Lightweight Authentication Protocol in Edge-based Smart Grid Environment

Chien-Ming Chen
Shandong University of Science and Technology https://orcid.org/0000-0002-6502-472X
Lili Chen
Shandong University of Science and Technology
Yanyu Huang
harbin Institute of Technology (Shenzhen)
Sachin Kumar
Ajay Kumar Garg Engineering College
Jimmy Ming-Tai Wu (wmt@wmt35.idv.tw)
https://orcid.org/0000-0003-3740-2102

Research

Keywords: smart grid, edge computing, mutual authentication, network security

Posted Date: January 18th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-53314/v2

Version of Record: A version of this preprint was published at EURASIP Journal on Wireless Communications and Networking on March 29th, 2021. See the published version at https://doi.org/10.1186/s13638-021-01930-6.

RESEARCH

Lightweight Authentication Protocol in Edge-based Smart Grid Environment

Chien-Ming Chen[1], Lili Chen[1], Yanyu Huang[2], Sachin Kumar[3] and Jimmy Ming-Tai Wu[1*]

*Correspondence:
wmt@wmt35.idv.tw
[1]Shandong University of Science and Technology, Shandong, China
Full list of author information is available at the end of the article

Abstract

A smart grid (SG) is an advanced power grid system deployed in a cloud center and smart meters (at the consumer end) that provides higher reliability, better data protection, improved power efficiency, automatic monitoring, and effective management of power consumption. However, an SG also poses certain challenges that need to be addressed. For example, data provided by a smart meter are time-sensitive and cannot handle high latency in an SG. Moreover, a smart meter depends on memory, energy, and other factors. Besides, the security between a cloud center and a smart meter is a critical issue that needs to be resolved. Edge computing, an extension of cloud computing deployed in an edge network between a cloud center and the end devices, is an efficient solution to the aforementioned issues. Therefore, in this study, we propose a secure mutual authentication protocol based on edge computing for use in an SG.

Keywords: smart grid; edge computing; mutual authentication; network security

1 Introduction

A traditional power grid provides four primary operations: power generation, electricity transmission, electricity distribution, and electricity management. Currently, with the rapid development of IoT, the electricity demand has significantly increased; however, the infrastructure used by traditional grids cannot sustain such high electricity demands. Then, the Smart Grid (SG) was developed. SG is an advanced electricity grid that uses a two-way flow of electricity and information, which differs from a traditional electricity infrastructure, providing more efficiency, protected data submissions, and a secure channel between a smart meter (SM) and service provider (SP). The remarkable advantages of SG include monitoring the electricity consumption of the end-users with different approaches in real time[1]. In general, SG infrastructure comprises three components: in-house deployed SMs, SPs in substations, and a control center (CC) in the cloud[2, 3, 4, 5, 6, 7].

Fig. 1 shows a typical infrastructure o SG. SM is used to monitor consumers' energy consumption and send the processed data to SP, which likely contains the consumers' private information. SP offers services and electricity to consumers. CC processes, handles, and manages the sensors, SM, and actuators' resources and stores the data in a cloud computing data center. However, with an increasing number of IoT (end-user) devices, an SG based on cloud computing cannot meet such requirements.

On the other hand, in 2012, Cisco specified certain disadvantages of cloud computing, including a high latency, low mobility support, and low location awareness.

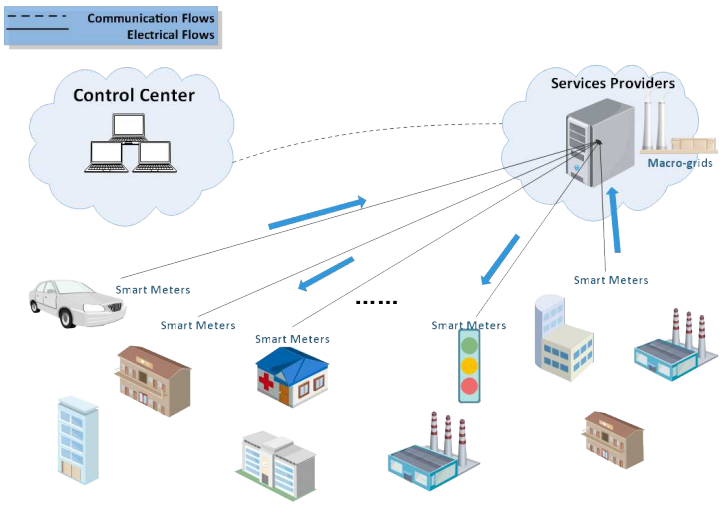

Figure 1 Smart grid infrastructure

As a result, the company proposed the concept of fog computing [8], which is a type of edge computing as previously mentioned, as an extension of cloud computing [9, 10, 11]. An edge layer is employed between the end devices and the cloud center; in addition, each end device is directly connected to the edge nodes, and the edge nodes are interconnected, each linked to the cloud [12]. The edge nodes in edge computing consist of certain devices with limited computation power, such as switches, routers, mobile devices, and idle servers. The main role of an edge node is to collect and process the data from the end devices, issue control commands to the actuators, locally filter the data, and send the remaining data to the cloud center[13, 14]. Edge computing has the following characteristics [15]. First, with the rapid development of mobile devices, it is important for the edge nodes to directly communicate with these devices, such as mobile phones, mobile sensors, and moving cars. Second, the batch process used in cloud computing cannot facilitate real-time interactions; however, with large numbers of edge nodes deployed in distributed locations, edge computing can provide real-time interactions. Third, the edge nodes used in edge computing are distributed in different places. Although the computational abilities of the edge nodes are limited, use of a large number of nodes can solve this problem, for example, by using an SG or a smart vehicle network. Finally, in edge computing, the edge nodes are deployed in different places closer to the end-users. Cloud computing applies a centralized architecture, in sharp contrast to the distributed edge architecture.

As previously mentioned, edge computing is an extension of cloud computing, which provides numerous advantages over the cloud computing infrastructure. In recent years, many researchers have attempted to extend the edge computing infrastructure for applications based on cloud computing[1, 16, 17, 18, 19]. In this study, we extended the infrastructure of edge computing to an SG, which is different from the case of a traditional SG. The advantage of an edge-computing based SG over a

traditional SG includes minimum latency, providing services to resource-constrained devices, reduced stress on the cloud center, and preprocessing of unimportant data.

