RESEARCH ARTICLE

Blockchain assisted signature and certificate based protocol for efficient data protection and transaction management in smart grids

Keyan Abdul-Aziz Mutlaq[1,2], **Vincent Omollo Nyangaresi**[3,4], **Mohd Adib Omar**[1]*, **Zaid Ameen Abduljabbar**[5,6,7,8], **Junchao Ma**[6]*, **Mustafa A. Al Sibahee**[9,10], **Abdulla J. Y. Aldarwish**[5], **Ali Hasan Ali**[11,12]

1 School of Computer Sciences, Universiti Sains Malaysia, USM, Gelugor, Penang, Malaysia, **2** IT and Communications Center, University of Basrah, Basrah, Iraq, **3** Department of Computer Science and Software Engineering, Jaramogi Oginga Odinga University of Science & Technology, Bondo, Kenya, **4** Department of Applied Electronics, Saveetha School of Engineering, SIMATS, Chennai, Tamil Nadu, India, **5** Department of Computer Science, College of Education for Pure Sciences, University of Basrah, Basrah, Iraq, **6** College of Big Data and Internet, Shenzhen Technology University, Shenzhen, China, **7** Shenzhen Institute, Huazhong University of Science and Technology, Shenzhen, China, **8** Department of Business Management, Al-Imam University College, Balad, Iraq, **9** Department of Management and Marketing, College of Industrial Management for Oil and Gas, Basrah University for Oil and Gas, Basrah, Iraq, **10** National Engineering Laboratory for Big Data System Computing Technology, Shenzhen University, Shenzhen, China, **11** Department of Mathematics, College of Education for Pure Sciences, University of Basrah, Basrah, Iraq, **12** Technical Engineering College, Al-Ayen University, Dhi Qar, Iraq

* adib@usm.my (MAO); majunchao@sztu.edu.cn (JM)

Citation: Mutlaq KA-A, Nyangaresi VO, Omar MA, Abduljabbar ZA, Ma J, Al Sibahee MA, et al. (2025) Blockchain assisted signature and certificate based protocol for efficient data protection and transaction management in smart grids. PLoS One 20(5): e0318182. https://doi.org/10.1371/journal.pone.0318182

Editor: Iftekhar Salam, Xiamen University Malaysia, MALAYSIA

Received: May 9, 2024

Accepted: January 11, 2025

Published: May 28, 2025

Data availability statement: The data underlying the results presented in the study are available from (https://figshare.com/articles/dataset/Blockchain_Assisted_Signature_and_Certificate_Based_Protocol_for_Efficient_Data_docx/27194079?file=49690452)

Abstract

Smart grids collect real-time power consumption reports that are then forwarded to the utility service providers over the public communication channels. Compared with the traditional power grids, smart grids integrate information and communication technologies, cyber physical systems, power generation and distribution domains to enhance flexibility, efficiency, transparency and reliability of the electric power systems. However, this integration of numerous heterogeneous technologies and devices increases the attack surface. Therefore, a myriad of security techniques have been introduced based on technologies such as public key cryptosystems, blockchain, bilinear pairing and elliptic curve cryptography. However, majority of these protocols have security challenges while the others incur high complexities. Therefore, they are not ideal for some of the smart grid components such as smart meters which are resource-constrained. In this paper, a protocol that leverages on digital certificates, signatures, elliptic curve cryptography and blockchain is developed. The formal verification using Real-Or-Random (ROR) model shows that the derived session keys are secure. In addition, semantic security analysis shows that it is robust against typical smart grid attacks such as replays, forgery, privileged insider, side-channeling and impersonations. Moreover, the performance evaluation shows that our protocol achieves a 17.19% reduction in the computation complexity and a 46.15% improvement in the supported security and privacy features.

Funding: This research is supported by Natural Science Foundation of Top Talent of SZTU under Grant No.GDRC202137. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

1. Introduction

The traditional power grid network faces numerous challenges regarding flexibility, energy utilization efficiency, safety and environmental protection [1]. This has led to the development of Smart Grids (SGs) which have advanced computing and sensing abilities using a number of sensors and actuators that generate and transmit real-time power related information in a bidirectional manner. In SGs, Information and communication Technologies (ICTs) are deployed to facilitate data exchange between Utility Service Providers (USPs) and the clients. This helps in the control, adjustments and optimization of power consumption based on real-time client needs as reported by the smart meters [1]. As explained in [2], SGs offer seamless integration of ICTs, distribution domains, cyber physical systems as well as power generation domains. Therefore, a typical smart grid comprises of automation technologies, power generation, distribution, transmission as well as advanced sensing and control components. These technologies help boost efficiency, reliability, transparency and flexibility of the electric power systems [3,4]. In addition, the advanced metering infrastructure, self-healing and demand response of the SGs result in optimum utilization of power stations as well as better control of consumer costs. The Smart Meter (SM) is the main component in the SGs and can generate real-time power consumption reports which are then periodically transmitted to the USPs. This normally happens after every 15 minutes. The analysis of these reports at the USPs facilitates the prediction of power demands as well as the adjustments in its generation and distribution. In so doing, the SGs reduce costs and energy consumptions while facilitating the integration of renewable energy sources [5].

Although the smart grid brings forth numerous merits in the face of increasing demand for electricity, these systems are vulnerable to numerous attacks. This situation is worsened by the many connected devices in a typical smart grid. Therefore, data confidentiality, integrity and authentication challenges are common in SGs. Authors in [6] attribute this to the heterogeneous connectivity in smart grid networks in which numerous Internet of Things (IoT) devices are incorporated to generate, distribute and transmit data in various systems such as smart meters and Supervisory Control And Data Acquisition (SCADA). In addition, the integration of ICTs in power systems has been noted in [7] to render the grid vulnerable to attacks such as impersonation, replay and Man-in-the Middle (MitM). As explained in [2], Demand Response Management (DRM) is crucial for improved reliability and efficiency smart grid ecosystem. This is normally enabled by the frequent data transfer between the USPs and smart meters. Unfortunately, these data transfers are prone to many threats such as tampering. This is made worse by the transmission of the data over insecure public channels [8,9]. Therefore, adversaries can intercept the communication process and recover consumer's secret information. Consequently, the balancing of security, privacy, functionality and efficiency is one of the greatest challenges facing the SGs [10]. Authors in [6] explain that if data and device security are not handled properly, they can lead to grid failure.

In addition to security, user privacy leakage is another serious issue that must be solved in SGs. In this context, the adversaries can intercept electricity consumption data and try to associate it with particular users [11]. For instance, the tracking of power consumption patterns by various appliances may help attackers monitor consumer behavior, hobbies, future plans, and lifestyle as well as establish the status of home. This helps the attackers determine when to break-in and commit crimes [12]. It is evident that the large number of heterogeneous devices in the SGs exposes them to a myriad of security and privacy risks [13,14]. To counter these challenges, robust authentication must be executed to ensure that only authorized entities get access to system resources [15]. In addition, session keys must be established to facilitate secure message exchanges among the authenticated entities and uphold their privacy. Unfortunately, majority of the conventional authentication protocols are computationally intensive and hence not suitable for resource-limited smart grid networks [5].

1.1 Contributions

The major contributions of this paper included the following:

- We deploy digital signatures to preserve data integrity by preventing malicious tampering of the transmitted data. Since these signatures are verified at the receiver terminals, forgery and repudiation are thwarted.
- To preserve user privacy, the real identities of the users are never sent over the public channels. In addition, all exchanged messages are enciphered using the negotiated session keys to prevent attackers from eavesdropping the communication channel and obtain user sensitive information such as real-time power consumption reports.
- During transactions management, we validate all blocks before their addition to the blockchain. This makes it difficult for attackers to modify or corrupt the smart grid transactions.
- The performance evaluation is executed to show that the proposed scheme has the least computation complexity and relatively low communication costs. As such, our protocol is able to offer user privacy and real-time power consumption reports protection at improved efficiencies.
- Extensive security analysis is carried out to show that our scheme is provably secure. In addition, it is shown to support mutual authentication, key agreement, key secrecy, anonymity and untraceability. Moreover, it is demonstrated to be robust against typical smart grid attacks such as ephemeral secret leakage, eavesdropping, key escrow, session hijacking, KSSTI, replays, forgery, MitM, privileged insider, physical, side-channeling and impersonations.

The rest of this paper is structured as follows: Section 2 discusses the related works while Section 3 describes the proposed protocol. On the other hand, Section 4 presents the security analysis of the proposed protocol while Section 5 discusses its performance evaluation. Towards the end of this paper, Section 6 describes the conclusions and future works.

1.2 Motivation

The reliance on public channels for data exchanges in smart grids exposes these networks to numerous attacks such as replay, impersonation, forgery and MitM. In addition, the incorporation of ICTs has been shown to introduce numerous security threats to the SGs which can be exploited by adversaries. This might lead to the compromise of terminals such as smart meters which can then transmit falsified information to the grid, resulting in misleading data analytics, forecasting models and adjustments related to DRM. In addition, normal operations of the grid can be interfered with, or wrong power grid operations status can be fed to user terminals. Any successful interruptions on the access from smart meters to the metering system can render the control center unable to obtain real-time consumer load status, leading to power supply interruptions and grid collapse. It is also possible for attackers to monitor consumer load and correlate the time dimensions of diverse household appliances. This results in the determination of user behavioral patterns, personal preferences, activities and preferences, thereby infringing on personal privacy. Although many protocols have been developed to tackle these challenges, many of them are either vulnerable to security and privacy attacks or incur high computation [16] and communication overheads. Due to the hardware, storage capacity and computing power limitations of the smart grid components, they cannot execute highly complex cryptographic operations such as bilinear pairings. There is therefore need to develop an efficient protocol that will help address some of these performance and security issues.

1.3 Adversarial model

We deploy the widely accepted Canetti–Krawczyk (CK) threat model, in which an adversary is thought to have a range of capabilities that can compromise the smart grid communication process. The assumption in this model is that insecure public communication channels are utilized for message exchanges, and the Registration Authority (RA) is sufficiently protected. Therefore, adversary $\mathring{A}$ can eavesdrop the channel, intercept, alter, replay and delete the transmitted data but cannot compromise RA. In addition, $\mathring{A}$ can physically capture the smart grid components such as smart meters and use power analysis attacks to retrieve memory resident secrets. Moreover, session states and keys can be accessed by $\mathring{A}$.

1.4 Key design principles

Smart grid faces numerous security, performance and privacy challenges that must be addressed. Therefore, many protocols have been developed over the recent past. For instance, to preserve privacy and integrity, aggregate signature based schemes have been presented. However, signature verification in these schemes incurs high computation complexities [11]. As explained in [17], majority of the current protocols fail to support flexible key management and conditional anonymity. In addition, most of the current authentication algorithms utilize the Rivest Shamir and Adleman (RSA) for asymmetric encryption of the digital signatures.

- Due to perfections and developments of large integer factorization, the required RSA algorithm key length has increased. Therefore, the encryption and decryption speeds have been reducing, making its hardware implementation difficult [14]. Fortunately, Elliptic Curve Encryption (ECC) algorithm attains the same enciphering strength as RSA but at shorter key lengths. Therefore, it can solve the challenges in RSA algorithm. ECC security is basically hinged on the problem of the Elliptic Curve Discrete Logarithm (ECDL) over the Galois fields. Mathematically, there is no sub-exponential algorithm to the ECDL problem. Since the chips in most of the smart grid devices have limited RAM size and processing power, the digital signatures must be implemented using public key cryptography algorithm with low computation overheads but strong encryption. As explained in [14], a 160-bit ECC algorithm offers the same level of security as the 1024-bit RSA algorithm, while a 210-bit ECC algorithm's security level is equivalent to a 2048-bit RSA algorithm. Therefore, we adopt ECC in the proposed protocol.
- To protect the smart grid terminals, their identities and communication channels security are taken into consideration. We authenticate all terminals using digital certificates to uphold their legitimacy. On the other hand, confidentiality and integrity of the transmitted data in appendix A is protected via the negotiated session keys that are used to encipher the communication channel.
- The smart meters collect real-time data and upload it to the USPs to facilitate DRM, which is critical for the maintenance of smart grid demand and supply stability. Therefore, assigning the USPs an additional responsibility of transactions management increases their data processing pressure, communication load and system response latencies. Therefore, we reduce pressure at the USPs by incorporating the cloud servers and blockchain centers to management the smart meter transactions. This is due to their distributed nature, high storage capacity, computing power and low latencies.

1.5 Security and performance requirements

Mutual authentication: All the network entities must validate their identities before sharing their data.

Session key agreement: Upon successful mutual authentication, the communicating parties should negotiate session keys to encipher the exchanged data.
Key secrecy: An adversary in possession of the current session key should be unable to derive the keys for the previous as well as subsequent communication session.
Anonymity and untraceability: Attackers should be unable to discern the real identities of the smart grid entities upon eavesdropping the channel. In addition, it should be difficult to associate the captured data to any smart grid device or user.
Formal verification: The derived session keys for data enciphering should be mathematically secure.
Attacks resilience: To offer enhanced security and privacy protection, the proposed protocol should thwart conventional smart grid attacks such as ephemeral secret leakage, eavesdropping, key escrow, session hijacking, KSSTI, replays, forgery, MitM, privileged insider, physical, side-channeling and impersonations.
Low complexities: The smart grid supports high number of smart meters whose real-time power consumption data must be processed and responded to. Therefore, the proposed protocol must be lightweight to facilitate efficient processing of the massive smart meter data. This will ensure low network and processing latencies for delay-sensitive smart grid applications.

2. Related work

Efficient, reliable and secure communication procedures are crucial for the smart grid networks [18]. Therefore, many schemes have been put forward over the recent past. For instance, certificate based authentication protocols are presented in [2,19]. In addition, a certificate-based data aggregation technique is introduced in [20]. However, the demand response management scheme in [2] has high computation costs due to numerous elliptic curve point multiplications and has not been analyzed against attacks such as session hijacking, privileged insider and ephemeral secret leakage. Similarly, the security mechanisms in [19,20] have not been evaluated against attacks such as privileged insider and side-channeling. On its part, the scheme in [21] does not support untraceability and protection against attacks such as side-channeling. To offer enhanced security, blockchain-based schemes are developed in [5,22–26]. However, security analyses in [5,24] fail to include attacks such as privileged insider and ephemeral secret leakage. Similarly, security analysis of the scheme in [25] is missing while the privacy preserving technique in [22] is not evaluated against side-channeling and forgery attacks. On the other hand, the protocol in [23] is never analyzed against many attacks such as forgery while the scheme in [26] lacks formal security evaluation.

To support user privacy and data integrity, conventional blind signature based schemes have been developed in [27–29] while ring signature based protocol is introduced in [11]. Similarly, identity-based blind signature protocols are presented in [30–33] while signature and encryption technique is developed in [34]. In addition, group signature-based scheme is introduced in [35] while the protocol in [36] combines blind and group signatures to offer privacy protection. Moreover, certificate-less blind signature technique is developed in [1] while a certificate-based blind signature mechanism is presented in [37]. Although these signature-based schemes solve user data integrity issues, they are susceptible to quantum attacks [33]. In addition, most of these signature schemes have numerous security issues and some of them are inefficient due to bilinear pairing operations [1,38]. For instance, the scheme in [11] cannot offer key secrecy, untraceability and lacks formal verification. As explained in [11], group signature is facilitated by group administrator and hence may trace the identity of the group members. On its part, the scheme in [33] lacks semantic security analysis while the protocol in [1] has not been evaluated against attacks such as privileged insider,side-channeling and MitM. On the other hand, the protocol in [32] fails to offer support for trust measurement. Due to its requirement for the maintenance of the certificate revocation list, this approach incurs extra overheads. Although the scheme in [39] can address this problem, it relies on a third party for secure session establishment between smart grid devices.

To protect against various insider and outside attacks, an ECC-based scheme is developed in [40]. However, this scheme incurs extensive communication and computation overheads. To preserve privacy during data sharing, secure aggregation techniques are presented in [41–43]. Although the scheme in [41] does not depend on trusted third parties and can prevent collusion attacks, its fault tolerance is low [10] and the computation costs [44] at the smart meter side is high [3]. Similarly, the protocol in [42] can prevent collusion attacks but at high communication overheads and complicated key management procedures [10]. Although the Chebyshev chaotic maps based scheme in [45] addresses this problem, it lacks evaluation against forgery and session hijacking attacks. On its part, the protocol in [43] offers privacy protection and flexible user management at high computation costs due to frequent key updates for each time slot [3]. To preserve anonymity during the authentication process, bilinear pairing based security techniques are introduced [46,47]. Although the technique in [46] thwarts smart meter private key leakages, it only achieves one-way authentication which might expose intelligent terminals to malicious control and operation by adversaries. On its part, the protocol in [47] is susceptible to impersonation and ephemeral secret leakage attacks [48]. Due to the pairing operations, these two protocols have high computation costs [49]. This problem can be solved by the lightweight scheme in [50]. However, its versatility and communication complexity are increased due to the requirement that the USPs assign random nonces to the smart meters prior to each data collection. Although the protocol in [51] potentially solves this inefficiency issue, its formal security verification has not been done. To prevent cloning attacks, a physically unclonable function (PUF) based protocol are presented in [52–54]. However, PUF-based schemes have stability challenges. On the other hand, the schemes in [55,56] incur extensive computation overheads due to bilinear and scalar multiplications, respectively. Although the protocol in [57] is relatively lightweight, it is not evaluated against threats such as session hijacking and ephemeral secret leakage. Table 1 presents a summary of these current security solutions.

It is evident that many protocols have been developed for security enhancement in smart grids. However, a number of security, performance and privacy issues still lurk in these schemes. The proposed protocol is shown to be efficient, privacy preserving and thwarts most of the attacks inherent in the above schemes.

3. The proposed scheme

The Smart Meter (SM), Service Provider (SP), the Registration Authority (RA) and the Cloud Servers (CSs) are the main components of the proposed protocol as shown in Fig 1. Here, the smart meter collects and forwards power consumption reports to the utility service provider.

However, all the smart meters and service provider must first register at the registration authority so that they are assigned the security tokens to use in the later phases. As already explained, we deploy cloud servers to offload the transaction management tasks from the service providers. Table 2 presents the symbols used throughut this paper.

Basically, our protocol comprises of 5 major phases, which include system setup, registration, mutual authentication and key negotiation, key and transactions management. The sub-sections below describe these phases in greater details.

3.1 System setup

The goal of this phase is to have the RA generate security parameters for all the network entities. These parameters are then deployed in the proceeding phases of the proposed protocol. For signature generation and verification, we deply the Elliptic Curve Digital Signature Algorithm (ECDSA). However, the Practical Byzantine Fault Tolerance (PBFT) is utilized as a consensus algorithm. Here, non-singular ellipic curve (NS-EC) and Galois field (GF) are utilized as described in the following steps.

Step 1: The RA generates ID_{RA} as its unique identity before selecting some large prime number q and NS-EC over the GF(q). Considering some two constants a and b, where a, $b \in Z_q = \{0, 1, 2, \ldots q-1\}$, then the condition $4a^3 + 27b^2 \neq 0$ (mod q) must be satisfied. Here, NS-EC is of the form E_q (a,b): $y^2 = x^3 + ax + b$ (mod q).

