/d}$ as time required for computing SHA-256 hash function, point multiplication, point addition, symmetric encryption/decryption, respectively. Each experiment for a cryptographic primitive is run 100 times. All cryptographic functions in SP_n are implemented on the system. All of these outcomes are obtained after implementing the related and proposed schemes in a system environment with specifications such as Ubuntu, 16 GB RAM, clock speed 2.9 GHz, and the PyCharm integrated development environment. We used the PyCrypto library in Python language to exploit the experimental results of these operations. We get $T_h \approx 0.00948$ ms, $T_{pm} \approx 0.00532$ ms, $T_{pa} \approx 0.00468$ ms, and $T_{e/d} \approx 0.00658$ ms. Moreover, all cryptographic functions in SM_m are implemented on the Arduino device. All of these outcomes are obtained after implementing the related and proposed schemes on the Arduino device with specifications such as 2 kB RAM (ATmega 328), microcontroller processor, the Arduino IDE, and a 16 MHz clock speed. Then, we get $T_h \approx 1.870$ ms, $T_{pm} \approx 0.765$ ms, $T_{pa} \approx 0.645$ ms, and $T_{e/d} \approx 0.880$ ms. The Table 4 provides information regarding the components of a system and an Arduino device. In addition, the Table 5 demonstrates the amount of time required for each cryptographic primitive, such as the hash function, point addition, and point multiplication, to complete their operations within the implementation environment.

5.2. Computation cost

The entire number of bits exchanged by the entities to complete the authentication procedure is shown in the communication cost. For the analysis of computational cost of proposed and related schemes [8,27–29], we denote T_h, T_{pm}, T_{pa}, and $T_{e/d}$ as time require for computing SHA-256 hash function, point multiplication, point addition, symmetric encryption/decryption, respectively. The findings presented in Table 5 are utilized for a variety of cryptographic primitives. Every side of the computation costs is calculated, i.e., SM_m and SP_n. In proposed scheme the required computational cost for SM_m and SP_n are 16.795 ms and 0.10548 ms, respectively. So, our scheme's overall computation cost is 16.900 ms. The computation cost of all related schemes is also computed by applying the same method and depicted in Table 6.

Table 6
Computation costs comparison.

Schemes	SM_m	SP_n	Total cost
Proposed	$7T_h + 4T_{pm} + 1T_{pa} \approx 16.795$ ms	$7T_h + 4T_{pm} + 1T_{pa} + 2T_{e/d} \approx 0.10548$ ms	16.900 ms
Bera et al. [8]	$15T_h + 2T_{pm} + 1T_{pa} \approx 30.225$ ms	$15T_h + 2T_{pm} + 1T_{pa} \approx 0.15752$ ms	30.3825 ms
Badshah et al. [27]	$8T_h + 1T_{pm} \approx 15.725$ ms	$8T_h 1T_{e/d} \approx 0.08242$ ms	15.807 ms
Park et al. [28]	$7T_h + 4T_{pm} + 1T_{pa} \approx 16.49$ ms	$7T_h + 4T_{pm} + 1T_{pa} + 2T_{e/d} + 1T_{pa} \approx 0.077$ ms	16.567 ms
Tomar et al. [29]	$8T_h + 4T_{pm} + 1T_{pa} \approx 17.135$ ms	$15T_h + 4T_{pm} + 6T_{pa} \approx 0.1915$ ms	17.3265 ms

Table 7
Communication costs comparison.

Schemes	SM_m	SP_n	Total cost
Proposed	928 bits	800 bits	1728 bits
Bera et al. [8]	1760 bits	1856 bits	3616 bits
Badshah et al. [27]	928 bits	800 bits	1728 bits
Park et al. [28]	992 bits	832 bits	1824 bits
Tomar et al. [29]	832 bits	992 bits	1824 bits

Table 8
Comparison on security features.