It is necessary to extend edge computing to an SG; however, the deployment of an SM is unsafe, and the meter needs to be protected by a physical lock to avoid a possible attack by an adversary. An adversary can obtain the data stored in an SM and pose as an SG to communicate with the SPs or consumers. Therefore, secure communication between the SG and SP is extremely important. A key agreement and mutual authentication protocol are efficient solutions to solving this problem. Several studies have also proposed protocols related to SGs[20, 21, 22, 23, 24]. However, a mutual authentication protocol for an SG based on the use of edge computing has yet to be proposed. An SG requires real-time data transmission. Addition of edge nodes to an SG can guarantee a low latency and real-time data response. We therefore propose a protocol based on edge computing for use in such a grid.

Contributions The contribution of this paper is listed as follows.

1 We propose a secure and lightweight key exchange and mutual authentication protocol for an edge-based SG environment. Our design uses one-way hash functions, XOR computations, and an elliptic curve cryptosystem (ECC) instead of another heavy cryptography functions.
2 We provide a formal proof to demonstrate the security of the proposed protocol. Besides, we use Burrows-Abadi-Needham (BAN) logic to guarantee the security of our design. Furthermore, we describe the proposed protocol is secure against various kinds of attacks.
3 we present a performance evaluation/comparison of our protocol.

Organization The remainder of this paper is organized as follows. Section 2 briefly presents the recent studies related to the security of an SG. In section 3, we present an adversary model. The details of the proposed protocol are presented in section 4. To establish the security of the proposed protocol, a security analysis is presented in section 5, concluding with a formal security analysis using formal proof and BAN logic. Section 7 further discuss the proposed protocol is secure against various kinds of attacks. A comparison and performance analysis are provided in Section 7. Finally, some concluding remarks are presented in section 8.

2 Related literature

With the emergence of an SG in 2001, numerous researchers have worked on ensuring security in an SG. Hassan et al. [25] encountered several problems with the use of an SG. For instance, owing to uncertainties in system planning and maintenance, it is challenging to predict real-time system controls. Besides, communication between system operators in a CC is another problem that needs to be considered. Moreover, the lack of predictive control signals for operating the devices and lack of energy storage devices also affect the deployment of SMs. In 2010, Ericsson [26] pointed out that the essential aspects of an SG infrastructure are cybersecurity and power system communication (PSC). Also, information security has become increasingly important because the deployment occurs in an exoteric and integrated energy management system instead of through isolated automation as previously applied. Moreover, with the development of the Internet, attackers can steal data from an SM and a cloud center.

To ensure the information security of an SG, researchers have proposed several security mechanisms. Kim et al. [27] proposed a security mechanism based on an SG according to the security requirements of remote meters using power-line communication (PLC), including authentication and key sharing between devices, as well as revocation management of the remaining devices. However, the SM server used in this mechanism demands authentication of all nodes, which may cause heavy stress on the SM server. When numerous devices are added to an SG environment, the mechanism will become overburdened. For the purpose of efficient resource management and information security, some researchers have begun adding edge computing to an SG; for instance, Zahoor et al. [1] introduced a new SG model based on edge computing for resource management. The proposed model is based on an edge-cloud hierarchical infrastructure to separate the role of the cloud, providing different types of services to consumers. Compared with a traditional SG model, the proposed approach can improve the response time for effective resource utilization and reduce the latency. In 2016, Nazmudeen et al. [28] proposed a distributed data aggregation method based on an edge-computing architecture, limiting the amount of data sent to the centralized storage space, thereby improving the capacity of the PLC without affecting its functionality.

Although edge computing solves the problems inherent to cloud computing, information security, i.e., the security of a transmitted message through an insecure channel, is vital in an SG. A mutual authentication protocol can guarantee the security of inter-communication. In recent years, some mutual authentication protocols have been proposed to ensure the security between parties [29, 30, 31, 32].

Zhang et al.[23] designed an authentication protocol based on elliptic curve cryptography for an SG, which can provide privacy protection. The authors claimed that the protocol has the advantages of identity protection, mutual authentication, and key agreement. However, after analyzing the protocol, we found that it cannot resist an impersonation attack on an SM or SP or a replay attack. Tsai et al.[22] proposed a new anonymous key distribution scheme in an SG environment using identity-based signature schemes and identity-based encryption schemes. The advantages of the scheme include a trusted authority separate from the authentication phases and direct access of an SM to the SP without a trusted authority, which can lower the computation time. However, the proposed protocol is still vulnerable, cannot withstand a privileged insider attack, and provides imperfect forward secrecy.

3 Method

In this section, we first introduce the infrastructure of a smart grid based on edge computing. Then, we describe the adversary model used in this paper.

3.1 Edge computing infrastructure for smart grid

In this study, we proposed a protocol based on edge computing for an SG. Our SG infrastructure based on edge computing is shown in Fig. 2. An edge layer is used to join the infrastructure, acting as an SP in an SG. The cloud is separated to handle data from the edge layer and transmit them to the CC. This infrastructure using different SPs from the macro-grids, which can reduce the burden of the

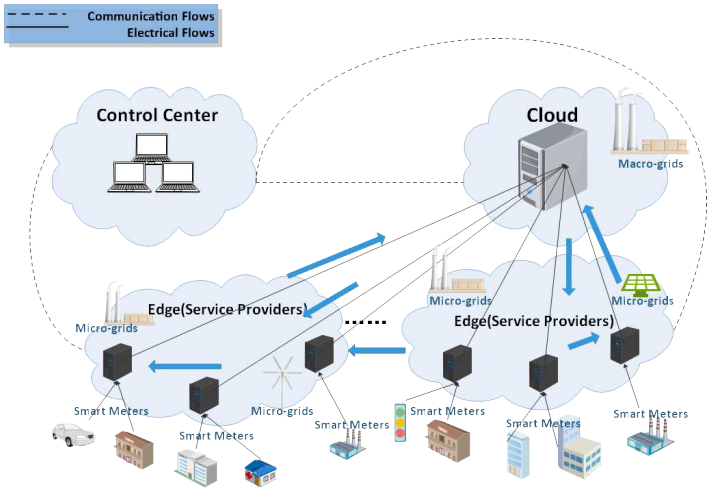

Figure 2 edge based Smart grid infrastructure

cloud, integrates the main capacities of the cloud to communicate with the CC and management control.

In our protocol, edge nodes act as SPs that can quickly process the data and authenticate an SM. Because of the limited computations of the edge nodes and SM, the proposed protocol only uses an ECC and a one-way hash function for encrypting the parameters. Our protocol comprises the following phases.