Table 1. Summary of related works.

Scheme	Technique	Gap (s)
[2]	Certificate	High computation costs
[5]	Blockchain	Not evaluated against such as privileged insider and ephemeral secret leakage
[21]	Signature	Fails to support untraceability and protection against attacks such as side-channeling
[19,20]	Certificate	Lack evaluation against attacks such as privileged insider & side-channeling
[22]	Blockchain	Not analyzed against side-channeling and forgery attacks
[23]	Blockchain	Lacks evaluation against forgery attacks
[24]	Blockchain	Not evaluated against such as privileged insider and ephemeral secret leakage
[25]	Blockchain	Lacks security analysis
[26]	Blockchain	Lacks formal security analysis
[11,27–37]	Signature	Susceptible to quantum attacks
[39]	ECC	Reliance on a third party for secure session establishment
[40]	ECC	High communication and computation overheads
[41]	Homomorphic	Low fault tolerance
[42]	Homomorphic	High communication overheads
[43]	Homomorphic	High computation costs
[46]	Bilinear pairing	Vulnerable to malicious control and operation
[47]	Bilinear pairing	Susceptible to impersonation & ephemeral secret leakage attacks
[50]	Hashing, XOR	High communication complexity
[51]	ECC	Formal security verification is not done
[52–54]	PUF	Stability challenges
[55]	Bilinear pairing	High computation costs
[56]	ECC	Extensive computation overheads
[57]	ECC	Not evaluated against threats such as ephemeral secret leakage and session hijacking.

https://doi.org/10.1371/journal.pone.0318182.t001

Step 2: The RA picks $G \in E_q(a, b)$as the base point, whose order is g, which is as large as q. Next, it chooses $h(.)$ as some collision-resistant one way hashing function. It also chooses E_{sig} and E_{ver} as ECDSA signature generation and verification algorithms respectively. Moreover, PBFT is chosen as the consensus algorithm.
Step 3: The RA selects $MK_{RA} \in Z_q^*$as its secret master key before using this key to derive its corresponding public key PK_{RA} = G. MK_{RA}. At the end, the RA secretly stores MK_{RA} before publishing parameter set {G, PK_{RA}, PBFT, h(.), $E_q(a, b)$, E_{sig}, E_{ver}} as shown in Fig 2.

3.2 Registration phase

The aim of this phase is to register the cloud server, smart meters and the utility service providers. This registration is carried out with the help of the RA and is one-time process which is described in the sub-sections below.

3.2.1 Cloud server registration.

The following 3 steps are executed to register cloud server CS_i to the RA.
Step 1: The RA generates CS_i unique identity ID_{CS} and some symmetric n-degree bivariate polynomial $f(c, d)$ over finite field GF(q). Here, $p(c, d) = \sum_{i=0}^{n} \sum_{j=0}^{n} e_{ij} c^i d^j$, where $p(c, d) = p(d, c)$ and$e_{ij} \in Z_q$. The RA determines the current timestamp T_1 and derives CS_i pseudo-identity $PID_{CS} = h(ID_{CS}||MK_{RA}||T_1)$.
Step 2: The RA generates random secret key $\eta_1 \in Z_q^*$ that is used to compute its corresponding public key $P_{CS} = G.\eta_1$. Next, RA creates CS_i certificate as $C_{CS} = \eta_1 + h(PID_{CS}||ID_{RA}||P_{CS}||PK_{RA}) * MK_{RA}$ (mod q). It then publishes P_{CS} before constructing registration message R_1 = {ID_{RA}, C_{CS}, PID_{CS}, $f(PID_{CS}, d)$} that is forwarded to CS_i over secure channels. Finally, the RA deletes random secret key η_1 as shown in Fig 2.

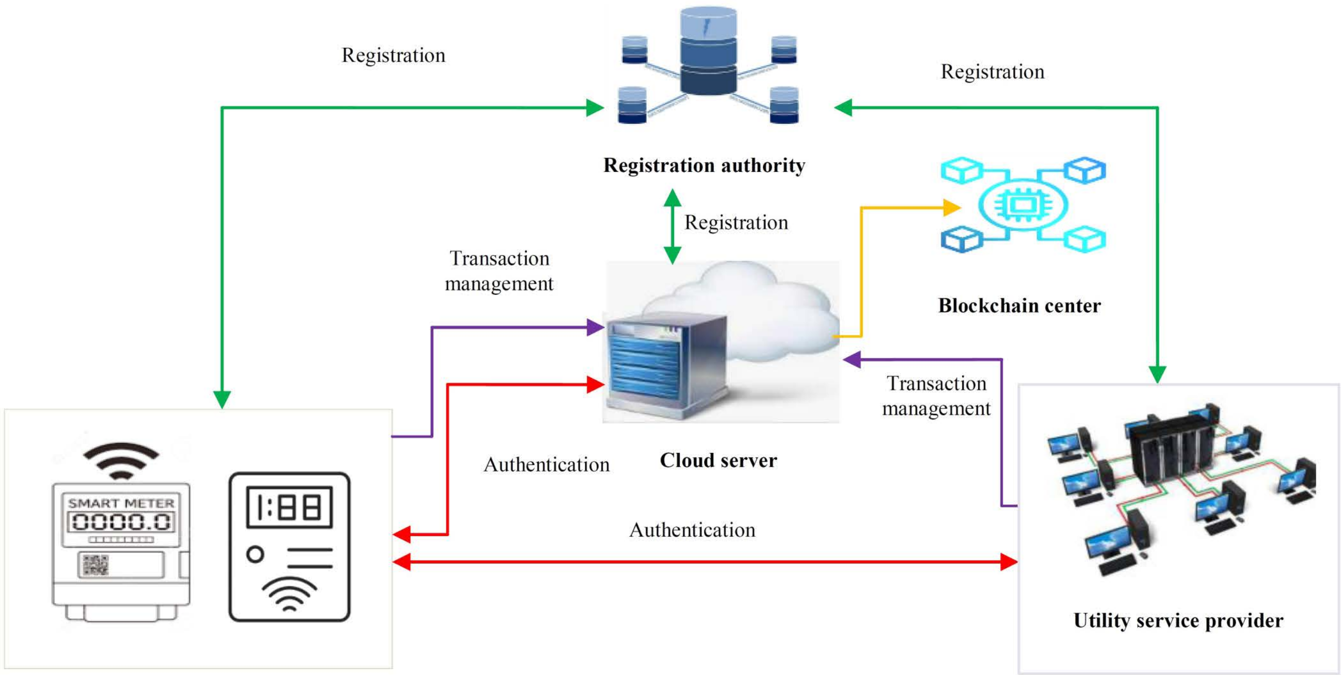


Fig 1. Network model.

https://doi.org/10.1371/journal.pone.0318182.g001

Step 3: Upon receiving registration message R_1 from RA, CS_i proceeds to generate its secret key $SK_{CS} \in Z_q^*$ and computes its corresponding public key $PK_{CS} = G.SK_{CS}$. Finally, CS_i stores parameter set $\{ID_{RA}$, C_{CS}, PID_{CS}, $f(PID_{CS}, d)$, $(SK_{CS}, PK_{CS})\}$.

3.2.2 Smart meter registration. The following 4 steps are carried out during the registration of smart meter SM_j to the RA.

Step 1: The RA determines the current timestamp T_2 and generates smart meter unique identity ID_{SM} that is used to derive its pseudo-identity $PID_{SM} = h(ID_{SM}||MK_{RA}||T_2)$. Next, it generates its random transient identity TID_{SM}.

Step 2: RA chooses some random private key $\eta_2 \in Z_q^*$ and derives its corresponding public key $P_{SM} = G.\eta_2$. This is followed by the generation of SM_j certificate $C_{SM} = \eta_2 + h(PID_{SM}||ID_{RA}||P_{SM}||PK_{RA}) * MK_{RA} \pmod q$.

Step 3: The RA generates secret key $SK_{SM} \in Z_q^*$ together with its corresponding public key $PK_{SM} = G.SK_{SM}$ Next, it composes registration message $R_2 = \{(TID_{SM}, PID_{SM}), C_{SM}, f(PID_{SM}, d), ID_{RA}, (SK_{SM}, PK_{SM})\}$ that is forwarded to the SM_j for safe storage of this parameter set as shown in Fig 2.

Step 4: RA composes registration message $R_3 = \{TID_{SM}, PID_{SM}\}$ that is forwarded to CS_i over secure channels. Finally, the RA erases random secret key η_2.

3.2.3 Utility service provider. The following 3 steps are involved during the registration of utility service provider SP_j.

Step 1: The RA determines current timestamp T_3 and selects some unique identity ID_{SP} for SP_j that is used to compute its pseudo-identity $PID_{SP} = h\,(ID_{SP}||MK_{RA}||T_3)$. Next, it chooses random transient identity TID_{SP} for SP_j.

Step 2: The RA generates secret key $SK_{SP} \in Z_q^*$ and equivalent public key $PK_{SP} = G.SK_{SP}$. This is followed by the publishing of PK_{SP}.

Step 3: RA securely stores parameter set $\{(TID_{SP}, PID_{SP}), (SK_{SP}, PK_{SP})\}$. Next, it constructs registration message $R_4 = \{TID_{SP}, PID_{SP}\}$ that is sent to the associatated SP_j over secure channels as shown in Fig 2.

Table 2. Symbols.

Symbol	Description
G	Base point in an elliptic curve
ID_{RA}	Unique identity of the registration authority
MK_{RA}	RA's private master key
PK_{RA}	RA's public key
E_{sig}	ECDSA signature generation algorithm
E_{ver}	ECDSA signature verification algorithm
D_k	Decryption using key k
ID_{CS}	Unique identity of the cloud server
PID_{CS}	Cloud server pseudo-identity
T_i	Timestamp i
ΔT	Maximum permissible transmission delay
η_i	Random secret key i
C_{CS}	Cloud server certificate
SK_{CS}	Cloud server secret key
PK_{CS}	Cloud server public key
ID_{SM}	Smart meter unique identity
SK_{SM}	Smart meter secret key
TID_{SM}	Transient smart meter identity
ID_{SP}	Unique identity of the utility service provider
TID_{SP}	Transient utility service provider identity
SK_{SP}	Utility service provider secret key
$\|\|$	Concatenation operation
R_i	Random nonce i
$\phi_{SM,}\ \phi_{SP,}\ \phi_{SC}$	Session keys
$h(.)$	Collision-resistant one-way hash function
$\oplus$	XOR operation

https://doi.org/10.1371/journal.pone.0318182.t002

3.3 Mutual authentication and key negotiation

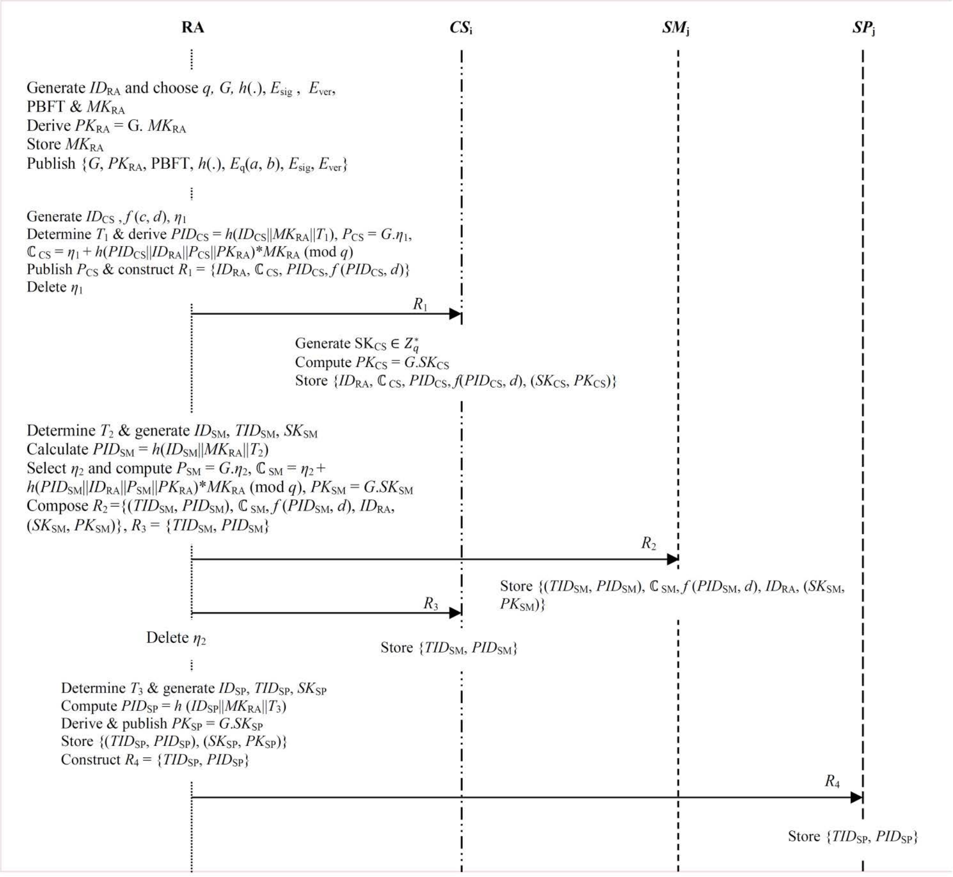
During the mutual authentication between SP_j and SM_j, steps 1–6 are executed. Fig 3 presents a summary of the message exchanges during these procedures.

Step 1: The SP_j generates random nonce $\mathbf{R}_1 \in Z_q^*$ and determines the current timestamp T_4. Next, it derives $A_1 = G.h\,(\mathbf{R}_1||TID_{SP}||PID_{SP}||SK_{SP}||T_4)$ as well as signature $Z_1 = h\,(\mathbf{R}_1||TID_{SP}||SK_{SP}||) + h(PK_{SP}||PK_{SM}||PK_{CS}||T_4) * SK_{SP}$ (mod q). Finally, it composes message authenticatiom message $AM_1 = \{TID_{SP}, A_{1,}\ Z_{1,}\ T_4\}$ that is sent over to SM_j over public channels as shown in Fig 3.

Step 2: Upon receiving authentication message AM_1, SM_j determines current timestamp T_5 and checks if $|T_5 - T_4| < \Delta T$. Basically, the session is aborted when this verification fails. Otherwise, SM_j validates signature Z_1 by confirming whether $Z_1.G \overset{?}{=} A_1 + h(PK_{SP}||PK_{SM}||PK_{CS}||T_4) * PK_{SP}$. Provided that this confirmation is valid, SM_j determines the current timestamp T_6 and generates random nonce $R_2 \in Z_q^*$.

Step 3: The SM_j derives $A_2 = G.h\,(R_2||TID_{SM}||PID_{SM}||SK_{SM}||T_6)$ and $A_3 = A_1.h\,(R_2||TID_{SM}||PID_{SM}||SK_{SM}||T_6)$ as well as session key $\phi_{SM} = h\,(A_3||Z_1||T_4||T_6)$. Next, it computes the signature on R_2 and ϕ_{SM} as $Z_2 = h\,(R_2||TID_{SM}||PID_{SM}||SK_{SM}||T_6) + h(\phi_{SM}||PK_{SP}||PK_{CS}||T_6) * SK_{SM}$ (mod q).

Step 4: SM_j generates a new transient identity TID_{SP}^{New} for SP_j. Next, it computes $TID_{SP}^{\,*} = TID_{SP}^{New \oplus h(}TID_{SP}||\phi_{SM}||Z_2||T_6)$. Finally, it composes authentication message $AM_2 = \{TID_{SP}^{\,*}, A_2, Z_2, T_6\}$ that is transmitted over to the SP_j through public channels as shown in Fig 3.



Generate ID_{RA} and choose q, G, $h(.)$, E_{sig}, E_{ver}, PBFT & MK_{RA}
Derive $PK_{RA} = G.\ MK_{RA}$
Store MK_{RA}
Publish $\{G, PK_{RA}, \text{PBFT}, h(.), E_q(a, b), E_{sig}, E_{ver}\}$

Generate ID_{CS}, $f(c, d)$, η_1
Determine T_1 & derive $PID_{CS} = h(ID_{CS}||MK_{RA}||T_1)$, $P_{CS} = G.\eta_1$, $\mathbb{C}_{CS} = \eta_1 + h(PID_{CS}||ID_{RA}||P_{CS}||PK_{RA})^* MK_{RA} \pmod q$
Publish P_{CS} & construct $R_1 = \{ID_{RA}, \mathbb{C}_{CS}, PID_{CS}, f(PID_{CS}, d)\}$
Delete η_1

Generate $SK_{CS} \in Z_q^*$
Compute $PK_{CS} = G.SK_{CS}$
Store $\{ID_{RA}, \mathbb{C}_{CS}, PID_{CS}, f(PID_{CS}, d), (SK_{CS}, PK_{CS})\}$

Determine T_2 & generate ID_{SM}, TID_{SM}, SK_{SM}
Calculate $PID_{SM} = h(ID_{SM}||MK_{RA}||T_2)$
Select η_2 and compute $P_{SM} = G.\eta_2$, $\mathbb{C}_{SM} = \eta_2 + h(PID_{SM}||ID_{RA}||P_{SM}||PK_{RA})^* MK_{RA} \pmod q$, $PK_{SM} = G.SK_{SM}$
Compose $R_2 = \{(TID_{SM}, PID_{SM}), \mathbb{C}_{SM}, f(PID_{SM}, d), ID_{RA}, (SK_{SM}, PK_{SM})\}$, $R_3 = \{TID_{SM}, PID_{SM}\}$

Store $\{(TID_{SM}, PID_{SM}), \mathbb{C}_{SM}, f(PID_{SM}, d), ID_{RA}, (SK_{SM}, PK_{SM})\}$

Delete η_2

Store $\{TID_{SM}, PID_{SM}\}$

Determine T_3 & generate ID_{SP}, TID_{SP}, SK_{SP}
Compute $PID_{SP} = h(ID_{SP}||MK_{RA}||T_3)$
Derive & publish $PK_{SP} = G.SK_{SP}$
Store $\{(TID_{SP}, PID_{SP}), (SK_{SP}, PK_{SP})\}$
Construct $R_4 = \{TID_{SP}, PID_{SP}\}$

Store $\{TID_{SP}, PID_{SP}\}$

Fig 2. System setup and registration.

https://doi.org/10.1371/journal.pone.0318182.g002

Step 5: On receiving authentication message AM_2 at timestamp T_7, the SP_j checks if $|T_7 - T_6| < \Delta T$. Provided that this verification fails, the session is terminated. Otherwise, it computes $A_4 = A_2.h\ (R_1||TID_{SP}||PID_{SP}||SK_{SP}||T_4)$ and session key $\phi_{SP} = h\ (A_4||Z_1||T_4||T_6)$.