Schemes	S[1]	S[2]	S[3]	S[4]	S[5]	S[6]	S[7]	S[8]	S[9]	S[10]
Proposed	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Bera et al. [8]	✓	✓	✗	✓	✓	✓	N/A	N/A	N/A	N/A
Badshah et al. [27]	✗	✓	✓	✗	✗	✓	N/A	N/A	N/A	N/A
Park et al. [28]	✓	✓	N/A	✓	✗	✓	N/A	N/A	N/A	N/A
Tomar et al. [29]	✗	✗	N/A	✓	N/A	✓	✓	✓	✓	N/A

S[1]: Smart Meter Impersonation Attack; S[2]: Service Provider Impersonation Attack; S[3] : Smart Meter Physical Capture Attack;
S[4] : Man-in-the-Middle Attack; S[5]: Ephemeral Secret Leakage Attack; S[6]: Preserves both Anonymity and Untraceability;
S[7]: No Online TA; S[8]: Prevents Stolen Verifier Attack; S[9]: Provides Mutual Authentication; S[10]: Prevents Desynchronization Attack;
✓: Offers; ✗: Does not offer; N/A: Not Applicable.

5.3. Communication cost

The total number of bits sent between the entities to accomplish the authentication procedure represents the communication cost. For the analysis of communication cost, we consider numerous parameters and their sizes in bits such as point addition, point multiplication, random number, identity each using 160 bits, output of the hash function, private/public keys require 256 bits, the PUF challenge require 32 bits, and symmetric encryption/decryption requires 128 bits. In our proposed scheme both entities SM_m and SP_n transfer two messages M_1 and M_2. The SM_m require 928 bits to transfer message M_1 towards SP_n. Whereas, the total number of bit require to send message M_2 are 800 bits. The total costs of our proposed scheme is 1728 bits. The communication cost of all related schemes [8,27–29] is also computed by applying the same method and depicted in Table 7.

5.4. Security features

The analysis of all security features between proposed and related schemes [8,27–29] are represents in Table 8. The proposed scheme ensures all of the significant aspects of security. Whereas, the related schemes does not provide resilience against different security features like impersonation attacks, smart meter physical capture attack, man-in-the-middle attack, ephemeral secret leakage attack etc.

6. Conclusion

This article designs a new access control scheme using blockchain and PUF in an IoT-enabled SG system. Our scheme provides resistance to various assaults using techniques such as security analysis and formal analysis based on ROR model. The performance comparison also shows that the designed scheme prevents physical and cyber threats via PUF. However, it has lower communication and computation costs and provides more security features than related schemes. Consequently, the proposed scheme is appropriate for SG systems because it is more efficient and secure than related schemes.

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.