3.2 Adversary model

Before introducing our proposed protocol, it is important to describe the adversary model applied. A polynomial time adversary *Adv* has full control over the insecure network traffic desires to break the security of the proposed scheme. *Adv* may control limited/completed messages transmitted over an insecure channel, such as intercepting, modifying, and deleting the transmitted message. *Adv* can extract the security parameters stored in a smart card using a power analysis technique. *Adv* can try to obtain sensitive information (e.g., passwords) using off-line password guessing attacks. The gola of *Adv* is to achieve one of the following.

- Compute the session key after a successful run of the authentication scheme.
- Compute the long-term secret key of the server.
- Have the server falsely accept an authentication scheme when they are not communicating with a legitimate entity,

4 Proposed protocol

Herein, we describe the proposed protocol, which consists of three phases, an edge node registration phase, an SM registration phase, and a login and authentication phase. Table 1 summarizes the notations used in our proposed protocol.

4.1 Edge node registration phase

If an edge node ES_j wants to join the system, the edge node registration phase is applied. This phase is shown in fig4 and is described as follows:

Table 1 Notations used in the proposed protocol

Notations	Descriptions
SM	Smart Meter
ES_j	j^{th} edge node
TA	Trusted Authority
$E_p(a,b),p$	Elliptic curve cryptosystem, $E_p : y^3 = x^3 + ax + b(mod\ p)$ is an elliptic curve over prime field F_p, p is a prime, where $x,y,a,b \in F_p$ and $(4a^3 + 27b^2) mod\ p \neq 0$
kG	$kG = G + G + ... + G(ktimes, k \in F_p)$,Scalar multiplication in Elliptic Curve Cryptosystem.
r_i	The register parameters generated by Trusted Authority
Adv	Adversary
s	Secret Key of TA
SK_{ij}, SK_{ji}	Session Key by SMs and edge nodes
n_i, n_j	random number chosen by SMs and edge nodes
$H(\cdot)$	Cryptographic one-way hash function
$\|\|$	Concatenation
$\oplus$	Bitwise XOR operation

(i) ES_j first selects an identity SID_j and transmits $\{SID_j\}$ to TA through a secure channel.

(ii) After receiving the above messages, TA checks the validity of ES_j. Then, TA computes $RSID_j = H(SID_j||s)$, stores $\{SID_j, RSID_j\}$ in the database of TA, and transmits $\{RSID_j\}$ back to ES_j through a secure channel.

(iii) ES_j stores $\{RSID_j\}$ in its database.

4.2 Smart meter registration phase

The SM registration phase (fig3) is executed if an SM registers with TA. We assume that an SM, whose identity is ID_i, wants to join this system, and there are n edge nodes, denoted as $ES_1, \ldots, ES_n$, that have been previously registered. The following steps are conducted.

(i) SM selects identity ID_i and a random number r_i, and then transmits $\{ID_i, r_i\}$ to the trusted authority TA through a secure channel.

(ii) After receiving the message from SM, TA first finds all values of $RSID_j$ stored in the database where $1 \leq j \leq n$. Next, TA computes $A_j = H(ID_i||RSID_j)$, $BAu_j = A_j \oplus H(ID_i||r_i)$, $C_j = H(SID_j||RSID_j)$ and $DAu_j = C_j \oplus H(ID_i||r_i)$, and then transmits $\{SID_j, BAu_j, DAu_j, H(\cdot)\}$ back to SM through a secure channel.

(iii) The SM computes $SID_j^* = SID_j \oplus H(ID_i||r_i)$, $BAu_j^* = BAu_j \oplus H(ID_i||r_i)$, $DAu_j^* = SID_j \oplus BAu_j \oplus DAu_j$ and stores $\{SID_j^*, BAu_j^*, DAu_j^*, H(\cdot)\}$ into memory.

4.3 Login and authentication phase

When a legal SM wants to log in and communicate with ES_j, the SM needs to authenticate ES_j and establish a session key with ES_j using the following steps.

(i) The SM first enters its identity ID_i and r_i, and then computes $SID_j = SID_j^* \oplus H(ID_i||r_i)$, $BAu_j = BAu_j^* \oplus H(ID_i||r_i)$, $DAu_j = SID_j \oplus BAu_j \oplus DAu_j^*$, and $C_j = DAu_j \oplus H(ID_i||r_i)$. Next, SM generates the current timestamp T_i and calculates $E_i = ID_i \oplus H(C_j||T_i)$. In addition, SM then generates a random number n_i, computing $N_i = n_i P$, $A_j = BAu_j \oplus H(ID_i||r_i)$,

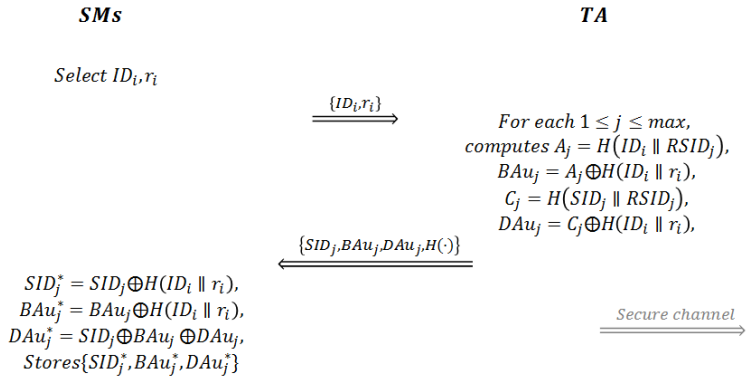

Figure 3 Smart meter registration phase

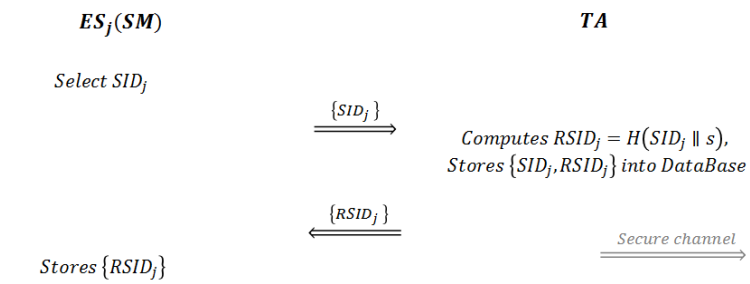

Figure 4 Edge node registration phase

$F_i = H(ID_i||A_j)$, $G_i = F_i \oplus N_i$, and authentication parameter $M_1 = H(ID_i||F_i||N_i||T_i)$, sending $\{E_i, G_i, M_1, T_i\}$ to ES_j through an unsecure channel.