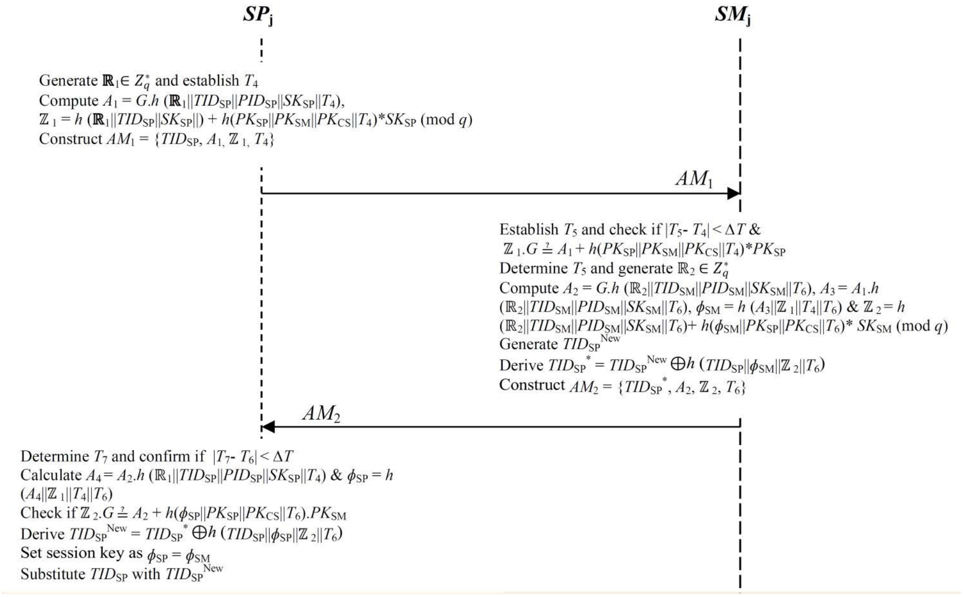
Step 6: The SP_j validates signature Z_2 by confirming if $Z_2.G \overset{?}{=} A_2 + h(\phi_{SP}||PK_{SP}||PK_{CS}||T_6).\ PK_{SM}$. Provided that this validation succeeds, the SP_j computes $TID_{SP}^{New} = TID_{SP}^{* \oplus h(TID_{SP}||\phi_{SP}||Z_2||T_6).\text{ Finally, it substitutes}}$ TID_{SP} with its updated version TID_{SP}^{New} in its repository.

3.4 Key management

The goal of this phase is to secure the transactions transmitted to the cloud servers from any of the smart grid device. For a given utility service area U_{SA} ($SA = 1,2,3,\ldots.N$), steps 1–6 are carried out to setup the session keys between the cloud server CS_i and any smart device such as SM_j.

Step 1: The CS_i generates random nonce $\mathbf{R}_3 \in Z_q^*$ and determines the current timestamp T_8. Next, it derives $B_1 = G.h\ (\mathbf{R}_3||SK_{CS}||PID_{CS}||T_8)$, $PID_{CS}^* = PID_{CS}^{\oplus h(PID_{SM}||ID_{RA}||T_8)\text{ and}}$ $Z_3 = h\ (\mathbf{R}_3||SK_{CS}||PID_{CS}||T_8) + h\ (PK_{CS}||C_{CS}||PID_{CS}^*||TID_{SM}) * SK_{CS} \pmod q$. Finally, it constructs key management message $KM_1 = \{TID_{SM}, B_1, C_{CS}, Z_3, PID_{CS}^*, T_8\}$ that is forwarded towards SM_j over public channels as shown in Fig 4.

Step 2: Upon receiving message KM_1 at timestamp T_9, SM_j confirms whether $|T_9 - T_8| < \Delta T$. On condition that this verification is successful, SM_j derives $PID_{CS} = PID_{CS}^{* \oplus h}(PID_{SM}||ID_{RA}||T_8)$. Next, it validates the received certificate C_{CS}, signature



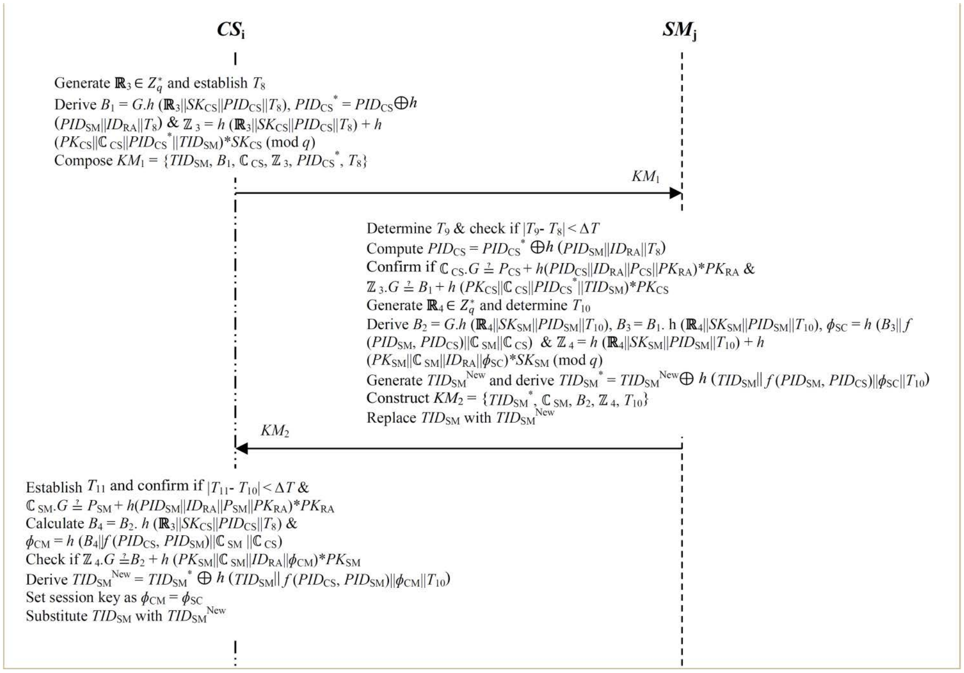
Generate $\mathbb{R}_1 \in Z_q^*$ and establish T_4
Compute $A_1 = G.h\ (\mathbb{R}_1||TID_{SP}||PID_{SP}||SK_{SP}||T_4)$,
$\mathbb{Z}_1 = h\ (\mathbb{R}_1||TID_{SP}||SK_{SP}||) + h(PK_{SP}||PK_{SM}||PK_{CS}||T_4)*SK_{SP}$ (mod q)
Construct $AM_1 = \{TID_{SP}, A_1, \mathbb{Z}_1, T_4\}$

Establish T_5 and check if $|T_5 - T_4| < \Delta T$ &
$\mathbb{Z}_1.G \stackrel{?}{=} A_1 + h(PK_{SP}||PK_{SM}||PK_{CS}||T_4)*PK_{SP}$
Determine T_5 and generate $\mathbb{R}_2 \in Z_q^*$
Compute $A_2 = G.h\ (\mathbb{R}_2||TID_{SM}||PID_{SM}||SK_{SM}||T_6)$, $A_3 = A_1.h$
$(\mathbb{R}_2||TID_{SM}||PID_{SM}||SK_{SM}||T_6)$, $\phi_{SM} = h\ (A_3||\mathbb{Z}_1||T_4||T_6)$ & $\mathbb{Z}_2 = h$
$(\mathbb{R}_2||TID_{SM}||PID_{SM}||SK_{SM}||T_6) + h(\phi_{SM}||PK_{SP}||PK_{CS}||T_6)*\ SK_{SM}$ (mod q)
Generate $TID_{SP}{}^{New}$
Derive $TID_{SP}{}^* = TID_{SP}{}^{New} \oplus h\ (TID_{SP}||\phi_{SM}||\mathbb{Z}_2||T_6)$
Construct $AM_2 = \{TID_{SP}{}^*, A_2, \mathbb{Z}_2, T_6\}$

Determine T_7 and confirm if $|T_7 - T_6| < \Delta T$
Calculate $A_4 = A_2.h\ (\mathbb{R}_1||TID_{SP}||PID_{SP}||SK_{SP}||T_4)$ & $\phi_{SP} = h$
$(A_4||\mathbb{Z}_1||T_4||T_6)$
Check if $\mathbb{Z}_2.G \stackrel{?}{=} A_2 + h(\phi_{SP}||PK_{SP}||PK_{CS}||T_6).PK_{SM}$
Derive $TID_{SP}{}^{New} = TID_{SP}{}^* \oplus h\ (TID_{SP}||\phi_{SP}||\mathbb{Z}_2||T_6)$
Set session key as $\phi_{SP} = \phi_{SM}$
Substitute TID_{SP} with $TID_{SP}{}^{New}$

Fig 3. Mutual authentication and key negotiation.

https://doi.org/10.1371/journal.pone.0318182.g003



Generate $\mathbb{R}_3 \in Z_q^*$ and establish T_8
Derive $B_1 = G.h\ (\mathbb{R}_3||SK_{CS}||PID_{CS}||T_8)$, $PID_{CS}{}^* = PID_{CS} \oplus h$
$(PID_{SM}||ID_{RA}||T_8)$ & $\mathbb{Z}_3 = h\ (\mathbb{R}_3||SK_{CS}||PID_{CS}||T_8) + h$
$(PK_{CS}||\mathbb{C}_{CS}||PID_{CS}{}^*||TID_{SM})*SK_{CS}$ (mod q)
Compose $KM_1 = \{TID_{SM}, B_1, \mathbb{C}_{CS}, \mathbb{Z}_3, PID_{CS}{}^*, T_8\}$

Determine T_9 & check if $|T_9 - T_8| < \Delta T$
Compute $PID_{CS} = PID_{CS}{}^* \oplus h\ (PID_{SM}||ID_{RA}||T_8)$
Confirm if $\mathbb{C}_{CS}.G \stackrel{?}{=} P_{CS} + h(PID_{CS}||ID_{RA}||P_{CS}||PK_{RA})*PK_{RA}$ &
$\mathbb{Z}_3.G \stackrel{?}{=} B_1 + h\ (PK_{CS}||\mathbb{C}_{CS}||PID_{CS}{}^*||TID_{SM})*PK_{CS}$
Generate $\mathbb{R}_4 \in Z_q^*$ and determine T_{10}
Derive $B_2 = G.h\ (\mathbb{R}_4||SK_{SM}||PID_{SM}||T_{10})$, $B_3 = B_1.\ \mathrm{h}\ (\mathbb{R}_4||SK_{SM}||PID_{SM}||T_{10})$, $\phi_{SC} = h\ (B_3||\ f$
$(PID_{SM}, PID_{CS})||\mathbb{C}_{SM}||\mathbb{C}_{CS})$ & $\mathbb{Z}_4 = h\ (\mathbb{R}_4||SK_{SM}||PID_{SM}||T_{10}) + h$
$(PK_{SM}||\mathbb{C}_{SM}||ID_{RA}||\phi_{SC})*SK_{SM}$ (mod q)
Generate $TID_{SM}{}^{New}$ and derive $TID_{SM}{}^* = TID_{SM}{}^{New} \oplus\ h\ (TID_{SM}||\ f\ (PID_{SM}, PID_{CS})||\phi_{SC}||T_{10})$
Construct $KM_2 = \{TID_{SM}{}^*, \mathbb{C}_{SM}, B_2, \mathbb{Z}_4, T_{10}\}$
Replace TID_{SM} with $TID_{SM}{}^{New}$

Establish T_{11} and confirm if $|T_{11} - T_{10}| < \Delta T$ &
$\mathbb{C}_{SM}.G \stackrel{?}{=} P_{SM} + h(PID_{SM}||ID_{RA}||P_{SM}||PK_{RA})*PK_{RA}$
Calculate $B_4 = B_2.\ h\ (\mathbb{R}_3||SK_{CS}||PID_{CS}||T_8)$ &
$\phi_{CM} = h\ (B_4||f\ (PID_{CS}, PID_{SM})||\mathbb{C}_{SM}\ ||\mathbb{C}_{CS})$
Check if $\mathbb{Z}_4.G \stackrel{?}{=} B_2 + h\ (PK_{SM}||\mathbb{C}_{SM}||ID_{RA}||\phi_{CM})*PK_{SM}$
Derive $TID_{SM}{}^{New} = TID_{SM}{}^* \oplus\ h\ (TID_{SM}||\ f\ (PID_{CS}, PID_{SM})||\phi_{CM}||T_{10})$
Set session key as $\phi_{CM} = \phi_{SC}$
Substitute TID_{SM} with $TID_{SM}{}^{New}$

Fig 4. Key management.

https://doi.org/10.1371/journal.pone.0318182.g004

Z_3 and TID_{SM} by checking if $C_{CS}.G \stackrel{?}{=} P_{CS} + h(PID_{CS}||ID_{RA}||P_{CS}||PK_{RA})*PK_{RA}$ and $Z_3.G \stackrel{?}{=} B_1 + h\ (PK_{CS}||C_{CS}||PID_{CS}{}^*||TID_{SM})*PK_{CS}$. Provided that these conditions do not hold, the session is aborted. Otherwise, SM_j generates random nonce $\mathbf{R}_4 \in Z_q^*$ and determines the current timestamp T_{10}.

Step 3: The SM_j derives $B_2 = G.h\,(\mathbf{R}_4||SK_{SM}||PID_{SM}||T_{10})$, $B_3 = B_1.\ \mathrm{h}\,(\mathbf{R}_4||SK_{SM}||PID_{SM}||T_{10})$, $\phi_{SC} = h\,(B_3||\ f\,(PID_{SM}, PID_{CS})||C_{SM}||C_{CS})$ and $Z_4 = h\,(\mathbf{R}_4||SK_{SM}||PID_{SM}||T_{10}) + h\,(PK_{SM}||C_{SM}||ID_{RA}||\phi_{SC}) * SK_{SM}$ (mod q).
Step 4: The SM_j generates $TID_{SM}{}^{New}$ and computes $TID_{SM}{}^{*} = TID_{SM}{}^{New \oplus h\,(}TID_{SM}{}^{||}\ f\,(PID_{SM}, PID_{CS})||\phi_{SC}||T_{10})$. It then composes key management message $KM_2 = \{TID_{SM}{}^{*}, C_{SM}, B_2, Z_4, T_{10}\}$ that is transmitted over to CS_i via public channels. Finally, SM_j substitutes TID_{SM} with its updated version $TID_{SM}{}^{New}$.
Step 5: On receiving message KM_2 at timestamp $T11$, the CS_i checks if $|\,T_{11} - T_{10}\,| < \Delta T$ such that the session is aborted upon validation failure. Otherwise, it validates the received certificate C_{SM} by confirming whether $C_{SM}.G \overset{?}{=} P_{SM} + h(PID_{SM}||ID_{RA}||P_{SM}||PK_{RA}) * PK_{RA}$. Provided that this verification is unsuccessful, the session is terminated. Otherwise, it derives $B_4 = B_2.\ h\,(\mathbf{R}_3||SK_{CS}||PID_{CS}||T_8)$ and session key $\phi_{CM} = h\,(B_4||f\,(PID_{CS}, PID_{SM})||C_{SM}\,||C_{CS})$.
Step 6: CS_i validates signature Z_4 by checking if $Z_4.G \overset{?}{=} B_2 + h\,(PK_{SM}||C_{SM}||ID_{RA}||\phi_{CM}) * PK_{SM}$. If this verification is successful, it derives $TID_{SM}{}^{New} = TID_{SM}{}^{* \oplus h\,(}TID_{SM}{}^{||}\ f\,(PID_{CS}, PID_{SM})||\phi_{CM}||T_{10})$. Next, both the CS_i and SM_j sets their respective session key for payload enciphering and substitutes TID_{SM} with its update version $TID_{SM}{}^{New}$ in its repository.

3.5 Transactions management

The data such as in appendix A collected by the smart devices in the smart grid system are regarded as being private and confidential. As such, the data from all the utility service provider coverage area are maintained in the private blockchain. In the proposed protocol, transactions are maintained in form of connected chain of blocks stored in the cloud servers. At each particular moment, the voting based consensus algorithm is deployed to ensure that each cloud server holds a similar copy of blockchain BC. Since most of the smart devices in the smart grid system are limited in terms of computation power, they cannot be charged with the creation of transactions for the blockchain. Therefore, the cloud servers are assigned this task since they have superior computational and storage resources. The four major phases in the PBFT consensus algorithm are depicted in Fig 5 below.

Using this consensus algorithm the four steps below describe the process of block addition and verification.
Step 1: The smart meters and cloud servers deploy session keys ϕ_{CM} and ϕ_{SC} already negotiated in Section 3.4 above to exchange all the collected data of appendix A. Thereafter, for a given block β_n, CS_i makes κ_τ transactions ($\Psi_1, \Psi_2, \Psi_3, \ldots, \Psi\kappa_\tau$). Next, CS_i enciphers these transactions using PK_{CS} as $\{E_{PK_{CS}}(\Psi_1), E_{PK_{CS}}(\Psi_2), E_{PK_{CS}}(\Psi_3), \ldots .E_{PK_{CS}}(\Psi_{\kappa_\tau})\}$.
Step 2: CS_i uses its secret key SK_{CS} to create a digital signature of the κ_τ transactions as $E_{sig_{\beta_n}} = E_{sig_{SK_{CS}}}(h(E_{PK_{CS}}(\Psi_1)||E_{PK_{CS}}(\Psi_2)||E_{PK_{CS}}(\Psi_3)||\ldots .||E_{PK_{CS}}(\Psi_{\kappa_\tau}))$. Next, it constructs transaction management message $T_M = \{(E_{PK_{CS}}(\Psi_i)\,|i = 1, 2, 3, \ldots \kappa_\tau), E_{sig_{\beta_n}}\}$ that basically uses to forward these enciphered κ_τ transactions to the blockchain center C_{B_i} as shown in Fig 6.
Step 3: Upon receiving message T_M, the C_B creates block β_n. Basically β_n contains details such as the previous block hash H_P, the current block hash H_C, signature $E_{sig_{\beta_n}}$, enciphered transactions κ_τ, Merkle tree root R on κ_τ, as well the public key PK_{CS} for CS_i. The detailed structure of β_n is depicted in Fig 7 below.
Step 4: Upon the formation of β_n at the C_{B_i}, the leader selection algorithm is invoked to choose the leader. Next, consensus is built for block verification and addition to the blockchain as detailed in Algorithm 1.