Data availability

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

References

[1] M. Wazid, A.K. Das, V. Odelu, N. Kumar, W. Susilo, Secure remote user authenticated key establishment protocol for smart home environment, IEEE Trans. Dependable Secure Comput. 17 (2) (2017) 391–406.
[2] R. Alvaro-Hermana, J. Fraile-Ardanuy, P.J. Zufiria, L. Knapen, D. Janssens, Peer to peer energy trading with electric vehicles, IEEE Intell. Transp. Syst. Mag. 8 (3) (2016) 33–44.
[3] M. Andoni, V. Robu, D. Flynn, S. Abram, D. Geach, D. Jenkins, P. McCallum, A. Peacock, Blockchain technology in the energy sector: A systematic review of challenges and opportunities, Renew. Sustain. Energy Rev. 100 (2019) 143–174.
[4] A.S. Musleh, G. Yao, S. Muyeen, Blockchain applications in smart grid–review and frameworks, Ieee Access 7 (2019) 86746–86757.
[5] J. Wang, L. Wu, K.-K.R. Choo, D. He, Blockchain-based anonymous authentication with key management for smart grid edge computing infrastructure, IEEE Trans. Ind. Inform. 16 (3) (2019) 1984–1992.
[6] M. Qi, J. Chen, Two-pass privacy preserving authenticated key agreement scheme for smart grid, IEEE Syst. J. 15 (3) (2020) 3201–3207.
[7] S.A. Chaudhry, H. Alhakami, A. Baz, F. Al-Turjman, Securing demand response management: A certificate-based access control in smart grid edge computing infrastructure, IEEE Access 8 (2020) 101235–101243.
[8] B. Bera, S. Saha, A.K. Das, A.V. Vasilakos, Designing blockchain-based access control protocol in IoT-enabled smart-grid system, IEEE Internet Things J. 8 (7) (2020) 5744–5761.
[9] J.-L. Tsai, N.-W. Lo, Secure anonymous key distribution scheme for smart grid, IEEE Trans. Smart Grid 7 (2) (2015) 906–914.
[10] S.-K. Kim, J.-H. Huh, A study on the improvement of smart grid security performance and blockchain smart grid perspective, Energies 11 (8) (2018) 1973.
[11] H. Wang, X.A. Wang, S. Xiao, J. Liu, Decentralized data outsourcing auditing protocol based on blockchain, J. Ambient Intell. Humaniz. Comput. 12 (2) (2021) 2703–2714.
[12] X.A. Wang, F. Xhafa, J. Ma, Z. Zheng, Controlled secure social cloud data sharing based on a novel identity based proxy re-encryption plus scheme, J. Parallel Distrib. Comput. 130 (2019) 153–165.
[13] V. Odelu, A.K. Das, M. Wazid, M. Conti, Provably secure authenticated key agreement scheme for smart grid, IEEE Trans. Smart Grid 9 (3) (2016) 1900–1910.
[14] K. Mahmood, X. Li, S.A. Chaudhry, H. Naqvi, S. Kumari, A.K. Sangaiah, J.J.P.C. Rodrigues, Pairing based anonymous and secure key agreement protocol for smart grid edge computing infrastructure, Future Gener. Comput. Syst. 88 (2018) 491–500.
[15] K. Gai, Y. Wu, L. Zhu, L. Xu, Y. Zhang, Permissioned blockchain and edge computing empowered privacy-preserving smart grid networks, IEEE Internet Things J. 6 (5) (2019) 7992–8004.
[16] H. Zhang, J. Wang, Y. Ding, Blockchain-based decentralized and secure keyless signature scheme for smart grid, Energy 180 (2019) 955–967.
[17] Y. Zhou, Y. Guan, Z. Zhang, F. Li, A blockchain-based access control scheme for smart grids, in: 2019 International Conference on Networking and Network Applications (NaNA), IEEE, 2019, pp. 368–373.
[18] R. Pappu, B. Recht, J. Taylor, N. Gershenfeld, Physical one-way functions, Science 297 (5589) (2002) 2026–2030.
[19] Y. Yilmaz, L. Aniello, B. Halak, ASSURE: A hardware-based security protocol for internet of things devices, in: Authentication of Embedded Devices, Springer, 2021, pp. 55–87.
[20] J. Delvaux, R. Peeters, D. Gu, I. Verbauwhede, A survey on lightweight entity authentication with strong PUFs, ACM Comput. Surv. 48 (2) (2015) 1–42.
[21] A.C.D. Resende, K. Mochetti, D.F. Aranha, PUF-based mutual multifactor entity and transaction authentication for secure banking, in: Lightweight Cryptography for Security and Privacy, Springer, 2015, pp. 77–96.
[22] D. Dolev, A. Yao, On the security of public key protocols, IEEE Trans. Inform. Theory 29 (2) (1983) 198–208.
[23] R. Canetti, H. Krawczyk, Universally composable notions of key exchange and secure channels, in: International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2002, pp. 337–351.
[24] D. Johnson, A. Menezes, S. Vanstone, The elliptic curve digital signature algorithm (ECDSA), Int. J. Inf. Secur. 1 (1) (2001) 36–63.
[25] M. Castro, B. Liskov, Practical Byzantine fault tolerance and proactive recovery, ACM Trans. Comput. Syst. (TOCS) 20 (4) (2002) 398–461.
[26] M. Abdalla, P.-A. Fouque, D. Pointcheval, Password-based authenticated key exchange in the three-party setting, in: International Workshop on Public Key Cryptography, Springer, 2005, pp. 65–84.
[27] A. Badshah, M. Waqas, G. Abbas, F. Muhammad, Z.H. Abbas, S. Vimal, M. Bilal, LAKE-BSG: Lightweight authenticated key exchange scheme for blockchain-enabled smart grids, Sustain. Energy Technol. Assess. 52 (2022) 102248.
[28] K. Park, J. Lee, A.K. Das, Y. Park, BPPS: Blockchain-enabled privacy-preserving scheme for demand-response management in smart grid environments, IEEE Trans. Dependable Secure Comput. (2022) http://dx.doi.org/10.1109/TDSC.2022.3163138.
[29] A. Tomar, S. Tripathi, Blockchain-assisted authentication and key agreement scheme for fog-based smart grid, Cluster Comput. 25 (1) (2022) 451–468.