(ii) After receiving the message from SM, ES_j calculates $C_j^{'} = H(SID_j||RSID_j)$, $ID_i^{'} = E_i \oplus H(C_j^{'}||T_i)$, $A_j^{'} = H(ID_i^{'}||RSID_j)$, $F_i^{*} = H(ID_i^{'}||A_j^{'})$, $N_i^{'} = F_i^{'} \oplus G_i$, and $M_1^{'} = H(ID_i^{'}||F_i^{'}||N_i^{'}||T_i)$, and then checks the validity of M_1 to verify if SM is legal. If so, ES_j generates a random number n_j and current timestamp T_j, and then computes $N_j = n_j P$, $M_2 = H(A_j^{'}||C_j^{'}||N_j||T_j)$, $I_j = H(C_j^{'}) \oplus N_i^{'}$, $K_j = I_j \oplus N_j$, and the session key $SK_{ji} = H(ID_i^{'}||A_j^{'}||n_j N_i^{'})$, and transmits the message $\{M_2, K_j, T_j\}$ to SM.

(iii) When SM receives the messages from ES_j, SM first computes $I_j^{'} = H(C_j) \oplus N_i$, $N_j^{'} = I_j^{'} \oplus K_j$, and $M_2^{'} = H(A_j||C_j||N_j^{'}||T_j)$, and then checks the validity of ES_j by checking whether $M_2^{'} = M_2$. If so, SM computes the session key $SK_{ij} = H(ID_i||A_j||n_i N_j^{'})$, which means that SM and ES_j can securely communicate with each other.

The SM login and authentication phase is illustrated in Fig. 5.

5 Security analysis of the proposed protocol

In this section, we first provide a formal proof of the proposed protocol. Then, we further evaluate the security of the proposed protocol with BAN logic.

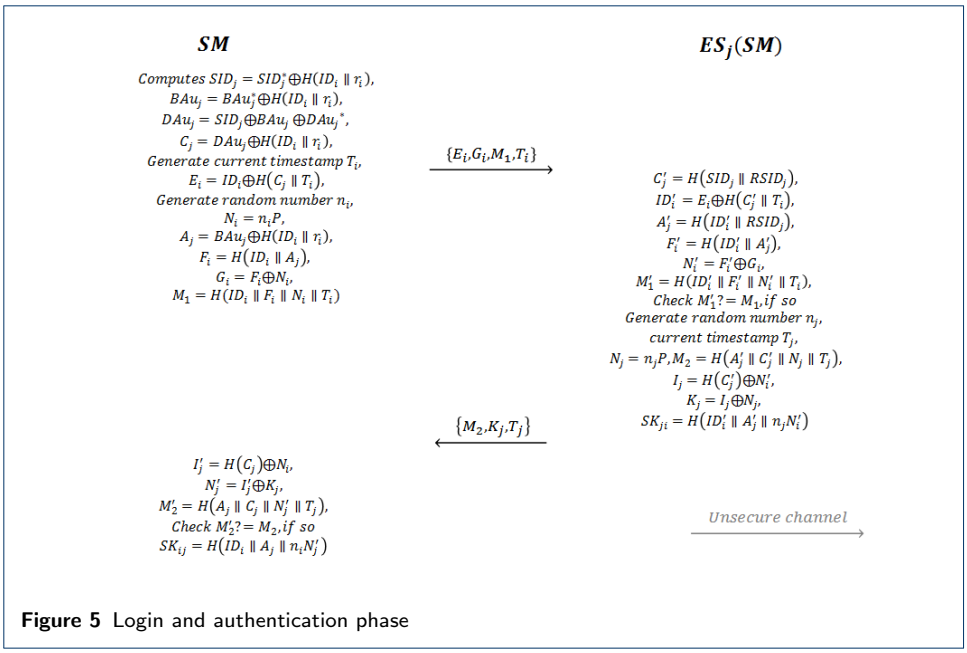

Figure 5 Login and authentication phase

5.1 Formal proof

Here, we prove the security of the proposed protocol under the Real-Or-Random (ROR) model. In the introduction section, we have defined the capabilities of the adversary [2]. Assume that I_{SM}^x, I_{ES}^y, and I_{TA}^z respectively represent the x-th instance of SM_s, the y-th instance of ES_j, and the z-th instance of TA. The adversary $\mathcal{A}$ can initiate the following queries.

$Execute(\mathcal{O})$: $\mathcal{A}$ executes the query and can obtain the messages $\{E_i, G_i, M_1, T_i\}$ and $\{M_2, K_j, T_j\}$, where $\mathcal{O} = \{I_{SM}^x, I_{ES}^y, I_{TA}^z\}$.

$Hash(string)$: $\mathcal{A}$ executes the query and can get the hash value of the input parameter $string$.

$Send(\mathcal{O}, M)$: $\mathcal{A}$ executes the query, sends the message M to $\mathcal{O}$, and can receive the corresponding response.

$Corrupt(\mathcal{O})$: $\mathcal{A}$ executes the query and can obtain a secret value, such as the long-term private key, temporary information, etc.

$Test(\mathcal{O})$: $\mathcal{A}$ executes the query and judges the correctness of the session key by flipping a coin $\mathcal{C}$. If the result is $\mathcal{C} = 1$, $\mathcal{A}$ will receive the correct session key returned; if the result is $\mathcal{C} = 0$, $\mathcal{A}$ will receive a random string.

Definition 1. (Elliptic Curve Discrete Logarithm Problem (ECDLP)). Assuming that $\mathcal{E}$ is an elliptic curve generation group. Given points, P and aP, where P belongs to $\mathcal{E}$ and a belongs to F_p, it is computationally infeasible to obtain a. In polynomial time ξ, the probability of an adversary $\mathcal{A}$ solving this problem is defined as: $Adv_{\mathcal{A}}^{ECDLP}(\xi) = Pr[\mathcal{A}(P, aP) = a : a \in F_p, P \in \mathcal{E}]$. For a sufficiently small η, we have: $Adv_{\mathcal{A}}^{ECDLP}(\xi) < \eta$.

Theorem: Under the ROR model, if $\mathcal{A}$ attempts to initiate some queries in polynomial time, then the advantage that it can break the proposed protocol $\mathcal{P}$ is: $Adv_{\mathcal{A}}^{\mathcal{P}}(\xi) \leq (q_{send} + q_{exe})^2/p + q_{hash}^2/2^{l-1} + q_{send}/2^{l-1} + 2Adv_{\mathcal{A}}^{ECDLP}(\xi)$, where q_{send} represents the number of $Send$ query executed, q_{exe} represents the number of

$Execute$ query executed, q_{hash} represents the number of $Hash$ query executed, and l represents the bits of the hash operation.