During consensus building, each of the blockchain centers C_{B_i} is characterized by a pair of public-secret key pair $\{SK_{C_{B_i}}, PK_{C_{B_i}}\}$. Here, $SK_{C_{B_i}} \in Z_q^*$ is the secret key while $PK_{C_{B_i}} = G.SK_{C_{B_i}}$ is its corresponding public key. Basically, $PK_{C_{B_i}}$

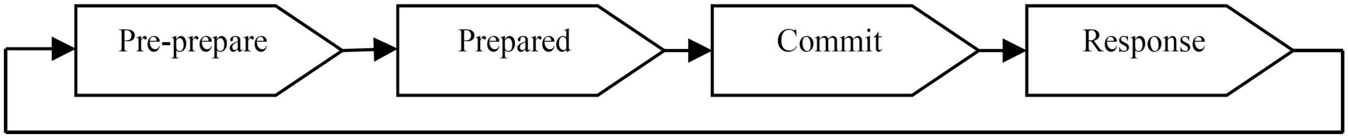
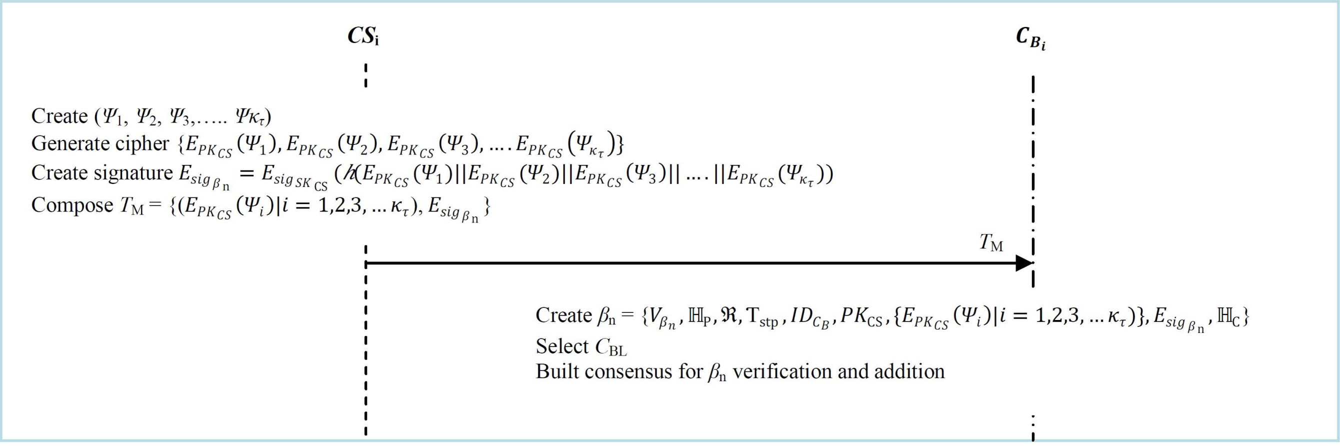


Fig 5. The PBFT consensus algorithm.

https://doi.org/10.1371/journal.pone.0318182.g005

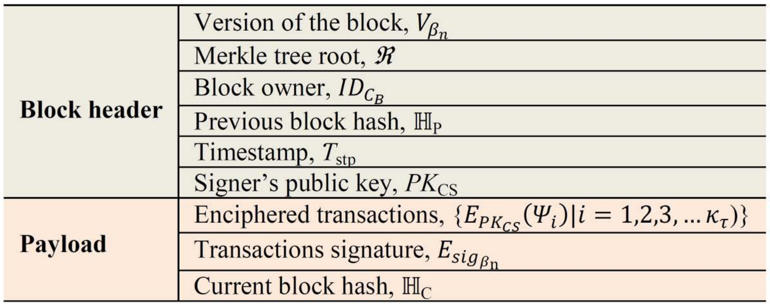


Create $(\Psi_1, \Psi_2, \Psi_3, \ldots\ldots \Psi\kappa_\tau)$
Generate cipher $\{E_{PK_{CS}}(\Psi_1), E_{PK_{CS}}(\Psi_2), E_{PK_{CS}}(\Psi_3), \ldots . E_{PK_{CS}}(\Psi_{\kappa_\tau})\}$
Create signature $E_{sig_{\beta_n}} = E_{sig_{SK_{CS}}}(\hbar(E_{PK_{CS}}(\Psi_1)||E_{PK_{CS}}(\Psi_2)||E_{PK_{CS}}(\Psi_3)|| \ldots . ||E_{PK_{CS}}(\Psi_{\kappa_\tau}))$
Compose $T_M = \{(E_{PK_{CS}}(\Psi_i)|i = 1,2,3, \ldots \kappa_\tau), E_{sig_{\beta_n}}\}$

Create $\beta_n = \{V_{\beta_n}, \mathbb{H}_P, \mathfrak{R}, T_{stp}, ID_{C_B}, PK_{CS}, \{E_{PK_{CS}}(\Psi_i)|i = 1,2,3, \ldots \kappa_\tau)\}, E_{sig_{\beta_n}}, \mathbb{H}_C\}$
Select C_{BL}
Built consensus for β_n verification and addition

Fig 6. Transactions management.

https://doi.org/10.1371/journal.pone.0318182.g006



Block header	Version of the block, V_{β_n}
	Merkle tree root, $\mathfrak{R}$
	Block owner, ID_{C_B}
	Previous block hash, $\mathbb{H}_P$
	Timestamp, T_{stp}
	Signer's public key, PK_{CS}
Payload	Enciphered transactions, $\{E_{PK_{CS}}(\Psi_i)|i = 1,2,3, \ldots \kappa_\tau)\}$
	Transactions signature, $E_{sig_{\beta_n}}$
	Current block hash, $\mathbb{H}_C$

Fig 7. Structure of block β_n.

https://doi.org/10.1371/journal.pone.0318182.g007

of all blockchain centers are known to each other. As shown in Algorithm 1, the inputs to the consensus process include number of faulty nodes X_F in the C_{B_i}, β_n and $\{SK_{C_{B_i}}, PK_{C_{B_i}}\}$. Here, the leader C_{B_i} is denoted by C_{BL} and one of its responsibilities is to generate voting requests V_{rs}. Therefore, it initially generates numeous enciphered voting requests V_{RS} utilizing the public keys of the receiver C_{B_i}, denoted as C_{BR}. In addition, it maintains some valid vote counter C_L for the received votes. Thereafter, C_{BL} signs these V_{RS} before forwarding them to the respective followers C_{BRs} together with β_n. Upon receiving the signed V_{RS}, the C_{BR} verifies the signature $\mathbb{Z}_{V_{rs}}$in this request, deciphers it using its $SK_{C_{B_i}}$and validates the timestamp $T_{C_{BR}}$in V_{RS}, R as well as H_P.

Provided that these validations are successful, C_{BR} forwards its signature, voting response V_r along with the status of this verification V_S to the C_{BL}. Here, V_r is encrypted using the public key$PK_{C_{BL}}$ of C_{BL}.

After getting this response, C_{BL} validates the C_{BR}'s signature before counting the votes maintained by C_L. This happens only when both V_r is valid and β_n validation is successful. Upon receiving all the responses, C_{BL} checks whether $C_L \quad 1 + 2 * X_F$. Provided that this condition holds, C_{BL} sends a commit block command C_{BC} to all C_{BRs}. Consequently, β_n is appended to the distributed ledgers of all the peer nodes.

3.6 Secure addition of new smart grid devices

In this phase, we detail how additional smart devices such as SD_k may be incorporated into the existing smart grid network. This is a 4-step process as described below and summarized in Fig 8.

Algorithm 1. Consensus for β_n verification and addition.

```
BEGIN

    Choose $C_{BL}$ with $\beta_n = \{V_{\beta_n}, \mathbb{H}_P, \mathfrak{R}, T_{stp}, ID_{C_B}, PK_{CS}, \{E_{PK_{CS}} \Psi_{i} \mid i = 1,2,3,\ldots\kappa_\tau)\}, E_{sig_{\beta_n}}, \mathbb{H}_C\}$
    Generate $T_{C_{BR}}$ for each $C_{BR}$ peer node
    Initialize $C_L$ and initiate voting process
    FOR each $C_{BR_k}, (k = 1,2,3,..\kappa_{C_B}, C_{BL} \neq C_{BR})$ DO:
        Derive $V_{RS} = E_{PK_{C_{B_i}}}(V_{rs}, T_{C_{BR}})$ and signature on $V_{rs}$ as $\mathbb{Z}_{V_{rs}} = E_{sig_{kC_{BL}}}(V_{RS})$
        Compose $M_1 = \{\beta_n, V_{RS}, \mathbb{Z}_{V_{rs}}\}$ to each $C_{BR}$
        Each $C_{BR}$ receives $M_1$ from $C_{BL}$ at time $T^{\square}_{C_{BR}}$
        FOR each$C_{BR_k}$ DO:
            Validate $\mathbb{Z}_{V_{rs}}$ using $E_{ver}$
            IF $\mathbb{Z}_{V_{rs}}$ is legitimate THEN:
                Derive $(V_{rs}, T_{C_{BR}}) = D_{SK_{C_{B_i}}}[V_{RS}]$
                IF $|T^{\square}_{C_{BR}} - T_{C_{BR}}| < \Delta T$ THEN:
                    Compute $\mathfrak{R}^*$ on enciphered transactions in $\beta_n$
                    IF $\mathfrak{R}^* = \mathfrak{R}$ THEN:
                        Derive block hash $\mathbb{H}_C^*$ on $\{V_{\beta_n}, \mathbb{H}_P, \mathfrak{R}, T_{stp}, ID_{C_B}, PK_{CS}, \{E_{PK_{CS}} \Psi_{i} \mid i = 1,2,3,\ldots\kappa_\tau)\}, E_{sig_{\beta_n}}\}$
                        IF $\mathbb{H}_C^* = \mathbb{H}_C$ THEN:
                            Computed $\mathbb{Z}_{V_r} = E_{sig_{SK_{C_{B_i}}}}[E_{PK_{C_{BL}}}(V_r, V_S)]$
                            Construct Send $V_S$ and $V_r$ as $M_2 = \{E_{PK_{C_{BL}}}(V_r, V_S), \mathbb{Z}_{V_r}\}$
                        ENDIF
                    ENDIF
                ENDIF
            ENDIF
        ENDFOR
     Set $C_L \leftarrow 0$
    FOR each received $M_2$ from $C_{BR_k}$ DO:
        Validate $\mathbb{Z}_{V_r}$ using $E_{ver}$
        IF $\mathbb{Z}_{V_r}$ is legitimate THEN:
            Derive $(V_r, V_S) = D_{kC_{BL}}[E_{PK_{C_{BL}}}(V_r, V_S)]$
            IF $V_r$ and $V_S$ are valid THEN:
                Increament $C_L$ as $C_L = C_L + 1$
            ENDIF
        ENDIF
    ENDFOR
    IF $C_L > 1 + 2 \square X_F$ THEN:
        Send $C_{BC}$ to all $C_{BRs}$
        Add $\beta_n$ to the $BC$
    ENDIF
    ENDFOR

END
```

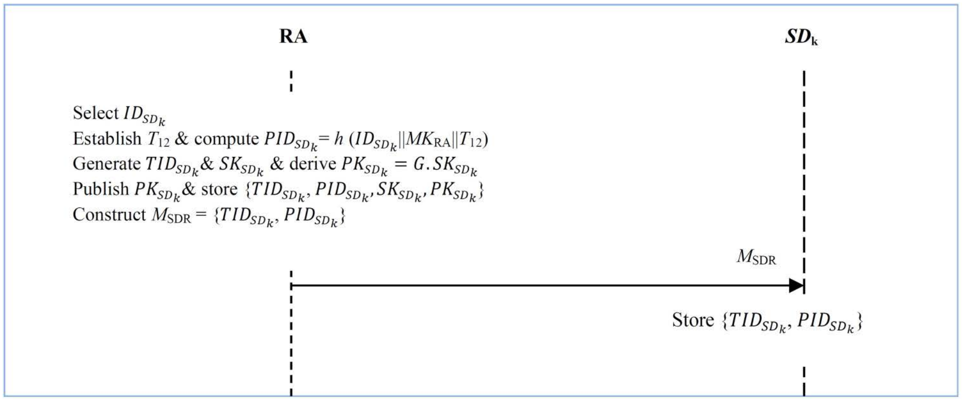

RA
SD_k
Select ID_{SD_k}
Establish T_{12} & compute $PID_{SD_k} = h\,(ID_{SD_k}||MK_{RA}||T_{12})$
Generate TID_{SD_k} & SK_{SD_k} & derive $PK_{SD_k} = G.SK_{SD_k}$
Publish PK_{SD_k} & store $\{TID_{SD_k}, PID_{SD_k}, SK_{SD_k}, PK_{SD_k}\}$
Construct $M_{SDR} = \{TID_{SD_k}, PID_{SD_k}\}$
M_{SDR}
Store $\{TID_{SD_k}, PID_{SD_k}\}$

Fig 8. New smart grid devices addition.

https://doi.org/10.1371/journal.pone.0318182.g008

Step 1: The RA chooses some unique real identity ID_{SD_k} for SD_k. Next, it determines current timestamp and T_{12} derives the pseudo-identity for SD_k as $PID_{SD_k} = h\,(ID_{SD_k}||MK_{RA}||T_{12})$ as shown in Fig 8.
Step 2: RA generates random transient identity TID_{SD_k}. This is followed by the derivation of its secret key $SK_{SD_k} \in Z_q^*$. Next, it computes its corresponding public key as $PK_{SD_k} = G.SK_{SD_k}$.
Step 3: The RA makes PK_{SD_k}public before securely storing parameter set $\{TID_{SD_k}, PID_{SD_k},\ SK_{SD_k}, PK_{SD_k}\}$ in its repository.
Step 4: RA constructs smart device registration message $M_{SDR} = \{TID_{SD_k}, PID_{SD_k}\}$ that is forwarded to the smart device SD_k.

4. Security analysis

In this section, the formal and informal security analysis of the proposed protocol ae presented. The sub-sections below describe these process in greater details.

4.1 Formal security analysis

In this section, we deploy the oracle model Real-Or-Random (ROR) to demonstrate that provable secure nature of the derived session keys. In our scheme, session keys are derived between utility service provider SP_j and smart meter SM_j, as well as between cloud server CS_i and any smart device within the smart grid, such as smart meter SM_j. We denote the adversary as $\mathring{A}$, which is capable of launching *Execute* (), *Corrupt* (), *Reveal* () and *Test* () queries. Taking O_T as an arbitrary outcome of a flipped a fair coin ε, these queries are described in more detail in Table 3. In addition to these queries, *h* (.) is modeled as random oracle *Hash* which is available to $\mathring{A}$ as well as all other network entities SM_j, SP_j and CS_i. We denote the ith, jth and kth instances (random oracles) of SM_j, SP_j and CS_i as $\lneqq^i_{SM}$, $\lneqq^j_{SP}$ and $\lneqq^k_{CS}$. Suppose that instant $\lneqq^m$ receives the final legitimate exchanged message. In this case, $\lneqq^m$ is regarded as being in an accepted state. In addition, we denote the sequential ordering of all the exchanged messages in a given communication session as $\dot{S}$. Basically, $\dot{S}$ becomes the session identifier of $\lneqq^m$ for this particular communication session. When random oracles $\lneqq^p$ and $\lneqq^q$ are mutual associates of each other, share the same $\dot{S}$ for mutual authentication and both are in accepted states, then they become associates to each other.

Suppose that SP_j and SM_j share session key $\phi_{SP} = \phi_{SM}$ between them. Then, random oracle $\lneqq^i_{SM}$or $\lneqq^j_{SP}$ is regarded as being fresh this session key is unknown to $\mathring{A}$ even after executing the *Reveal* ($\lneqq^m$) query. Similarly, random oracle $\lneqq^i_{SM}$or $\lneqq^k_{CS}$ is fresh if session key $\phi_{CM} = \phi_{SC}$ remains unknown to $\mathring{A}$ even after executing the *Reveal* ($\lneqq^m$) query. We let p_t denote polynomial time and λ_{CSB} as the proposed certificate, signature and blockchain (CSB) based protocol. In this scenario, the advantage that $\mathring{A}$ (running in p_t) has of breaking λ_{CSB}'s semantic security is represented as $Adv_{\mathring{A}}^{\lambda_{CSB}}(p_t)$. This basically

Table 3. Adversarial queries.

Query	Rationale
Reveal **($\underset{\mp}{<}^{m}$)**	Carried out by $\mathring{A}$ disclose session key $\phi_{SP} = \phi_{SM}$ as well as $\phi_{CM} = \phi_{SC}$ between $\underset{\mp}{<}^{m}$ and its respective associates
Execute ($\underset{\mp}{<}^{i}_{SM}, \underset{\mp}{<}^{j}_{SP}, \underset{\mp}{<}^{k}_{CS}$)	Executed by $\mathring{A}$ to intercept messages exchanged among SP_j, SM_j and CS_i
Test ($\underset{\mp}{<}^{m}$)	Performed by $\mathring{A}$ using O_T to verify the disclosed session keys $\phi_{SP} = \phi_{SM}$ as well as $\phi_{CM} = \phi_{SC}$
Corrupt ($\underset{\mp}{<}^{i}_{SM}, \underset{\mp}{<}^{j}_{SP}$)	Implemented by $\mathring{A}$ to extract secret tokens stored in compromised SM_j and SP_j respectively

https://doi.org/10.1371/journal.pone.0318182.t003

involves the compromise of session key $\phi_{SP} = \phi_{SM}$ established between SM_j and SP_j, as well as $\phi_{CM} = \phi_{SC}$ negotiated between SM_j and CS_i during a given session. Taking ε and ε^* as valid and guessed bits respectively, then,

$$Adv_{\mathring{A}}^{\lambda_{CSB}}(p_t) = |2Pr[\varepsilon* = \varepsilon] - 1| \tag{1}$$

Suppose that the Elliptic Curve Decisional Diffie-Hellman Problem (ECDDHP), volume of *Hash* () queries and range space of *h* (.) are represented by ω, $|\mu|$ and H_n respectively. Using these notations, the advantage that adversary $\mathring{A}$ (running in p_t) has in breaking ω is denoted as $Adv_{\mathring{A}}^{\omega}(p_t)$.

With the above notations, the following hypothesis can be stated.

Hypothesis 1: Suppose that $\mathring{A}$ is running in polynomial time p_t and wants to derive session key $\phi_{SP} = \phi_{SM}$ negotiated between SP_j and SM_j, as well as session key $\phi_{CM} = \phi_{SC}$ established between SM_j and CS_i during a certain communication session. Therefore,

$$Adv_{\mathring{A}}^{\lambda_{CSB}}(p_t) \leq \frac{H_n^2}{|\mu|} + 2(Adv_{\mathring{A}}^{\omega}(p_t)) \tag{2}$$

Proof: We deploy three games (denoted by $\dot{G}m_k$, where $k = 0, 1, 2$) to proof the above stated hypothesis. Suppose that $Suc_{\dot{G}m_k}$ denotes an incident of $\mathring{A}$ winning $\dot{G}m_k$ via the guessing of valid bit ε. Therefore, the advantage or success probability of $\mathring{A}$ winning $\dot{G}m_k$ becomes

$$Adv_{\mathring{A},\dot{G}m_k}^{\lambda_{CSB}}(p_t) = Pr[Suc_{\dot{G}_k}] \tag{3}$$

Thereafter, the following three adversarial games are played by $\mathring{A}$ in an effort to break the negotiated session keys.

$\dot{G}m_0$: In this game, adversary $\mathring{A}$ carries out the actual attack against the proposed protocol λ_{CSB}. Initially, $\mathring{A}$ chooses some random bit ε. Based on equation (1),

$$Adv_{\mathring{A}}^{\lambda_{CSB}}(p_t) = |2(Adv_{\mathring{A},\dot{G}m_0}^{\lambda_{CSB}}(p_t) - 1)| \tag{4}$$

$\dot{G}m_1$: The aim of this game is for $\mathring{A}$ to eavesdrop the communication channel. To accomplish this objective, $\mathring{A}$ carries out the *Execute* () query to intercept messages $AM_1 = \{TID_{SP}, A_1, Z_1, T_4\}$, $AM_2 = \{TID_{SP}^*, A_2, Z_2, T_6\}$, $KM_1 = \{TID_{SM}, B_1, C_{CS}, Z_3, PID_{CS}^*, T_8\}$ and $^{KM2\,=\,\{}TID_{SM}^*, C_{SM}, B_2, Z_4, T_{10}\}$. Next, $\mathring{A}$ performs *Reveal* () and *Test* () queries with the aim of establishing whether the derived session keys are valid or just some stochastic parameters. However, the derivation of these keys requires a combination of long terms as well as short term security tokens. Due to the difficulties of compromising these tokens using the eavesdropped messages AM_1, AM_2, KM_1 and $^{KM2,\text{ the probability that}}$ $\mathring{A}$ has in successfully winning $\dot{G}m_1$ remain the same as that of $\dot{G}m_0$. Therefore,

$$Adv^{\lambda_{CSB}}_{\mathring{A},\dot{G}m_1}(p_t) = Adv^{\lambda_{CSB}}_{\mathring{A},\dot{G}m_0}(p_t) \tag{5}$$

$\dot{G}m_2$: The ultimate goal of this game is to perform some active attack on λ_{CSB}. To attain this goal, three queries and an attempt to solve ω are carried out. The executed queries include *Corrupt* (SP_j), *Hash* () and *Corrupt* (SM_j). The assumption made is that $\mathring{A}$ has already eavesdropped all the exchanged messages, including AM_1, AM_2, KM_1 and ^{KM2}. At the start, $\mathring{A}$ tries to compute $\phi_{SM} = h\ (A_3||Z_1||T_4||T_6)$ and $\phi_{SC} = h\ (B_3||\ f\ (PID_{SM}, PID_{CS})||C_{SM}||C_{CS})$. However, this requires that $\mathring{A}$ correctly derives $A_1 = G.h\ (\mathbf{R}_1||TID_{SP}||PID_{SP}||SK_{SP}||T_4)$, $A_2 = G.h\ (R_2||TID_{SM}||PID_{SM}||SK_{SM}||T_6)$, $B_1 = G.h\ (\mathbf{R}_3||SK_{CS}||PID_{CS}||T_8)$ and $B_2 = G.h\ (\mathbf{R}_4||SK_{SM}||PID_{SM}||T_{10})$ among other tokens. Evidently, each of these parameters by the collision-resistant one-way hashing function h (.) and hence attacker $\mathring{A}$ needs to solve ω in polynomial time p_t. As already demonstrated, the success probability of $\mathring{A}$ in solving ω in polynomial time p_t is $Adv^{\omega}_{\mathring{A}}(p_t)$. It is also clear that these parameters also incorporate time-stamps, short term secrets (such as random nonces) and long term secrets (such as private keys). To check for collisions in the message digests incorporated in the eavesdropped messages (AM_1, AM_2, KM_1 and ^{KM2}), adversary $\mathring{A}$ executes the *Hash* () query. However, since the chosen h (.) is collision-resistant, the success of this query is negligible. As such, the exclusion of these four queries renders $\dot{G}m_2$ and $\dot{G}m_1$ indistinguishable. To find the hash collision, the birthday paradox is applied, yielding the following:

$$Adv^{\lambda_{CSB}}_{\mathring{A},\dot{G}m_1}(p_t) - Adv^{\lambda_{CSB}}_{\mathring{A},\dot{G}m_2}(p_t) \leq \frac{H_n^2}{2|\mu|} + Adv^{\omega}_{\mathring{A}}(p_t) \tag{6}$$

Upon adversarial execution of all these three games, $\mathring{A}$ finally attempts to guess the correct bit ε so as to win the game. Therefore,

$$Adv^{\lambda_{CSB}}_{\mathring{A},\dot{G}m_2}(p_t) = \frac{1}{2} \tag{7}$$

Based on the semantic security definition of the proposed protocol in equation (4),

$$\tfrac{1}{2}Adv^{\lambda_{CSB}}_{\mathring{A}}(p_t) = |(Adv^{\lambda_{CSB}}_{\mathring{A},\dot{G}m_0}(p_t) - \tfrac{1}{2})| \tag{8}$$

Using the triangular inequality, equations (5), (6) and (7) on equation (8) yields the following:

$$\begin{aligned} \frac{1}{2}Adv^{\lambda_{CSB}}_{\mathring{A}}(p_t) &= \left|Adv^{\lambda_{CSB}}_{\mathring{A},\dot{G}m_0}(p_t) - Adv^{\lambda_{CSB}}_{\mathring{A},\dot{G}m_2}(p_t)\right| \\ &= \left|Adv^{\lambda_{CSB}}_{\mathring{A},\dot{G}m_1}(p_t) - Adv^{\lambda_{CSB}}_{\mathring{A},\dot{G}m_2}(p_t)\right| \\ &\leq \frac{H_n^2}{2|\mu|} + Adv^{\omega}_{\mathring{A}}(p_t) \end{aligned} \tag{9}$$

Multiplying the left hand side (LHS) and right hand side (RHS) by 2 yields the following:

$$Adv^{\lambda_{CSB}}_{\mathring{A}}(p_t) \leq \frac{H_n^2}{|\mu|} + 2(Adv^{\omega}_{\mathring{A}}(p_t)) \tag{10}$$

Since both $\frac{H_n^2}{|\mu|}$ and $Adv_{\mathring{A}}^{\omega}(p_t)$ are both infinitesimal, it follows that $Adv_{\mathring{A}}^{\lambda_{CSB}}(p_t)$ is also infinitesimal in polynomial time p_t. This effectively completes the proof of *Hypothesis 1*.

4.2 Informal security analysis

In this sub-section, we formulate and proof a number of theorems with the aim of demonstrating that our protocol is secure under all the adversarial capabilities in the Canetti–Krawczyk threat model.

Theorem 1: Ephemeral secret leakage attacks are prevented

Proof: During the mutual authentication between SM_j and SP_j, the SM_j derives the session key as $\phi_{SM} = h\ (A_3||Z_1||T_4||T_6)$ that it shares with SP_j. Here, $A_3 = A_1.h\ (R_2||TID_{SM}||PID_{SM}||SK_{SM}||T_6)$ and $Z_1 = h\ (\mathbf{R}_1||TID_{SP}||SK_{SP}||) + h(PK_{SP}||PK_{SM}||PK_{CS}||T_4) * SK_{SP}$ (mod q). Similarly, SP_j computes the session key as $\phi_{SP} = h\ (A_4||Z_1||T_4||T_6)$, where $A_4 = A_2.h\ (R_1||TID_{SP}||PID_{SP}||SK_{SP}||T_4)$ and $Z_1 = h\ (\mathbf{R}_1||TID_{SP}||SK_{SP}||) + h(PK_{SP}||PK_{SM}||PK_{CS}||T_4) * SK_{SP}$ (mod q). The derivation of both A_3 and A_4 incorporates short term secrets such as random nonces R_2 and R_1 respectively. In addition, long term secrets such as private keys (SK_{SM} and SK_{SP}) for SM_j and SP_j are incorporated. As such, the adversary $\mathring{A}$ can only derive valid session keys when in possession of both long term and short term secrets (ephemerals) of SM_j and SP_j. Since authentication messages $AM_1 = \{TID_{SP}, A_{1,} Z_{1,} T_4\}$ and $AM_2 = \{TID_{SP}{}^*, A_2, Z_2, T_6\}$ exchanged during mutual authentication do not contain these parameters in plaintext, $\mathring{A}$ cannot access them. As such, adversarial derivation of valid session keys flops.

Theorem 2: Backward and forward key secrecy is preserved

Proof: In the proposed protocol, four session keys are derived. During $SM_j \leftrightarrow SP_j$ authentication, SM_j derives the session key as $\phi_{SM} = h\ (A_3||Z_1||T_4||T_6)$ while SP_j computes the session key as $\phi_{SP} = h\ (A_4||Z_1||T_4||T_6)$. Similarly, during $CS_i \leftrightarrow SM_j$, the CS_i derives session key $\phi_{SC} = h\ (B_3||\ f\ (PID_{SM}, PID_{CS})||C_{SM}||C_{CS})$ while the SM_j calculates session key $\phi_{CM} = h\ (B_4||f\ (PID_{CS}, PID_{SM})||C_{SM}\ ||C_{CS})$. Here, $A_3 = A_1.h\ (R_2||TID_{SM}||PID_{SM}||SK_{SM}||T_6)$, $A_1 = G.h\ (\mathbf{R}_1||TID_{SP}||PID_{SP}||SK_{SP}||T_4)$, $Z_1 = h\ (\mathbf{R}_1||TID_{SP}||SK_{SP}||) + h(PK_{SP}||PK_{SM}||PK_{CS}||T_4) * SK_{SP}$ (mod q), $A_4 = A_2.h\ (R_1||TID_{SP}||PID_{SP}||SK_{SP}||T_4)$, $A_2 = G.h\ (R_2||TID_{SM}||PID_{SM}||SK_{SM}||T_6)$, $B_3 = B_1.\ \mathrm{h}\ (\mathbf{R}_4||SK_{SM}||PID_{SM}||T_{10})$, $C_{SM} = \eta_2 + h(PID_{SM}||ID_{RA}||P_{SM}||PK_{RA}) * MK_{RA}$ (mod q), $C_{CS} = \eta_1 + h(PID_{CS}||ID_{RA}||P_{CS}||PK_{RA}) * MK_{RA}$ (mod q) and $B_4 = B_2.\ h\ (\mathbf{R}_3||SK_{CS}||PID_{CS}||T_8)$. Evidently, all these session keys incorporate random nonces. These random nonces are independently derived by each of the communication entities and are never shared in plaintext over public channels. As such, even if $\mathring{A}$ captures the current session keys, adversarial derivation of session keys for past and subsequent communication session based on these keys will fail.

Theorem 3: Eavesdropping and session hijacking attacks are thwarted

Proof: Suppose that $\mathring{A}$ is interested in hijacking the communication session so as to convince unsuspecting network entities that they are communicating with legitimate entity. Therefore, an attempt is made to eavesdrop the communication channel for ephemerals that may facilitate the derivation of valid session keys. During $SM_j \leftrightarrow SP_j$ authentication, messages $AM_1 = \{TID_{SP}, A_{1,} Z_{1,} T_4\}$ and $AM_2 = \{TID_{SP}{}^*, A_2, Z_2, T_6\}$ are exchanged. Here, $A_1 = G.h\ (\mathbf{R}_1||TID_{SP}||PID_{SP}||SK_{SP}||T_4)$, $Z_1 = h\ (\mathbf{R}_1||TID_{SP}||SK_{SP}||) + h(PK_{SP}||PK_{SM}||PK_{CS}||T_4) * SK_{SP}$ (mod q), $A_2 = G.h\ (R_2||TID_{SM}||PID_{SM}||SK_{SM}||T_6)$ and $Z_2 = h\ (R_2||TID_{SM}||PID_{SM}||SK_{SM}||T_6) + h(\phi_{SM}||PK_{SP}||PK_{CS}||T_6) * SK_{SM}$ (mod q). Next, adversarial derivation of $\phi_{SM} = h\ (A_3||Z_1||T_4||T_6)$ and $\phi_{SP} = h\ (A_4||Z_1||T_4||T_6)$ is attempted. Although timestamp T_4 and T_6 as well as signature Z_1 may be obtained from the eavesdropped messages, the attacker still needs parameters A_3 and A_4 to successfully derive the much needed session keys ϕ_{SM} and ϕ_{SP}. While A_3 is independently calculated at the SM_j, A_4 is independently derived at the SP_j. Since these two values are never transmitted in messages AM_1 and AM_2 and hence cannot be eavesdropped, these attacks fail. Similarly, messages $KM_1 = \{TID_{SM}, B_1, C_{CS}, Z_3, PID_{CS}{}^*, T_8\}$ and $KM_2 = \{TID_{SM}{}^*, C_{SM}, B_2, Z_4, T_{10}\}$ are exchanged during $CS_i \leftrightarrow SM_j$ authentication. Here, $B_1 = G.h\ (\mathbf{R}_3||SK_{CS}||PID_{CS}||T_8)$, $C_{CS} = \eta_1 + h(PID_{CS}||ID_{RA}||P_{CS}||PK_{RA}) * MK_{RA}$ (mod q), $Z_3 = h\ (\mathbf{R}_3||SK_{CS}||PID_{CS}||T_8) + h\ (PK_{CS}||C_{CS}||PID_{CS}{}^*||TID_{SM}) * SK_{CS}$ (mod q), $PID_{CS}{}^* = PID_{CS}{}^{\oplus h\,(PID_{SM}||ID_{RA}||T_8),}$ C_{SM}

$= \eta_2 + h(PID_{SM}||ID_{RA}||P_{SM}||PK_{RA})*MK_{RA} \pmod q$, $B_2 = G.h\,(\mathbf{R}_4||SK_{SM}||PID_{SM}||T_{10})$ and $\mathrm{Z}_4 = h\,(\mathbf{R}_4||SK_{SM}||PID_{SM}||T_{10}) + h\,(PK_{SM}||\mathrm{C}_{SM}||ID_{RA}||\phi_{SC})*SK_{SM} \pmod q$. To derive session keys $\phi_{CM} = h\,(B_4||f\,(PID_{CS}, PID_{SM})||\mathrm{C}_{SM}\,||\mathrm{C}_{CS})$ and $\phi_{SC} = h\,(B_3||\,f\,(PID_{SM}, PID_{CS})||\mathrm{C}_{SM}||\mathrm{C}_{CS})$, $\mathring{A}$ still needs B_4, PID_{SM} and B_3. Whereas B_4 is derived at the CS_i, PID_{SM} is generated at the RA while B_3 is calculated at the SM_j. Once again, these two attacks fail since these parameters cannot be eavesdropped from the exchanged messages.

Theorem 4: Our scheme offers anonymity and untraceability

Proof: The aim of the adversary here is to listen to the communication channel with the aim of associating the communication sessions to particular network entities. As already demonstrated, messages AM_1, AM_2, KM_1 and KM_2 are exchanged over the public channels. Here, $AM_1 = \{TID_{SP}, A_1, \mathrm{Z}_1, T_4\}$, $AM_2 = \{TID_{SP}^*, A_2, \mathrm{Z}_2, T_6\}$, $KM_1 = \{TID_{SM}, B_1, \mathrm{C}_{CS}, \mathrm{Z}_3, PID_{CS}^*, T_8\}$ and $KM_2 = \{TID_{SM}^*, \mathrm{C}_{SM}, B_2, \mathrm{Z}_4, T_{10}\}$. Evidently, real identities of the communicating entities are not included in these messages. As such, only transient identities for SP_j (TID_{SP}) and SM_j (TID_{SM}) as well as the pseudo-identity of CS_i (PID_{CS}^*) can be deciphered. Therefore, these messages cannot be linked to any communicating entities. Random nonces are incorporated in all these messages since $A_1 = G.h\,(\mathbf{R}_1||TID_{SP}||PID_{SP}||SK_{SP}||T_4)$, $\mathrm{Z}_1 = h\,(\mathbf{R}_1||TID_{SP}||SK_{SP}||) + h(PK_{SP}||PK_{SM}||PK_{CS}||T_4)*SK_{SP} \pmod q$, $A_2 = G.h\,(\mathrm{R}_2||TID_{SM}||PID_{SM}||SK_{SM}||T_6)$, $\mathrm{Z}_2 = h\,(\mathrm{R}_2||TID_{SM}||PID_{SM}||SK_{SM}||T_6) + h(\phi_{SM}||PK_{SP}||PK_{CS}||T_6)*\,SK_{SM} \pmod q$, $B_1 = G.h\,(\mathbf{R}_3||SK_{CS}||PID_{CS}||T_8)$ and $B_2 = G.h\,(\mathbf{R}_4||SK_{SM}||PID_{SM}||T_{10})$. In addition, timestamps are also part of all the exchanged messages. As such, all the exchanged messages are always unique for each session and hence cannot be easily associated to the communicating parties.

Theorem 5: Known session-specific temporary information (KSSTI) are prevented

Proof: In our scheme, the session keys are computed using a number of short term and long term keys. These session keys include $\phi_{SM} = h\,(A_3||\mathrm{Z}_1||T_4||T_6)$, $\phi_{SP} = h\,(A_4||\mathrm{Z}_1||T_4||T_6)$, $\phi_{CM} = h\,(B_4||f\,(PID_{CS}, PID_{SM})||\mathrm{C}_{SM}\,||\mathrm{C}_{CS})$ and $\phi_{SC} = h\,(B_3||\,f\,(PID_{SM}, PID_{CS})||\mathrm{C}_{SM}||\mathrm{C}_{CS})$. Here, $A_3 = A_1.h\,(\mathrm{R}_2||TID_{SM}||PID_{SM}||SK_{SM}||T_6)$, $A_1 = G.h\,(\mathbf{R}_1||TID_{SP}||PID_{SP}||SK_{SP}||T_4)$, $\mathrm{Z}_1 = h\,(\mathbf{R}_1||TID_{SP}||SK_{SP}||) + h(PK_{SP}||PK_{SM}||PK_{CS}||T_4)*SK_{SP} \pmod q$, $A_4 = A_2.h\,(\mathrm{R}_1||TID_{SP}||PID_{SP}||SK_{SP}||T_4)$, $A_2 = G.h\,(\mathrm{R}_2||TID_{SM}||PID_{SM}||SK_{SM}||T_6)$, $B_3 = B_1.h\,(\mathbf{R}_4||SK_{SM}||PID_{SM}||T_{10})$, $\mathrm{C}_{SM} = \eta_2 + h(PID_{SM}||ID_{RA}||P_{SM}||PK_{RA})*MK_{RA} \pmod q$, $\mathrm{C}_{CS} = \eta_1 + h(PID_{CS}||ID_{RA}||P_{CS}||PK_{RA})*MK_{RA} \pmod q$ and $B_4 = B_2.h\,(\mathbf{R}_3||SK_{CS}||PID_{CS}||T_8)$. The short term keys are exampled by random nonces such as $\mathbf{R}_1$, R_2, $\mathbf{R}_3$ and $\mathbf{R}_4$. On the other hand, long term keys include secret keys such as SK_{SM}, SK_{SP}, PK_{SP}, PK_{SM}, PK_{CS}, PK_{RA} and MK_{RA}. As such, the loss of session specific ephemerals such as short term keys does not enable the attacker to compromise the session keys.

Theorem 6: Our protocol is resilient against message replay attacks

Proof: To prevent this attack, timestamps and random nonces are incorporated in all the exchanged messages during the mutual authentication phase. For instance, messages $AM_1 = \{TID_{SP}, A_1, \mathrm{Z}_1, T_4\}$, $AM_2 = \{TID_{SP}^*, A_2, \mathrm{Z}_2, T_6\}$, $KM_1 = \{TID_{SM}, B_1, \mathrm{C}_{CS}, \mathrm{Z}_3, PID_{CS}^*, T_8\}$ and $KM_2 = \{TID_{SM}^*, \mathrm{C}_{SM}, B_2, \mathrm{Z}_4, T_{10}\}$ all contain timestamps. On the other hand, ephemerals $A_1 = G.h\,(\mathbf{R}_1||TID_{SP}||PID_{SP}||SK_{SP}||T_4)$, $\mathrm{Z}_1 = h\,(\mathbf{R}_1||TID_{SP}||SK_{SP}||) + h(PK_{SP}||PK_{SM}||PK_{CS}||T_4)*SK_{SP} \pmod q$, $A_2 = G.h\,(\mathrm{R}_2||TID_{SM}||PID_{SM}||SK_{SM}||T_6)$, $\mathrm{Z}_2 = h\,(\mathrm{R}_2||TID_{SM}||PID_{SM}||SK_{SM}||T_6) + h(\phi_{SM}||PK_{SP}||PK_{CS}||T_6)*\,SK_{SM} \pmod q$, $B_1 = G.h\,(\mathbf{R}_3||SK_{CS}||PID_{CS}\,||T_8)$ and $B_2 = G.h\,(\mathbf{R}_4||SK_{SM}||PID_{SM}||T_{10})$ all incorporate random nonces in their derivations. Any replays of old messages will be easily detected by the timestamp checks and the sessions will only continue provided that $|T_5 - T_4| < \Delta T$, $|T_7 - T_6| < \Delta T$, $|T_9 - T_8| < \Delta T$ and $|T_{11} - T_{10}| < \Delta T$. Otherwise, the sessions will be aborted in all the instances.