Proof: We use the game sequence GM_0 to GM_5 to verify the above theorem. $Succ_{\mathcal{A}}^{GM_n}(\xi)$ is the probability that $\mathcal{A}$ succeeds in the game GM_n. The specific description is as follows.

GM_0: GM_0 represents a real attack, and $\mathcal{A}$ will not initiate any query at this time. Therefore, in GM_0, the probability of $\mathcal{A}$ breaking $\mathcal{P}$ is: $Adv_{\mathcal{A}}^{\mathcal{P}}(\xi) = |2Pr[Succ_{\mathcal{A}}^{GM_0}(\xi)] - 1|$.

GM_1: GM_1 adds $Execute$ query based on GM_0. Therefore, we have: $Pr[Succ_{\mathcal{A}}^{GM_1}(\xi)] = Pr[Succ_{\mathcal{A}}^{GM_0}(\xi)]$.

GM_2: GM_2 adds the $Send$ query based on GM_1. According to Zipf's law [3], we have: $|Pr[Succ_{\mathcal{A}}^{GM_3}(\xi)] - Pr[Succ_{\mathcal{A}}^{GM_2}(\xi)]| \leq q_{send}/2^l$.

GM_3: GM_3 adds $Hash$ query based on GM_2. According to the birthday paradox, we can get that the maximum probability of a hash collision is $q_{hash}^2/2^{l+1}$; the maximum probability of a conflict occurring in the transmitted text is $(q_{send} + q_{exe})^2/2p$ [4][5]. Therefore, we have: $|Pr[Succ_{\mathcal{A}}^{GM_2}(\xi)] - Pr[Succ_{\mathcal{A}}^{GM_1}(\xi)]| \leq (q_{send} + q_{exe})^2/2p + q_{hash}^2/2^{l+1}$.

GM_4: In this game, we consider the security of the session key. Here, we divide the discussion into two situations. The first is to obtain a long-term private key to verify perfect forward security; the second is temporary information leakage to verify whether can resist ephemeral secret leakage attack.

(1) Perfect forward security. $\mathcal{A}$ uses $Corrupt(I_{TA}^z)$ to try to get TA's long-term private key s, or uses $Corrupt(I_{SM}^x)$ or $Corrupt(I_{ES}^y)$ to try to get a secret value in the registration phase.

(2) Ephemeral secret leakage attack. $\mathcal{A}$ uses $Corrupt(I_{SM}^x)$ or $Corrupt(I_{ES}^y)$ to try to obtain temporary information from one party.

In both cases, the ECDLP needs to be solved to compute the session key $SK_j = h(ID_i||A_j||n_jN_i)$ or $SK_i = h(ID_i||A_j||n_iN_j)$. For the first formula $SK_j = h(ID_i||A_j||n_jN_i)$ in the first case, even if $RSID_j = H(SID_j||s), C_j = H(SID_j||RSID_j), ID_i = E_i \oplus H(C_j||T_i), A_j = H(ID_i||RSID_j)$ are calculated by s, the random number n_j is unknown. And through $Corrupt(I_{SM}^x)$ or $Corrupt(I_{ES}^y)$ to get $\{SID_j, BAu_j, DAu_j\}$ or $\{RSID_j\}$, $\mathcal{A}$ cannot get any value in session key; in the second case, even if n_jN_i is calculated by n_j, but the long-term private key s is unknown. Similarly, for the second formula $SK_i = h(ID_i||A_j||n_iN_j)$ is also true. Therefore, we have: $|Pr[Succ_{\mathcal{A}}^{GM_4}(\xi)] - Pr[Succ_{\mathcal{A}}^{GM_3}(\xi)]| \leq q_{send}Adv_{\mathcal{A}}^{ECDLP}(\xi)$.

GM_5: The purpose of this game is to verify the impersonation attack. The difference between GM_5 and GM_4 is that the game is terminated if $\mathcal{A}$ issues $h(ID_i||A_j||n_jN_i)$ query. At this point, the probability of $\mathcal{A}$ guessing the session key is $|Pr[Succ_{\mathcal{A}}^{GM_5}(\xi) - Pr[Succ_{\mathcal{A}}^{GM_4}(\xi)| \leq q_{hash}^2/2^{l+1}$. Since GM_5 is equally successful and unsuccessful, we have: $Pr[Succ_{\mathcal{A}}^{GM_5}(\xi)] = 1/2$.

In summary, we can get the following conclusion:

$1/2Adv_{\mathcal{A}}^{\mathcal{P}}(\xi) = Pr[Succ_{\mathcal{A}}^{GM_0}(\xi)] - 1/2 = Pr[Succ_{\mathcal{A}}^{GM_0}(\xi)] - Pr[Succ_{\mathcal{A}}^{GM_5}(\xi)]$

$= Pr[Succ_{\mathcal{A}}^{GM_1}(\xi)] - Pr[Succ_{\mathcal{A}}^{GM_5}(\xi)]$

$\leq Pr[Succ_{\mathcal{A}}^{GM_5}(\xi)] - Pr[Succ_{\mathcal{A}}^{GM_4}(\xi)] + Pr[Succ_{\mathcal{A}}^{GM_4}(\xi)] - Pr[Succ_{\mathcal{A}}^{GM_3}(\xi)] + Pr[Succ_{\mathcal{A}}^{GM_3}(\xi)] - Pr[Succ_{\mathcal{A}}^{GM_2}(\xi)] + Pr[Succ_{\mathcal{A}}^{GM_2}(\xi)] - Pr[Succ_{\mathcal{A}}^{GM_1}(\xi)]$

$= q_{hash}^2/2^{l+1} + q_{send}Adv_{\mathcal{A}}^{ECDLP}(\xi) + q_{send}/2^l + (q_{send} + q_{exe})^2/2n + q_{hash}^2/2^{l+1}$.

Further, we have $Adv_{\mathcal{A}}^{\mathcal{P}}(\xi) = (q_{send} + q_{exe})^2/n + q_{hash}^2/2^{l-1} + q_{send}/2^{l-1} + 2Adv_{\mathcal{A}}^{ECDLP}(\xi)$.

5.2 Security analysis using BAN logic

In this subsection, we demonstrate the security of our solution during the authentication phase through the BAN logic. BAN logic is widely used for analyzing the security of authentication and key agreement protocols [33, 34, 35].

In this study, the user (SM) and the edge node ES_j authenticate each other and calculate a session key. Below are some of the symbols and rules defined when using the BAN logic.