Theorem 7: Strong mutual authentication is achieved

Proof: In our scheme, all the exchanged messages after the registration phase are mutually verified by the receivers. For instance, on receiving $AM_1 = \{TID_{SP}, A_1, \mathrm{Z}_1, T_4\}$, SM_j validates it by checking if $|T_5 - T_4| < \Delta T$ and $\mathrm{Z}_1.G \overset{?}{=} A_1 + h(PK_{SP}||PK_{SM}||PK_{CS}||T_4)*PK_{SP}$. On the other hand, upon receiving message $AM_2 = \{TID_{SP}^*, A_2, \mathrm{Z}_2, T_6\}$, the SP_j checks if $|T_7 - T_6| < \Delta T$ and $\mathrm{Z}_2.G \overset{?}{=} A_2 + h(\phi_{SP}||PK_{SP}||PK_{CS}||T_6)$. Similarly, after getting message $KM_1 = \{TID_{SM}, B_1, \mathrm{C}_{CS}, \mathrm{Z}_3, PID_{CS}^*, T_8\}$, SM_j

confirms whether $|T_9 - T_8| < \Delta T$, $C_{CS}.G \overset{?}{=} P_{CS} + h(PID_{CS}||ID_{RA}||P_{CS}||PK_{RA}) * PK_{RA}$ and $Z_3.G \overset{?}{=} B_1 + h\,(PK_{CS}||C_{CS}||PID_{CS}{}^{*}||TID_{SM}) * PK_{CS}$. On the other hand, upon receiving message ${}^{KM2\,=\,\{}TID_{SM}{}^{*}$, C_{SM}, B_2, Z_4, $T_{10}{}^{\}, \text{ the}}$ CS_i checks if $|T_{11} - T_{10}| < \Delta T$, $C_{SM}.G \overset{?}{=} P_{SM} + h(PID_{SM}||ID_{RA}||P_{SM}||PK_{RA}) * PK_{RA}$ and $Z_4.G \overset{?}{=} B_2 + h\,(PK_{SM}||C_{SM}||ID_{RA}||\phi_{CM}) * PK_{SM}$. In all these verification instances, the sessions are terminated upon checks failure.

Theorem 8: Session keys are negotiated
Proof: Immediately after successful mutual authentication, all the interacting parties derive the shared session keys for traffic protection. For instance, after authenticating SP_j, the SM_j derives session key as $\phi_{SM} = h\,(A_3||Z_1||T_4||T_6)$. On the other hand, after verifying SM_j, the SP_j computes the session keys as $\phi_{SP} = h\,(A_4||Z_1||T_4||T_6)$. Similarly, session key $\phi_{SC} = h\,(B_3||\,f\,(PID_{SM}, PID_{CS})||C_{SM}||C_{CS})$ is calculated by SM_j upon validation of the CS_i. In the same manner, session key $\phi_{CM} = h\,(B_4||f\,(PID_{CS}, PID_{SM})||C_{SM}\,||C_{CS})$ is derived by the CS_i upon verification of SM_j.

Theorem 9: All the blocks are validated before addition to the blockchain
Proof: In the proposed protocol, three-level validation is executed on all the blocks before their addition to the blockchain. Suppose that a verifier γ is interested in verifying block β_n stored in a given blockchain. To accomplish this, γ derives R^{*} on all enciphered transactions in β_n. In addition, it computes the $H_C{}^{*}$ on β_n. Thereafter, it verifies whether $R^{*} \overset{?}{=} R$ as well as $H_C{}^{*} \overset{?}{=} H_C$. Basically, block β_n is rejected when these two validations flop. However, if the verifications are successful, γ proceeds to validate signature $E_{sig_{\beta_n}}$ on these transactions using E_{ver}. Since β_n incorporates H_P (hash value of the preceding block), it is infeasible for $\mathring{A}$ to modify or corrupt the information stored in β_n.

Theorem 10: Forgery and MitM attacks are prevented
Proof: The aim of these attacks is for $\mathring{A}$ to intercept exchanged messages and modify them to fool other network entities. Suppose that $\mathring{A}$ has captured authentication message $AM_1 = \{TID_{SP}, A_{1,}\ Z_{1,}\ T_4\}$ and wants to generate bogus message $AM_1{}^{\mathring{A}}$. Here, $A_1 = G.h\,(\mathbf{R}_1||TID_{SP}||PID_{SP}||SK_{SP}||T_4)$, $Z_1 = h\,(\mathbf{R}_1||TID_{SP}||SK_{SP}||) + h(PK_{SP}||PK_{SM}||PK_{CS}||T_4) * SK_{SP}\ (\text{mod } q)$. Evidently, $\mathring{A}$ requires private tokens such as SK_{SP} and PID_{SP} as well as timestamp T_4 to derive parameters $A_1{}^{\mathring{A}}$ and $Z_1{}^{\mathring{A}}$. Suppose that $\mathring{A}$ is interested in the derivation of message $AM_2 = \{TID_{SP}{}^{*}, A_2, Z_2, T_6\}$, where $A_2 = G.h\,(R_2||TID_{SM}||PID_{SM}||SK_{SM}||T_6)$ and $Z_2 = h\,(R_2||TID_{SM}||PID_{SM}||SK_{SM}||T_6) + h(\phi_{SM}||PK_{SP}||PK_{CS}||T_6) * SK_{SM}\ (\text{mod } q)$. Clearly, this requires timestamp T_6 and secret tokens SK_{SM} and PID_{SM}. Similarly, $KM_1 = \{TID_{SM}, B_1, C_{CS}, Z_3, PID_{CS}{}^{*}, T_8\}$ and $KM_2 = \{TID_{SM}{}^{*}, C_{SM}, B_2, Z_4, T_{10}\}$ derivation requires secrets SK_{CS}, PID_{CS}, SK_{SM}, PID_{SM} as well as timestamps T_8 and T_{10}. This is because $B_1 = G.h\,(\mathbf{R}_3||SK_{CS}||PID_{CS}||T_8)$, $C_{CS} = \eta_1 + h(PID_{CS}||ID_{RA}||P_{CS}||PK_{RA}) * MK_{RA}\ (\text{mod } q)$, $Z_3 = h\,(\mathbf{R}_3||SK_{CS}||PID_{CS}||T_8) + h\,(PK_{CS}||C_{CS}||PID_{CS}{}^{*}||TID_{SM}) * SK_{CS}\ (\text{mod } q)$, $PID_{CS}{}^{*} = PID_{CS}{}^{\oplus h\,(}PID_{SM}||ID_{RA}||T_8{}^{),}$ $C_{SM} = \eta_2 + h(PID_{SM}||ID_{RA}||P_{SM}||PK_{RA}) * MK_{RA}\ (\text{mod } q)$, $B_2 = G.h\,(\mathbf{R}_4||SK_{SM}||PID_{SM}||T_{10})$ and $Z_4 = h\,(\mathbf{R}_4||SK_{SM}||PID_{SM}||T_{10}) + h\,(PK_{SM}||C_{SM}||ID_{RA}||\phi_{SC}) * SK_{SM}\ (\text{mod } q)$. As such, forgery and MitM attacks are thwarted.

Theorem 11: This protocol is robust against privileged insider attacks
Proof: Suppose that the RA wants to obtain secret values for the SM_j, SP_j and CS_i. However, none of these devices generate and submit their secret parameters to the RA. On the contrary, it is the RA that creates these security tokens including secret keys. However, the RA erases these secret keys upon sending registration messages to the recipients. For instance, after sending registration message $R_1 = \{ID_{RA}, C_{CS}, PID_{CS}, f\,(PID_{CS}, d)\}$ to CS_i over secure channels, RA erases random secret key η_1 used to derive some of these parameters. Similarly, after sending registration message $R_2 = \{TID_{SM}, PID_{SM}\}$ to CS_i over secure channels, the RA erases random secret key η_2. Therefore, privileged insiders at the RA are unable to access these secret values that may enable them derive ephemerals for the network entities.

Theorem 12: Physical and side-channeling attacks are prevented
Proof: The assumption made in this attack is that $\mathring{A}$ can physically capture SM_j as well as any other smart device within the smart grid system. In our protocol, the SM_j store parameter set $\{(TID_{SM}, PID_{SM}), C_{SM}, f\,(PID_{SM}, d), ID_{RA}, (SK_{SM},$

$PK_{SM})\}$ during the registration phase. Therefore, $\mathring{A}$ may opt to use power analysis to retrieve these parameters. Next, an attempt is made to utilize these parameters to derive session keys $\phi_{SM} = h\,(A_3||Z_1||T_4||T_6)$ and $\phi_{SC} = h\,(B_3||\,f\,(PID_{SM}, PID_{CS})||C_{SM}||C_{CS})$. Here, $A_3 = A_1.h\,(R_2||TID_{SM}||PID_{SM}||SK_{SM}||T_6)$, $Z_1 = h\,(\mathbf{R}_1||TID_{SP}||SK_{SP}||) + h(PK_{SP}||PK_{SM}||PK_{CS}||T_4) * SK_{SP}$ (mod q), $B_3 = B_1.h\,(\mathbf{R}_4||SK_{SM}||PID_{SM}||T_{10})$, $C_{SM} = \eta_2 + h(PID_{SM}||ID_{RA}||P_{SM}||PK_{RA}) * MK_{RA}$ (mod q), $C_{CS} = \eta_1 + h(PID_{CS}||ID_{RA}||P_{CS}||PK_{RA}) * MK_{RA}$ (mod q). Evidently, $\mathring{A}$ still requires random nonces $\mathbf{R}_1$,R_2 and $\mathbf{R}_4$, timestamps T_4, T_6, and T_{10}, certificate C_{CS}, identity ID_{RA}, long term key MK_{RA} and ephemerals such as TID_{SP}, SK_{SP}, B_1, η_1 and η_2. Therefore, the communication process is still secure in the face of these two attacks.

Theorem 13: This protocol can withstand impersonation attacks

Proof: Suppose that adversary $\mathring{A}$ is interested in masquerading as legitimate SP_j with the intention of generating authentication message $AM_1 = \{TID_{SP}, A_1, Z_1, T_4\}$. Here, $A_1 = G.h\,(\mathbf{R}_1||TID_{SP}||PID_{SP}||SK_{SP}||T_4)$ and $Z_1 = h\,(\mathbf{R}_1||TID_{SP}||SK_{SP}||) + h(PK_{SP}||PK_{SM}||PK_{CS}||T_4) * SK_{SP}$ (mod q). Let us assume that $\mathring{A}$ has generated bogus timestamp T_4^* and random secret $\mathbf{R}_1^*$ and hence wants to compute legitimate A_1^* and signature Z_1^*. However, devoid of valid secret parameters SK_{SP} and PID_{SP}, it is infeasible for $\mathring{A}$ to succeed in these derivations. Therefore, the construction of message $AM_1 = \{TID_{SP}, A_1, Z_1, T_4\}$ flops. Suppose that $\mathring{A}$ wants to masquerade as smart meter SM_j by attempting to generate message $AM_2 = \{TID_{SP}^*, A_2, Z_2, T_6\}$. Here, $A_2 = G.h\,(R_2||TID_{SM}||PID_{SM}||SK_{SM}||T_6)$ and $Z_2 = h\,(R_2||TID_{SM}||PID_{SM}||SK_{SM}||T_6) + h(\phi_{SM}||PK_{SP}||PK_{CS}||T_6) * SK_{SM}$ (mod q). Once again, this impersonation will flop if $\mathring{A}$ cannot access secret parameters PID_{SM} and SK_{SM}.

Theorem 14: This protocol eliminates key escrow issues

Proof: During the registration phase, the SM_j store parameter set $\{(TID_{SM}, PID_{SM}), C_{SM}, f\,(PID_{SM}, d), ID_{RA}, (SK_{SM}, PK_{SM})\}$. On the other hand, the RA secretly stores MK_{RA} as well as parameter set $\{(TID_{SP}, PID_{SP}), (SK_{SP}, PK_{SP})\}$. Similarly, CS_i stores parameter set $\{ID_{RA}, C_{CS}, PID_{CS}, f(PID_{CS}, d), (SK_{CS}, PK_{CS})\}$. During key management, mutual authentication and key negotiation phases, our scheme does not need any verifier tables. Instead, all the required parameters are independently derived and validated by the SM_j, CS_i and SP_j.

5. Performance evaluation

In authentication protocols, computation costs, supported security features and communication costs are the most widely deployed performance metrics during their performance evaluations. Therefore, these three metrics are deployed in this section to appraise the proposed protocol. In addition, we describe the blockchain implementation of our protocol. Moreover, comparative evaluations of this scheme are provided against other related protocols as described in the subsections below.

5.1 Computation complexities

In this sub-section, the computation overheads are derived for both SP_j and SM_j authentication. The SP_j and CS_i experimentations are executed on a machine running Windows 10 Pro 64 bit operating system on Intel(R) Core i5-2310M processor, installed with 2 GB of RAM, and with 3 GHz Clock frequency. On the other hand, the SM_j experimentations are run on Raspberry Pi-3 quad-core, installed with a 1.2 GHz CPU and 1GB of RAM. To execute the various cryptographic primitives, the MIRACL Cryptographic library is deployed. Under these specifications, the notations and execution durations for the various cryptographic primitives are presented in Table 4.

During the $SM_j \leftrightarrow SP_j$ authentication, $11T_H$, $8T_{ECM}$ and $2T_{ECA}$ operations are executed. Here, the SM_j executes $5\ T_H + 3T_{ECM} + T_{ECA}$ while the SP_j executes $6\ T_H + 5T_{ECM} + T_{ECA}$ operations. On the other hand, using the cryptographic run-times in Table 4 above, the computation complexity of the proposed protocol is detailed in Table 5. In addition, the computation complexities of other related schemes is also elaborated.

Table 4. Execution time for various cryptographic operations.

Cryptographic primitive	Notation	SP_j/CS_i (ms)	SM_j (ms)
One-way hashing operation	T_H	0.0046	0.0406
Elliptic curve point multiplication	T_{ECM}	1.8212	6.9780
Elliptic curve point addition	T_{ECA}	0.0075	0.1436
Bilinear pairing	T_{BP}	8.4681	51.6620
Modular exponentiation	T_E	0.0862	0.5310
Modular multiplication	T_M	0.0628	0.4768
Symmetric encryption/decryption	T_{SE}	0.0035	0.0957
Map to point hash	T_{MH}	13.206	78.2710

https://doi.org/10.1371/journal.pone.0318182.t004

Table 5. Computation complexities comparisons.

Scheme	SM_j	SP_j/ Utility control	Total costs (ms)
Tsai & Lo [21]	$4T_{ECM} + 5T_H + T_E + T_{ECA}$	$3\,T_{ECM} + 5T_H + T_E + 2T_{BP} + T_{ECA}$	51.3061
Saxena et al. [40]	$4T_{ECM} + 2T_H + T_{SE} + T_{BP}$	$4T_{ECM} + 2T_H$	87.0449
Mahmood et al. [47]	$3T_{ECM} + 5T_H + T_{SE} + T_{BP} + 2T_{MH}$	$3T_{ECM} + 5T_H + T_{SE} + T_{BP} + 2T_{MH} + T_E$	269.8931
Odelu et al. [55]	$3T_{ECM} + 6T_H + T_E + 3T_{ECA}$	$2T_{ECM} + 6T_H + T_E + 2T_{BP} + 3T_{ECA}$	42.8543
He et al. [56]	$4T_{ECM} + 5T_H + T_{ECA}$	$6T_{ECM} + 6T_H + 2T_{ECA}$	39.2284
Khan et al. [57]	$4T_{ECM} + 10T_H + 11\,T_{SE}$	$4T_{ECM} + 9T_H + 11\,T_{SE}$	36.7354
Proposed	$3T_{ECM} + 5\,T_H + T_{ECA}$	$5T_{ECM} + 6\,T_H + T_{ECA}$	30.4217

https://doi.org/10.1371/journal.pone.0318182.t005

As shown in Fig 9, the protocol in [47] incurs the highest computation complexity of 269.8931ms. These high complexities can be explained by the high number of bilinear pairing operations and point multiplications that are executed in this scheme.

This is followed by the schemes in [19,40,55–57] with computation complexities of 87.0449 ms, 51.3061 ms, 42.8543 ms, 39.2284 ms and 36.7354 ms respectively. On the other hand, the proposed protocol incurs the least computation complexity of only 30.4217 ms. This is because our protocol majorly executes one-way hashing operations and a few point multiplication operations.

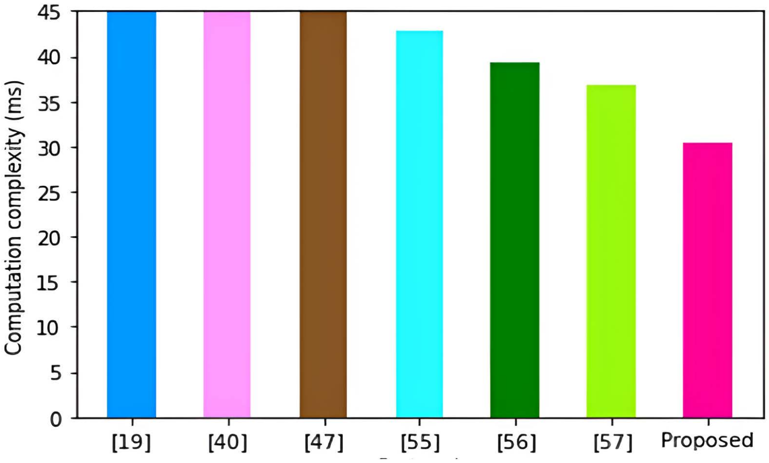


Fig 9. Computation complexities.

https://doi.org/10.1371/journal.pone.0318182.g009

5.2 Communication complexities

In this section, the number and size of the messages exchanged during the mutual authentication phase, as well as the key management phase are taken into consideration. For mutual authentication, messages $AM_1 = \{TID_{SP}, A_{1,} Z_{1,} T_4\}$ and $AM_2 = \{TID_{SP}{}^{*}, A_2, Z_2, T_6\}$ are exchanged. Here, $A_1 = G.h\ (\mathbf{R}_1||TID_{SP}||PID_{SP}||SK_{SP}||T_4)$, $Z_1 = h\ (\mathbf{R}_1||TID_{SP}||SK_{SP}||) + h(PK_{SP}||PK_{SM}||PK_{CS}||T_4) * SK_{SP}$ (mod q), $A_2 = G.h\ (R_2||TID_{SM}||PID_{SM}||SK_{SM}||T_6)$ and $Z_2 = h\ (R_2||TID_{SM}||PID_{SM}||SK_{SM}||T_6) + h(\phi_{SM}||PK_{SP}||PK_{CS}||T_6) * SK_{SM}$ (mod q). Using the values in [6], Table 6 presents the sizes of the various parameters used in the proposed protocol as well as in other related schemes.