5.2.1 Notations used in BAN logic

- $P \mid\equiv X$: The principal P believes X, or is entitled to do so. In particular, P may act as though X is true. This construct is central to the logic.
- $P \triangleleft X$: P sees X. Someone sends a message containing X to P, who can read and repeat X (possibly after a decryption).
- $P \mid\sim X$: P once stated X. At some point of time, P sent a message including statement X. It is unknown whether the message was sent long ago or during the current run of the protocol, but it is known that P believed X at that time.
- $P \mid\Longrightarrow X$: P has jurisdiction over X. The principal P is an authority on X and should be trusted in this matter. For example, a server is often trusted to properly generate encryption keys. This may be expressed based on the assumption that the principals believe that the server has jurisdiction over statements regarding the quality of these keys.
- $\sharp(X)$: The formula X is fresh; that is, X has not been sent in a message at any time before the current run of the protocol. This is typically true for a nonce, that is, an expression invented for the purpose of being fresh. A nonce commonly includes a timestamp or number that is used only once.
- $P \overset{K}{\longleftrightarrow} Q$: P and Q may use a shared key K to communicate. Key K is safe in that it will never be discovered by any principal except P or Q, or by a principal trusted by either P or Q.
- $P \overset{X}{\rightleftharpoons} Q$: Formula X is a secret known only to P and Q and possibly to principals trusted by them. Only P and Q may use X to prove their identities to one another. An example of a secret is a password.
- $\{X\}_K$: Formula X is encrypted under key K.
- $\langle X \rangle_Y$: Formula X is combined with formula Y.

5.2.2 BAN logic rules

(i) The message-meaning rule for shared keys is $\frac{P \mid\equiv P \overset{K}{\longleftrightarrow} Q, P \triangleleft \{X\}_K}{P \mid\equiv Q \mid\sim X}$. This indicates that, if P believes that K is a key shared with Q and if P sees X encrypted under K, then P believes that Q once stated X.

(ii) The message-meaning rule for shared secrets: $\frac{P \mid\equiv P \overset{Y}{\rightleftharpoons} Q, P \triangleleft \langle X \rangle_Y}{P \mid\equiv Q \mid\sim X}$. This means that, if P believes that Y is a secret known only to P and Q and P sees X under Y, then P believes that Q once stated X.

(iii) The nonce-verification rule is $\frac{P|\equiv\sharp(X),P|\equiv Q|\sim X}{P|\equiv Q|\equiv X}$. This means that, if P believes that X is fresh and Q once stated X, then P believes that Q believes X.

(iv) The jurisdiction rule is $\frac{P|\equiv Q|\Longrightarrow X,P|\equiv Q|\equiv X}{P|\equiv X}$. This means that, if P believes that Q has jurisdiction over X and believes that Q believes X, then P believes X.

(v) The session key rule is $\frac{P|\equiv\sharp(X),P|\equiv Q|\equiv X}{P|\equiv P\overset{K}{\longleftrightarrow}Q}$. This means that, if P trusts that statement formula X is fresh and P trusts that Q trusts X, which is an essential component of the session key, then P trusts that he or she shares the session key K with Q.

(vi) The freshness rule is $\frac{P|\equiv\sharp(X)}{P|\equiv\sharp(X,Y)}$. This means that, if P believes that X is fresh, then he or she believes the freshness of (X, Y).

(vii) The belief rule is $\frac{P|\equiv X,P|\equiv Y}{P|\equiv(X,Y)}$. This means that, if P believes X and Y, then P believes (X, Y).

5.2.3 Goals

- **G1**: $SM \mid\equiv SM\overset{SK}{\longleftrightarrow}ES_j$.
- **G2**: $ES_j \mid\equiv SM\overset{SK}{\longleftrightarrow}ES_j$.
- **G3**: $SM \mid\equiv ES_j \mid\equiv SM\overset{SK}{\longleftrightarrow}ES_j$.
- **G4**: $ES_j \mid\equiv SM \mid\equiv SM\overset{SK}{\longleftrightarrow}ES_j$.

5.2.4 Idealize the communication messages

- **Meg1**: $SM \rightarrow ES_j$: $\{E_i, G_i, M_1, T_i\}$.
- **Meg2**: $S_j \rightarrow SM$: $\{M_2, K_j, T_j\}$.

5.2.5 Initial state assumptions

- **A1**: $SM \mid\equiv \sharp(n_i)$.
- **A2**: $ES_j \mid\equiv \sharp(n_j)$.
- **A3**: $SM \mid\equiv SM\overset{C_j}{\rightleftharpoons}ES_j$.
- **A4**: $SM \mid\equiv ES_j \mid\Longrightarrow N_j$.
- **A5**: $ES_j \mid\equiv SM\overset{H(C_j\|T_i)}{\rightleftharpoons}ES_j$.
- **A6**: $ES_j \mid\equiv \sharp(ID_i)$.
- **A7**: $ES_j \mid\equiv SM \mid\Longrightarrow ID_i$.
- **A8**: $SM \mid\equiv \sharp(N_j)$.
- **A9**: $ES_j \mid\equiv \sharp(N_i)$.
- **A10**: $ES_j \mid\equiv SM \mid\Longrightarrow N_i$.

5.2.6 Main proofs using BAN rules and assumptions

Based on the BAN logic rules, we demonstrate that the proposed key exchange protocol can use the initial state assumptions to achieve the defined goals. Below are the steps used to prove the BAN logic.

For G1, according to the message Meg2 and using the seeing rule, we obtain **S1**: $SM \triangleleft \{M_2 : \langle A_j, N_j, T_j\rangle_{C_j}; K_j, T_j\}$. Using A3, S1, and the message-meaning rule, we obtain **S2**: $SM \mid\equiv ES_j \mid\sim (A_j, N_j, T_j)$. Using A3 and S2, and applying the freshness and nonce-verification rules, **S3**: $SM \mid\equiv ES_j \mid\equiv (A_j, N_j, T_j)$ is obtained. Applying the belief rule for each component, we obtain **S4**: $SM \mid\equiv ES_j \mid\equiv N_i$.

Using A4, S4, and the jurisdiction rule, **S5**: $SM \mid\equiv N_j$ is obtained. Because $SK = H(ID_i||A_j||n_iN_j)$, we obtain **S6**: $SM \mid\equiv SM \overset{SK}{\longleftrightarrow} ES_j$.

For G2, according to the message Meg1 and using the seeing rule, we obtain **S7**: $ES_j \triangleleft \{E_i : \langle ID_i \rangle_{H(C_j \| T_i)}; G_i, M_1 : \langle F_i, N_i, T_i \rangle_{ID_i}; T_i\}$. Using the seeing rule for each component, we obtain **S8**, i.e., $ES_j \triangleleft \{\langle ID_i \rangle_{H(C_j \| T_i)}\}$, and **S9**, i.e., $ES_j \triangleleft \{\langle F_i, N_i, T_i \rangle_{ID_i}\}$. Using A5, S8, and the message-meaning rule, we obtain **S10**: $ES_j \mid\equiv SM \mid\sim ID_i$. Using A6 and S10 and applying the nonce-verification rule, **S11**: $ES_j \mid\equiv SM \mid\equiv ID_i$ is obtained. Using A7, S11, and the jurisdiction rule, we obtain **S12**: $ES_j \mid\equiv ID_i$. Using A6, S11, and the session key rule, we obtain **S13**: $ES_j \mid\equiv SM \overset{ID_i}{\rightleftharpoons} ES_j$. Using S9, S13, and the message-meaning rule, we obtain **S14**: $ES_j \mid\equiv SM \mid\sim (F_i, N_i, T_i)$. According to A9 and S15 and using the freshness and nonce-verification rules, **S15**: $ES_j \mid\equiv SM \mid\equiv (F_i, N_i, T_i)$ is obtained. Based on the belief rule, we obtain **S16**: $ES_j \mid\equiv SM \mid\equiv N_i$. Using A10, S16, and the jurisdiction rule, we obtain **S17**: $SM \mid\equiv N_i$. Because $A_j = H(ID_i||RSID_j)$ and $SK = H(ID_i||A_j||n_jN_i)$, we obtain **S18**: $ES_j \mid\equiv SM \overset{SK}{\longleftrightarrow} ES_j$.

Applying the belief rule for each component, we obtain **S4**: $SM \mid\equiv ES_j \mid\equiv N_i$. Using A4, S4, and the jurisdiction rule, **S5**: $SM \mid\equiv N_j$ is obtained. Because $SK = H(ID_i||A_j||n_iN_j)$, we obtain **S6**: $SM \mid\equiv SM \overset{SK}{\longleftrightarrow} ES_j$.

For G3, according to S6, A1, and the session key rule, we obtain **S19**: $SM \mid\equiv ES_j \mid\equiv SM \overset{SK}{\longleftrightarrow} ES_j$.

For G4, according to S18, A2, and the session key rule, we obtain **S20**: $ES_j \mid\equiv SM \mid\equiv SM \overset{SK}{\longleftrightarrow} FS_j$.

6 Discussion

Numerous authenticated and key agreement protocols have been proven insecure against the following kinds of attacks [36, 37, 38, 39, 40]. In this section, we further discuss our protocol can resist such attacks. First, we assume that the adversary is represented as *Adv*.

Replay attack A replay attack resends the messages intercepted by *Adv*, which can obtain the messages of $\{E_i, G_i, M_1, T_i\}$ and $\{M_2, K_j, T_j\}$. We can see that they all have a timestamp in every transmitted message, which guarantees the freshness of the messages; timestamp T_i and T_j are both used in a later authentication parameter to check the validity of each other. Therefore, a faked timestamp cannot pass the verification stage. As a result, an adversary cannot replay the messages, and our protocol effectively resists a replay attack.

SMs and edge node impersonation attack

If adversary *Adv* wants to create a login message $\{E_a, G_a, M_{1a}, T_{ai}\}$ or $\{M_{2a}, K_a, T_{aj}\}$ to pose as a legal SM SM or legal ES_j. Taking SM as an example, ES_j is similar to SM. If *Adv* wants to log into ES_j, he or she first needs the parameters of C_j to calculate ID_i; however, without the knowledge of SID_j and $RSID_j$, *Adv* cannot obtain C_j without ID_i, and *Adv* cannot calculate F_i and N_i. Therefore, it is impossible for *Adv* to create a legal login message, and hence, our protocol can resist attacks on SMs and edge node impersonation attacks.

Man-in-middle attack As mentioned previously, *Adv* cannot obtain the messages of both $\{E_a, G_a, M_{1a}, T_{ai}\}$ and $\{M_{2a}, K_a, T_{aj}\}$, and thus *Adv* cannot forge

legal SMs or an edge node. Thus, our proposed protocol is resilient against a man-in-the-middle attack.

Perfect forward secrecy In perfect forward secrecy, the long-term key s indeterminable for Adv, and the messages over an insecure channel and the parameters from the memory of the SMs are revealed to Adv; however, even with these parameters, Adv still cannot expose the session key between SMs and edge nodes. In our proposed protocol, if Adv wants to calculate a session key $SK_{ij} = H(ID_i||A_j||n_iN_j')$, Adv needs to know the parameters, ID_i and A_j, and the random parameters, n_iN_j or n_jN_i, whereas ID_i, A_j, n_iN_j, and n_jN_i are independent of the long-term key and cannot be calculated based on messages from an insecure channel; therefore Adv cannot obtain the parameters used to compute the session key. Hence, our protocol can guarantee perfect forward secrecy.

Ephemeral secret leakage attack As mentioned above, if Adv wants to obtain a session key, he or she needs to first obtain the ephemeral secret n_i, n_j. In an ephemeral secret leakage attack, Adv can obtain the random parameters n_i and n_j; however, if Adv wants to obtain the session key, Adv needs another two parameters ID_i and A_j, which cannot be obtained from an insecure channel or the memory of the SMs. This proves that our proposed protocol is resilient against an ephemeral secret attack.

SM anonymity and untraceability attack In our protocol, the random numbers n_i and n_j and timestamp T_i, T_i are used in the login and authentication phase. Different sessions have different messages; thus, Adv cannot trace the message to focus on specific SMs, and the proposed protocol can have security against an SM anonymity attack. In addition, the identities of the SMs and edge nodes are masked by a random number and timestamp, which is also different. Hence our protocol can resist an untraceability attack.

7 Experimental Result and Comparison

In this section, some protocols related to an SG that do not employ edge computing are listed and compared with our proposed protocol to prove the higher performance of the latter. To objectively analyze the protocols, we used an iPhone 7 for accurately determining the computational costs. In the evaluation based on experimental data, to evaluate the proposed protocol, we use T_h, T_d, T_{pa}, T_{pm}, T_{ae}, T_{ad}, T_{exp}, and TG_e to represent the time required for performing a one-way hash function, a symmetric decryption/encryption operation, an ECC point addition, a point multiplication, asymmetric public key encryption, asymmetric public key decryption, modular exponentiation, and a bilinear pairing operation. The time required for a bitwise XOR computation is negligible, and therefore, we do not consider the XOR computation time. Table 3 lists the computation time for these operations.