Using the values in Table 6 above, the derivation of the communication complexities of our protocol is detailed in Table 7 below.

Based on the derivations in Table 7 above, the communication complexity of the proposed protocol is 1344 bits. Table 8 presents the communication complexities of other related schemes.

As shown in Fig 10, the protocol in [21] incurs the highest communication complexity of 3424 bits. This is followed by the schemes in [40,55–57], the proposed protocol and [47] with communication complexities of 3136 bits, 2627 bits, 1920 bits, 1632 bits, 1344 bits and 1340 bits respectively.

Table 6. Parameter sizes.

Operation	Size (bits)
Timestamp	32
Nonces	160
Temporary identifies	160
Public & secret keys	320
Certificate	160
Elliptic curve point	320
Signature	160
One way hashing	160

https://doi.org/10.1371/journal.pone.0318182.t007

Table 7. Derivation of communication complexity.

Message	Size (bits)
$AM_1 = \{TID_{SP}, A_{1,} Z_{1,} T_4\}$ $TID_{SP} = Z_1 = 160;\ A_1=320;\ T_4 = 32$	672
$AM_2 = \{TID_{SP}{}^{*}, A_2, Z_2, T_6\}$ $TID_{SP}{}^{*}= Z_2 = 160;\ A_2=320;\ T_6 = 32$	672
Total	1344

https://doi.org/10.1371/journal.pone.0318182.t008

Table 8. Communication complexities comparisons.

Scheme	Messages exchanged	Size (bits)
Tsai & Lo [21]	3	3424
Saxena et al. [40]	4	2627
Mahmood et al. [47]	3	1340
Odelu et al. [55]	3	1920
He et al. [56]	3	1632
Khan et al. [57]	2	3136
Proposed	2	1344

https://doi.org/10.1371/journal.pone.0318182.t009

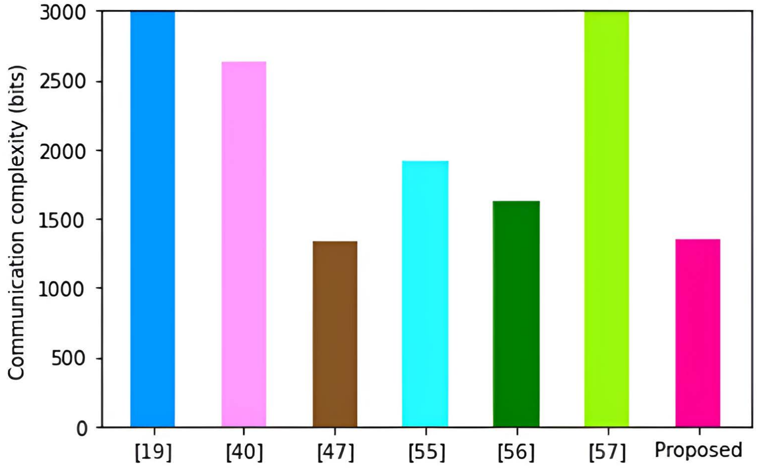


Fig 10. Communication complexities.

https://doi.org/10.1371/journal.pone.0318182.g010

Although the protocol in [47] incurs the lowest communication costs, it has not been evaluated against threats such as side-channeling, physical, eavesdropping and session hijacking.

5.3 Supported security features

In this section, the attacks prevented and other features supported by the proposed protocol are compared with the ones offered by other related schemes. Table 9 presents these comparisons.

As shown in Fig 11, the protocols in [21,55] support only 9 security features while the schemes in [40,56] support 11 features each. On the other hand, the schemes in [47,57] support 12 features and 13 features respectively. It is also evident that the proposed protocol offers support for all the 19 security features.

Using the scheme in [57] as the baseline, our scheme posts a 46.15% improvement in the supported security and privacy characteristics. Similarly, using the protocol in [57] as the baseline, our scheme posts a 17.19% reduction in the computation complexity. Considering smart grid components such as smart gas meters which are resource-limited, the proposed protocol is the most ideal for deployment in this environment.

5.4 Blockchain creation

To simulate the proposed protocol, we let κ_τ denote the transaction number threshold. When the number of transactions is equal to κ_τ, the cloud servers within the network have to vote to select a new C_{BL} amongst themselves in a round-robin manner. This new C_{BL} is now in control of block β_n creation, validation and addition to BC in accordance with Algorithm 1. The Node.js was deployed as the scripting environment and the sizes of the various are presented in Table 10 below.

Each of the generated transaction Ψ_k is enciphered with the help of ECC and hence its output consists of two EC points. Consequently, each enciphered Ψ_k is 640 bits in length and hence the total length of β_n is $1184 + 640\kappa_\tau$ bits. During the PBFT based consensus algorithm voting process, the crash fault tolerance and Byzantine tolerance were 33% while β_n verification lasted between 60–70 transactions/ms. Taking m as the number of peer to peer nodes in the cloud servers, then m^2 messages are exchange in each round of the four main phases of Algorithm 1. As such, the message complexity of this consensus algorithm is $O(m^2)$ and hence the cumulative volume of messages exchanged in this algorithm is given as $4m^2 = O(m^2)$. On the other hand, the computation complexity during the verification of β_n is $12T_{ECM} + 6T_H + 6T_{ECA}$, which is 21.927 ms. Similarly, the key management between any smart grid device and cloud server CS_i is $14T_H + 12T_{ECM} + 4T_{ECA} + 2T_{PL}$, which is 22.6288 ms.

Table 9. Supported features comparisons.

	[21]	[40]	[55]	[56]	[47]	[57]	Proposed
Security features							
Key agreement	√	√	√	√	√	√	√
Mutual authentication	√	√	√	√	√	√	√
Backward key secrecy	√	×	√	√	√	√	√
Forward key secrecy	√	×	√	√	√	√	√
Anonymity	√	√	√	√	√	√	√
Non-traceability	×	×	×	√	√	√	√
Formal verification	√	√	√	√	√	√	√
Resilient against:							
Ephemeral secret leakage	×	×	×	×	×	×	√
Eavesdropping	×	√	×	×	×	×	√
Key escrow	×	×	×	×	×	×	√
Session hijacking	×	×	×	×	×	×	√
KSSTI	×	√	×	×	×	√	√
Replays	√	√	√	√	√	√	√
Forgery	×	√	×	√	×	×	√
MitM	√	√	√	√	√	√	√
Privileged insider	×	×	×	×	√	√	√
Physical	×	√	×	×	√	×	√
Side-channeling	×	×	×	×	×	√	√
Impersonation	√	√	√	√	√	√	√
		√ Supported; × Not supported or not considered					

https://doi.org/10.1371/journal.pone.0318182.t010

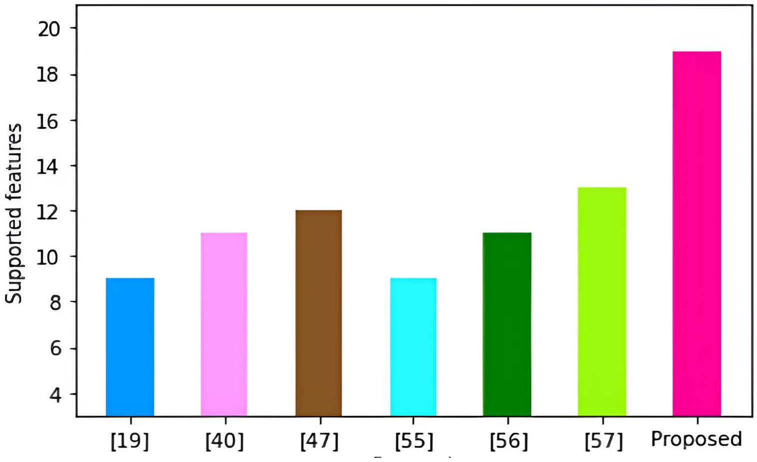


Fig 11. Supported functionalities.

https://doi.org/10.1371/journal.pone.0318182.g011

We then investigate the effect of increasing the number of mined blocks β_{ns} on the computation complexity of the consensus algorithm. In this simulation, the number of transactions κ_{τ} per β_{n_i} is kept constant. The results obtained are shown in Fig 12 below.

Table 10. Block β_n components size.

Component	Size (bits)
V_{β_n}	32
H_P	160
R	160
T_{stp}	32
ID_{C_B}	160
PK_{CS}	320
$\{E_{PK_{CS}}(\Psi_i)\,|i = 1, 2, 3, \ldots \kappa_\tau)\}$	640
H_C	160
$E_{sig_{\beta_n}}$	160

https://doi.org/10.1371/journal.pone.0318182.t011

As shown in Fig 12, there is a general increase in computation complexities upon increase in the number of β_{n_i}mined. Next, we investigate the effect of increasing the number of transactions κ_τ per β_{n_i}on the computation complexity of the consensus procedures. Here, we keep the number of mined β_{ns} constant for each chain. The results obtained are presented in Fig 13 below.

It is evident from Fig 13 that as transactions κ_τ in β_{n_i} surge, there is a corresponding increase in the computation complexities of the consensus procedures. Finally, we vary the number of nodes as the number of $\beta_{n_i}$and κ_τ are kept constant. Fig 14 shows the results obtained.

Based on the graph in Fig 14, it is clear that there is an exponential increase in computation complexity of the consensus procedures when the number of nodes is incremented. These increase in computation costs in all these three instances is attributed to the surging processing that must be accomplished during block generation, verification and addition to the blockchain.

6. Conclusion

Security and privacy issues in smart grids are serious challenges, owing to numerous vulnerabilities and threats that lurk in this environment. This has seen the development of numerous schemes to offer protection during message exchange between the smart meters and utility service providers. Majority of these security techniques are based on bilinear pairing

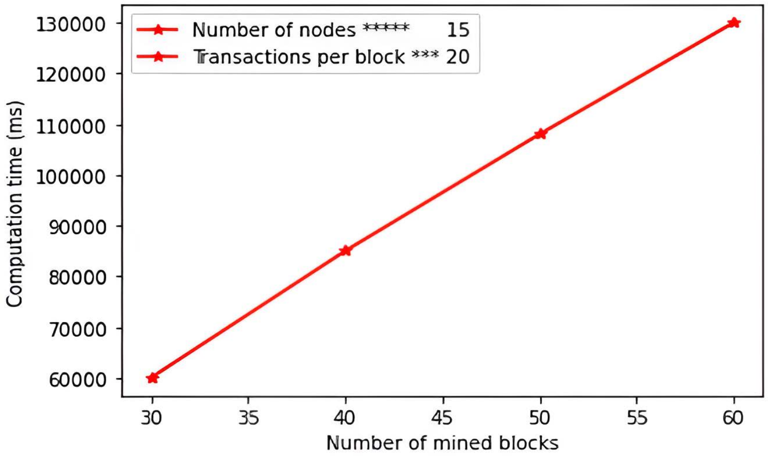


Fig 12. Effect of mined blocks on computation time.

https://doi.org/10.1371/journal.pone.0318182.g012

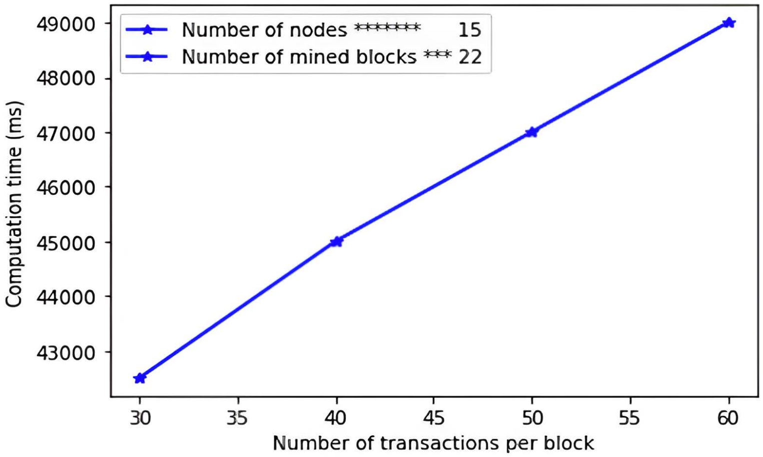


Fig 13. Effect of number of transactions on computation time.

https://doi.org/10.1371/journal.pone.0318182.g013

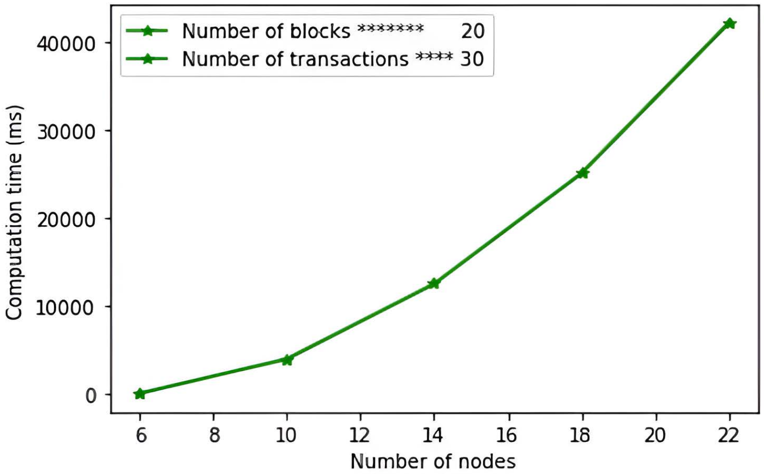


Fig 14. Effect of number of nodes on computation time.

https://doi.org/10.1371/journal.pone.0318182.g014

operations and public key cryptography that are shown to incur heavy computation overheads. The frequent transmission of power consumption reports exposes these reports to security threats and results in high communication complexities. Therefore, an ideal authentication protocol has been developed in this paper to tackle these issues. Extensive security analysis has shown that it is provably secure under the ROR model. In addition, it has been shown to offers salient security and privacy features such as key agreement, mutual authentication, key secrecy, anonymity and untraceability. Moreover, it is resilient against eavesdropping, ephemeral secret leakage, key escrow, session hijacking, KSSTI, replays, forgery, MitM, privileged insider, physical, side-channeling and impersonation attacks. Therefore, it is demonstrated to be robust under all the adversarial capabilities in the Canetti–Krawczyk model. In terms of performance, it requires only 30.4217 ms computation costs, which is the lowest. Since it supports the highest number of security and privacy features, it is the most secure among its peers. Specifically, our protocol achieves a 17.19% reduction in the computation complexity and a 46.15% improvement in the supported security and privacy features. Future work lies in further reduction on the incurred communication complexities so that its efficiency can be reduced further.

Appendix A: Data

$\boldsymbol{ID_{RA}}$	**1bc8e75c8ac2a1d7745bb6d338054b998b6b3f85**
G	5bf0dd40718108af15d664d44e386f0f3e3becb8cb0280155f4023d56a207a4c05b00247ef0d2d1e
MK_{RA}	ac92161fee6190c664294d4f7f011dca76ed323b9bd8117597c04570148dd294c5cb4040ad0db4f8
PK_{RA}	a515e63088c264d95a8c91c74bdb6c077517e2808db16d12d5274af614057f3012701844def11f1b
ID_{CS}	d1887486b3889ffabeae7c56da505a87603a1a74
T_1	c5f2eaa6
PID_{CS}	33b0fb30ae7fdc248e08b3205cc6435c6be9209e
η_1	e6c1371fae23a212ba14dec04dc531a0cb9ed579a5921a4d32d4143e444ed6bdebaea3a23c624729
P_{CS}	d8ade9b705ce815152e2445809ac1ded1f4134d2e62aafaccc96212cc581bc19bc330721ed0da3d2
C_{CS}	e0910483ba4e806d68fdcce29af792d31fdca35f0186652439b1d16a6605bdfd65503e6fc77d3d4c
SK_{CS}	d86a63a66ad68bc2698cf69d02c5a375d6a119edef54b7a93c3737c65f297f45d00f3ec50 cd354cc
PK_{CS}	3 cd47ce42038431eb8d8b1d7e525360d20a4cf093978ba9aef17f974872d5ca5920ddcce9e7133a3
T_2	2eb2459a
ID_{SM}	c652657799528d815d5e0dac700bf18ccb94e6ed
PID_{SM}	33b0fb30ae7fdc248e08b3205cc6435c6be9209e
TID_{SM}	7ad81c129b5fab92a7d280f79fefa72bcff1b0ec
η_2	258eccd718b51e2832b914b00fab01486cb6f23a801b0d3aa1912e7fd3a11938cca5ddf77dc41b10
P_{SM}	f12797dd1b62bc94b0e39800ddd2f45e86a754f8cfa433de08137da0a67333a459b91094c75db808
C_{SM}	fff557f20b23195267717feafcc728345d75b533d649d479278a3c6bc9390bbde754fb8b1908d37b
SK_{SM}	c828b317824e25437ba8f0825cde831ac1698c60077437f8abe834d067b43d42e33c0150b728a810
PK_{SM}	8686d1e33e0f87684345447b8b835e1fed84af045268f4822bbd9d0e81f325c9c120d5268d93f558
T_3	da555fde
ID_{SP}	f83549663bdb92cba628da33b7df6ed2da9ed81a
PID_{SP}	f9b954a9d1822401fbe5444a53c1fd873ff54d10
TID_{SP}	3d1e67fd3c86f793ae97ce05f6bb0baa0d24063b
SK_{SP}	aa3e10c351da5b4e044d520f14102589918b3d22ce0915053294e28df56568f77416143d51a21480
PK_{SP}	44cc49815f5d19fe74716abbdcc261bd60c48e71e64eca9ea6c00b87cbab333fe2dc3df99763a968
$\mathbf{R}_1$	1844197e94aa53c1e665b61d085d8617059b0a8e
T_4	90f98e57
A_1	77ae2baa3c19d3942e2a0f4de4b69bb37ea00c6f
Z_1	d50f3310c3ae1daed346cfcb5b9125a141d6e9b5
T_5	fb3d199c
T_6	844d1512
R_2	a2f43670749697dc6184f10e2cb71a44668c71cf
A_2	1599a3ca42ce2a9ab41f308aa37b5df661610763
A_3	9a11272d41d109bf67c5e3f1e3ea9ab83297caa9
ϕ_{SM}	a490325f380261dac87dfa22e69f1f0be0605aa4
Z_2	f970c169a3ab2bcf44f8d547d3563f6ebf96e016
TID_{SP}^{New}	cbebafaaf7acdd9e8e56207b5ac38d9b079ceb81
TID_{SP}^*	ea1a214e9dac82ca57f43f05dcc12021d320974c
T_7	a288fee5
A_4	71741324c7feb74445c94d73690ffc1da8087cf1
ϕ_{SP}	eea8d4cdb368876e59a7aa11208b275af1cba489
$\mathbf{R}_3$	17efd53cf1ce7f1897050303098b46f21e84bfc4
T_8	e93b54d8
B_1	e1b012d40fe67bc49b2752609f7b62e769077df0

ID_{RA}	**1bc8e75c8ac2a1d7745bb6d338054b998b6b3f85**
$PID_{CS}{}^{*}$	c2777693c08db7b103757b303cca2feb51e23f38
Z_3	87e92259194d88cdb03e3143097d25f11547d11f
T_9	afe6ded0
$\mathbf{R}_4$	548d809e4c0e4050e32c610d2bc6206fa3c82bf9
T_{10}	c289d7be
B_2	aba97cf8dfcdf19692beb3b1a67cdc9ba411af74
B_3	6a37120bbbfba34968c02dcb5712ba3e154beff5
ϕ_{SC}	b9449dd39a19ed0f8850116d65b047abdf67b7cf
Z_4	fc5298c9db0f326ec3d95abc76fe7a3a1e625331
$TID_{SM}{}^{*}$	30120309f51969b22bb81079a668271596cdef99
^{T11}	e76b09ca
B_4	f838a873ff8fcb84c7e01d42a1a72cec63b9d697
ϕ_{CM}	97996f0129d1f255be7924ecf70101114aee320a
$TID_{SM}{}^{New}$	989619de2bdb9f58ca8d9a2ba5711018abec3a0b
$E_{sig_{\beta_n}}$	ffc69a7d5a678427fad836a27d71d34c016e239b
H_P	147362e1d769d0df253f0310128684ae08692e79
H_C	4a24472953754a297d2772efde034898d23dc63e
ID_{C_B}	11977c6bd7abf2c7107b1104c5e8e586e6672d7e
T_{stp}	758e46c1
$SK_{C_{B_i}}$	ffaf8f370d9747d790c9f1abd761e4f4e61710552922b55bea1804f2c9860ca3e9e07fdcf951634d
$PK_{C_{B_i}}$	aae84e9c7bdb6820d49c5e810e6e4e442a8ed6f27275e7c9051e75a871aa2d8d8f79fb73fc929729
$\mathbb{Z}_{V_{rs}}$	5913332a8dab2f31c7b7167a76056c937cd9b85b
$T_{C_{BR}}$	5e28fb8c
ID_{SD_k}	315d3752b6cd74e5a2d24fc46fd5ec67a218ebd1
T_{12}	9de92225
PID_{SD_k}	55ce293c035e44134823fbbffd95bc9fe6088e96
TID_{SD_k}	6b38b67d4487a108cee3aaf1181f372f176ab5bd
SK_{SD_k}	96a51bb63b69ea5cec00acbeb62412f8015320499fb5c5e74f9ac4b5cf2ae4e5199dee2557d9fa7d
PK_{SD_k}	d1bcddcebd75c13df5f8ee363ba2e62a6eeb4ced602957 cd39924738577ce5a3d17b820c1baf7c28

Author contributions

Conceptualization: Ali Hasan Ali.

Data curation: Ali Hasan Ali.

Formal analysis: Vincent Omollo Nyangaresi, Zaid Ameen Abduljabbar, Abdulla J.Y. Aldarwish, Ali Hasan Ali.

Investigation: Zaid Ameen Abduljabbar.

Methodology: Keyan Abdul-Aziz Mutlaq, Zaid Ameen Abduljabbar, Mustafa A. Al Sibahee.

Project administration: Mohd Adib Omar, Mustafa A. Al Sibahee.

Resources: Vincent Omollo Nyangaresi, Junchao Ma, Abdulla J.Y. Aldarwish.

Software: Abdulla J.Y. Aldarwish.

Supervision: Mohd Adib Omar, Junchao Ma.

Validation: Mohd Adib Omar, Ali Hasan Ali.

Visualization: Vincent Omollo Nyangaresi, Junchao Ma, Ali Hasan Ali.

Writing – original draft: Keyan Abdul-Aziz Mutlaq, Junchao Ma, Mustafa A. Al Sibahee.

Writing – review & editing: Keyan Abdul-Aziz Mutlaq, Mustafa A. Al Sibahee.

References

1. Liu S, Zhu Y, Wang R. Pairing-free certificateless blind signature scheme for smart grid. J King Saud Univ - Comput Inf Sci. 2022;34(10):10145–56. https://doi.org/10.1016/j.jksuci.2022.10.012
2. Chaudhry SA, Alhakami H, Baz A, Al-Turjman F. Securing demand response management: a certificate-based access control in smart grid edge computing infrastructure. IEEE Access. 2020;8:101235–43. https://doi.org/10.1109/access.2020.2996093
3. Wang H, Gong Y, Ding Y, Tang S, Wang Y. Privacy-preserving data aggregation with dynamic billing in fog-based smart grid. Appl Sci. 2023;13(2):748. https://doi.org/10.3390/app13020748
4. Nyangaresi VO, Abd-Elnaby M, Eid MMA, Nabih Zaki Rashed A. Trusted authority based session key agreement and authentication algorithm for smart grid networks. Trans Emerging Tel Tech. 2022;33(9). https://doi.org/10.1002/ett.4528
5. Verma N, Purohit S, Narwal B. BASB-SG: a biohashing-based authentication scheme for secure blockchain-enabled smart grids. 2023 5th International Conference on Energy, Power and Environment: Towards Flexible Green Energy Technologies (ICEPE). IEEE; 2023. p. 1–5. https://doi.org/10.1109/ICEPE57949.2023.10201498
6. Sharma G, Joshi AM, Mohanty SP. Fortified-grid: fortifying smart grids through the integration of the trusted platform module in internet of things devices. Information. 2023;14(9):491. https://doi.org/10.3390/info14090491
7. Saxena N, Grijalva S. Dynamic secrets and secret keys based scheme for securing last mile smart grid wireless communication. IEEE Trans Ind Inf. 2017;13(3):1482–91. https://doi.org/10.1109/tii.2016.2610950
8. Gupta R, Tanwar S, Al-Turjman F, Italiya P, Nauman A, Kim SW. Smart contract privacy protection using AI in cyber-physical systems: tools, techniques and challenges. IEEE Access. 2020;8:24746–72. https://doi.org/10.1109/access.2020.2970576
9. Nyangaresi VO. Provably secure authentication protocol for traffic exchanges in unmanned aerial vehicles. High-Confid Comput. 2023;3(4):100154. https://doi.org/10.1016/j.hcc.2023.100154
10. Li H, Li X, Cheng Q. A fine-grained privacy protection data aggregation scheme for outsourcing smart grid. Front Comput Sci. 2022;17(3). https://doi.org/10.1007/s11704-022-2003-y
11. Wang H, Wang L, Wen M, Chen K, Luo Y. A lightweight certificateless aggregate ring signature scheme for privacy protection in smart grids. Wireless Pers Commun. 2022;126(2):1577–99. https://doi.org/10.1007/s11277-022-09809-5
12. Sultan S. Privacy-preserving metering in smart grid for billing, operational metering, and incentive-based schemes: a survey. Comput Secur. 2019;84:148–65. https://doi.org/10.1016/j.cose.2019.03.014
13. Lu M, Zhang Y, Xie Q, Cai X. Vector control of brushless double fed generator based on control winding orientation on smooth switch from stand-alone mode to grid-tied mode. Trait du Signal. 2018;35(1):85–95. https://doi.org/10.3166/ts.35.85-95
14. Wu K, Cheng R, Cui W, Li W. A lightweight SM2-based security authentication scheme for smart grids. Alex Eng J. 2021;60(1):435–46. https://doi.org/10.1016/j.aej.2020.09.008
15. Nyangaresi VO. Privacy preserving three-factor authentication protocol for secure message forwarding in wireless body area networks. Ad Hoc Networks. 2023;142:103117. https://doi.org/10.1016/j.adhoc.2023.103117
16. Al Sibahee MA, Lu S, Abduljabbar ZA, Liu X, Abdalla HB, Hussain MA, et al. Lightweight secure message delivery for E2E S2S communication in the IoT-cloud system. IEEE Access. 2020;8:218331–47. https://doi.org/10.1109/access.2020.3041809
17. Wang J, Wu L, Choo K-KR, He D. Blockchain-based anonymous authentication with key management for smart grid edge computing infrastructure. IEEE Trans Ind Inf. 2020;16(3):1984–92. https://doi.org/10.1109/tii.2019.2936278
18. Sharma R, Joshi AM, Sahu C, Sharma G, Akindeji KT, Sharma S. Semi supervised cyber attack detection system for smart grid. 2022 30th Southern African Universities Power Engineering Conference (SAUPEC). IEEE; 2022. p. 1–5. https://doi.org/10.1109/SAUPEC55179.2022.9730715
19. Abbasinezhad-Mood D, Nikooghadam M. An anonymous ECC-based self-certified key distribution scheme for the smart grid. IEEE Trans Ind Electron. 2018;65(10):7996–8004. https://doi.org/10.1109/tie.2018.2807383
20. Verma G, Gope P, Saxena N, Kumar N. CB-DA: lightweight and escrow-free certificate-based data aggregation for smart grid. IEEE Trans Depend Secure Comput. 2022:1–1. https://doi.org/10.1109/tdsc.2022.3169952
21. Tsai J-L, Lo N-W. Secure anonymous key distribution scheme for smart grid. IEEE Trans Smart Grid. 2015:1–1. https://doi.org/10.1109/tsg.2015.2440658
22. Umran SM, Lu S, Abduljabbar ZA, Lu Z, Feng B, Zheng L. Secure and privacy-preserving data-sharing framework based on blockchain technology for Al-Najaf/Iraq oil refinery. 2022 IEEE Smartworld, Ubiquitous Intelligence & Computing, Scalable Computing & Communications, Digital Twin, Privacy Computing, Metaverse, Autonomous & Trusted Vehicles (SmartWorld/UIC/ScalCom/DigitalTwin/PriComp/Meta). IEEE; 2022. p. 2284–92. Available from: https://doi.org/10.1109/SmartWorld-UIC-ATC-ScalCom-DigitalTwin-PriComp-Metaverse56740.2022.00325
23. Wang W, Huang H, Zhang L, Su C. Secure and efficient mutual authentication protocol for smart grid under blockchain. Peer-to-Peer Netw Appl. 2020;14(5):2681–93. https://doi.org/10.1007/s12083-020-01020-2

24. M. Umran S, Lu S, Ameen Abduljabbar Z, Tang X. A Blockchain-based architecture for securing industrial IoTs data in electric smart grid. Comput Mater Contin. 2023;74(3):5389–416. https://doi.org/10.32604/cmc.2023.034331

25. Zhou Y, Guan Y, Zhang Z, Li F. A blockchain-based access control scheme for smart grids. 2019 International Conference on Networking and Network Applications (NaNA). IEEE; 2019. p. 368–73. https://doi.org/10.1109/NaNA.2019.00070

26. Umran SM, Lu S, Abduljabbar ZA, Zhu J, Wu J. Secure data of industrial internet of things in a cement factory based on a blockchain technology. Appl Sci. 2021;11(14):6376. https://doi.org/10.3390/app11146376

27. Yi X, Lam KY. A new blind ECDSA scheme for bitcoin transaction anonymity. Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security; 2019. p. 613–20. https://doi.org/10.1145/3321705.3329816

28. Zhang W, Lin C, Lyu Z, Cen C, Luo M. An efficient blind signature scheme with untraceability for data privacy in smart grid. Security, Privacy, and Anonymity in Computation, Communication, and Storage: SpaCCS 2020 International Workshops; Nanjing, China; 2020 Dec 18-20, Proceedings 13. Springer International Publishing; 2021. p. 586–97. https://doi.org/10.1007/978-3-030-63176-2_47

29. Huang H, Liu ZY, Tso R. Partially blind ECDSA scheme and its application to bitcoin. 2021 IEEE Conference on Dependable and Secure Computing (DSC). IEEE; 2021 Jan. p. 1–8. https://doi.org/10.1109/DSC49826.2021.9346233

30. Kumar M, Chand S. A pairing-less identity-based blind signature with message recovery scheme for cloud-assisted services. Information Security and Cryptology: 15th International Conference, Inscrypt 2019; Nanjing, China; 2019 Dec 6–8, Revised Selected Papers 15. Springer International Publishing; 2020. p. 419–34. https://doi.org/10.1007/978-3-030-42921-8_24

31. Jiang Y, Deng L, Ning B. Identity-based partially blind signature scheme: cryptanalysis and construction. IEEE Access. 2021;9:78017–24. https://doi.org/10.1109/access.2021.3083529

32. Xu L, Guo Q, Yang T, Sun H. Robust routing optimization for smart grids considering cyber-physical interdependence. IEEE Trans Smart Grid. 2019;10(5):5620–9. https://doi.org/10.1109/tsg.2018.2888629

33. Zhu L, Jiang F, Luo M, Li Q. An efficient identity-based signature protocol over lattices for the smart grid. High-Confid Comput. 2023;3(4):100147. https://doi.org/10.1016/j.hcc.2023.100147

34. Yu H, Wang Z. Certificateless blind signcryption with low complexity. IEEE Access. 2019;7:115181–91. https://doi.org/10.1109/access.2019.2935788

35. Gai K, Wu Y, Zhu L, Xu L, Zhang Y. Permissioned blockchain and edge computing empowered privacy-preserving smart grid networks. IEEE Internet Things J. 2019;6(5):7992–8004. https://doi.org/10.1109/jiot.2019.2904303

36. Kong W, Shen J, Vijayakumar P, Cho Y, Chang V. A practical group blind signature scheme for privacy protection in smart grid. J Parallel Distrib Comput. 2020;136:29–39. https://doi.org/10.1016/j.jpdc.2019.09.016

37. Verma GK, Singh BB, Singh H. Provably secure certificate-based proxy blind signature scheme from pairings. Inf Sci. 2018;468:1–13. https://doi.org/10.1016/j.ins.2018.08.031

38. Nyangaresi VO. Lightweight anonymous authentication protocol for resource-constrained smart home devices based on elliptic curve cryptography. J Syst Architect. 2022;133102763. https://doi.org/10.1016/j.sysarc.2022.102763

39. Chaudhry SA, Shon T, Al-Turjman F, Alsharif MH. Correcting design flaws: an improved and cloud assisted key agreement scheme in cyber physical systems. Comput Commun. 2020;153:527–37. https://doi.org/10.1016/j.comcom.2020.02.025

40. Saxena N, Choi BJ, Lu R. Authentication and authorization scheme for various user roles and devices in smart grid. IEEE Trans Inform Forensic Secur. 2016;11(5):907–21. https://doi.org/10.1109/tifs.2015.2512525

41. Zuo X, Li L, Peng H, Luo S, Yang Y. Privacy-preserving multidimensional data aggregation scheme without trusted authority in smart grid. IEEE Syst J. 2021;15(1):395–406. https://doi.org/10.1109/jsyst.2020.2994363

42. Mohammadali A, Haghighi MS. A privacy-preserving homomorphic scheme with multiple dimensions and fault tolerance for metering data aggregation in smart grid. IEEE Trans Smart Grid. 2021;12(6):5212–20. https://doi.org/10.1109/tsg.2021.3049222

43. Xue K, Zhu B, Yang Q, Wei DSL, Guizani M. An efficient and robust data aggregation scheme without a trusted authority for smart grid. IEEE Internet Things J. 2020;7(3):1949–59. https://doi.org/10.1109/jiot.2019.2961966

44. Nyangaresi VO, Ibrahim A, Abduljabbar ZA, Hussain MA, Al Sibahee MA, Hussien ZA, et al. Provably secure session key agreement protocol for unmanned aerial vehicles packet exchanges. 2021 International Conference on Electrical, Computer and Energy Technologies (ICECET). IEEE; 2021 Dec. p. 1–6. https://doi.org/10.1109/ICECET52533.2021.9698744

45. Abduljabbar ZA, Omollo Nyangaresi V, Al Sibahee MA, Ghrabat MJJ, Ma J, Qays Abduljaleel I, et al. Session-dependent token-based payload enciphering scheme for integrity enhancements in wireless networks. JSAN. 2022;11(3):55. https://doi.org/10.3390/jsan11030055

46. Chen Y, Martínez J-F, Castillejo P, López L. An anonymous authentication and key establish scheme for smart grid: FAuth. Energies. 2017;10(9):1354. https://doi.org/10.3390/en10091354

47. Mahmood K, Li X, Chaudhry SA, Naqvi H, Kumari S, Sangaiah AK, et al. Pairing based anonymous and secure key agreement protocol for smart grid edge computing infrastructure. Future Gener Comput Syst. 2018;88491–500. https://doi.org/10.1016/j.future.2018.06.004

48. Liang XC, Wu TY, Lee YQ, Chen CM, Yeh JH. Cryptanalysis of a pairing-based anonymous key agreement scheme for smart grid. Advances in Intelligent Information Hiding and Multimedia Signal Processing: Proceedings of the 15th International Conference on IIH-MSP in Conjunction with the 12th International Conference on FITAT, Volume 1; Jilin, China; Jul 18–20. Springer Singapore; 2020. p. 125–131. https://doi.org/10.1007/978-981-13-9714-1_14

49. Nyangaresi VO. Terminal independent security token derivation scheme for ultra-dense IoT networks. Array. 2022;15:100210. https://doi.org/10.1016/j.array.2022.100210

50. Gope P, Sikdar B. Lightweight and privacy-friendly spatial data aggregation for secure power supply and demand management in smart grids. IEEE TransInformForensic Secur. 2019;14(6):1554–66. https://doi.org/10.1109/tifs.2018.2881730

51. Hussien ZA, Abdulmalik HA, Hussain MA, Nyangaresi VO, Ma J, Abduljabbar ZA, et al. Lightweight integrity preserving scheme for secure data exchange in cloud-based IoT systems. Appl Scie. 2023;13(2):691. https://doi.org/10.3390/app13020691

52. Zahoor A, Mahmood K, Saleem MA, Badar HMS, Le T-V, Das AK. Lightweight authenticated key agreement protocol for smart power grid systems using PUF. IEEE Open J Commun Soc. 2024;5:3568–80. https://doi.org/10.1109/ojcoms.2024.3409451

53. Zahoor A, Mahmood K, Shamshad S, Saleem MA, Ayub MF, Conti M, et al. An access control scheme in IoT-enabled Smart-Grid systems using blockchain and PUF. Internet Things. 2023;22:100708. https://doi.org/10.1016/j.iot.2023.100708

54. Ayub MF, Li X, Mahmood K, Shamshad S, Saleem MA, Omar M. Secure consumer-centric demand response management in resilient smart grid as industry 5.0 application with blockchain-based authentication. IEEE Trans Consum Electron. 2023.

55. Odelu V, Das AK, Wazid M, Conti M. Provably secure authenticated key agreement scheme for smart grid. IEEE Trans Smart Grid. 2016:1–1. https://doi.org/10.1109/tsg.2016.2602282

56. He D, Wang H, Khan MK, Wang L. Lightweight anonymous key distribution scheme for smart grid using elliptic curve cryptography. IET Commun. 2016;10(14):1795–802. https://doi.org/10.1049/iet-com.2016.0091

57. Chaudhry SA. Correcting "PALK: password-based anonymous lightweight key agreement framework for smart grid". Int J Electr Power Energy Syst. 2021;125:106529. https://doi.org/10.1016/j.ijepes.2020.106529