The computation times for the related SG protocols and the proposed protocol are presented in Table 3. A bar chart of the computation times is shown in Fig. 6. From the bar chart, it can be concluded that the proposed protocol involves the minimum computation time in all stages of entities. Although the computation time of Zhang et al.'s protocol was approximately 110.77 ms, which is similar to that of the proposed protocol, Zhang et al.'s protocol cannot resist a privileged insider attack or provide perfect forward secrecy. Therefore, taking all requirements into

Table 2 Computation time of protocols

Operation	Expression	Times(ms)
T_h	$h(\cdot)$	0.03
T_d	$E_k(\cdot)/D_k(\cdot)$	0.12
T_{pa}	$a_iP + b_iP$	0.18
T_{pm}	a_iP	20.23
T_{ae}	E_{ae}	34.99
T_{ad}	E_{ad}	34.78
T_{exp}	$g^{a_i} \ mod \ n$	25.27
TG_e	$e(aP, bQ) = e(P, Q)^{ab}$	25.64

Table 3 Comparisons of the security features among different protocols

Related Protocol	Smart meters	Service providers	Trusted Authority	Total costs	Total costs(ms)
Tsai et al.[22]	$4T_{pm}+T_{exp}+5T_h$	$2TG_e + 3T_{pm} + T_{exp} + 5T_h$	-	$7T_{pm} + 2T_{exp} + 2TG_e + 10T_h$	243.73
Zhang et al.[23]	$T_{pm}+T_{ae}+T_d+T_h$	$T_{pm} + T_{ad} + 3T_d + T_h$	-	$2T_{pm} + T_{ae} + T_{ad} + 4T_d + 2T_h$	110.77
K. Mahmood et al.[24]	$4T_{pm} + 4T_h + T_{pa}$	-	$T_{pm} + T_h$	$5T_{pm} + 5T_h + T_{pa}$	101.48
Wang et al.[41]	$TG_e + 3T_{pm} + 7T_h + T_d$	$TG_e + 3T_{pm} + T_{exp} + 7T_h + T_d$	-	$2TG_e + 6T_{pm} + T_{exp} + 14T_h + 2T_d$	198.59
Ghosh et al.[42]	$T_{pm} + 2T_{exp} + TG_e + 5T_h$	$3T_{pm} + T_{exp} + TG_e + 4T_h$	-	$4T_{pm} + 3T_{exp} + 2TG_e + 9T_h$	208.28
Ours	$2T_{pm} + 12T_h$	$2T_{pm} + 8T_h$	$5T_h$	$4T_{pm} + 25T_h$	81.67

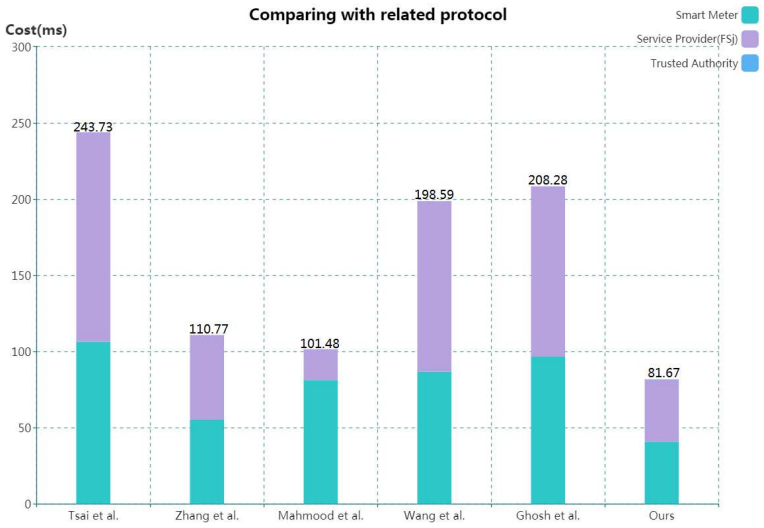

Figure 6 Comparison with related protocol

account, our proposed protocol can provide easy computations and security against various attacks.

8 Conclusion

In this study, we proposed using edge computing to solve the current security issues encountered in SGs. We designed a secure key exchange and mutual authentication protocol based on edge computing for such grids. To verify our proposed protocol's security, we analyzed the protocol using the automatic tool ProVerif and BAN logic to prove that it can resist various types of attacks. We also compared our proposed protocol with other protocols used in SGs without an edge computing infrastructure. We concluded that the computation time of our proposed protocol is lower than that of other protocols and that the proposed protocol is more secure and lightweight.

List of abbreviations
SG: smart grid; SM: smart meter; SP: service provider; CC: control center; ECC: curve cryptosystem;

Availability of data and materials
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Author's contributions
Conceptualization, Chien-Ming Chen and Lili Chen; Formal analysis, Yanyu Huang, Jimmy Ming-Tai Wu; Methodology, Chien-Ming Chen, Lili Chen and Yanyu Huang; Software, Sachin Kumar; Supervision, Chien-Ming Chen, Jimmy Ming-Tai Wu; Validation, Yanyu Huang; Writing, JImmy Ming-Tai Wui.

Acknowledgements
Not applicable.

Funding
Not applicable.

Author details
[1]Shandong University of Science and Technology, Shandong, China. [2]Harbin Institute of Technology (Shenzhen), Shenzhen, China. [3]Ajay Kumar Garg Engineering College, Ghaziabad, India.

References

1. Zahoor, S., Javaid, N., Khan, A., Ruqia, B., Muhammad, F.J., Zahid, M.: A cloud-fog-based smart grid model for efficient resource utilization. In: 2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC), pp. 1154–1160 (2018). IEEE
2. Wu, T.-Y., Lee, C. Yu-Qi, Chien-Ming, , Tian, Y., Al-Nabhan, N.A.: An enhanced pairing-based authentication scheme for smart grid communications. J Ambient Intell Human Comput (2021)
3. Gungor, V.C., Sahin, D., Kocak, T., Ergut, S., Buccella, C., Cecati, C., Hancke, G.P.: Smart grid technologies: Communication technologies and standards. IEEE transactions on Industrial informatics **7**(4), 529–539 (2011)
4. Fang, X., Misra, S., Xue, G., Yang, D.: Smart grid:the new and improved power grid: A survey. IEEE communications surveys & tutorials **14**(4), 944–980 (2011)
5. Amin, S.M., Wollenberg, B.F.: Toward a smart grid: power delivery for the 21st century. IEEE power and energy magazine **3**(5), 34–41 (2005)
6. Ipakchi, A., Albuyeh, F.: Grid of the future. IEEE power and energy magazine **7**(2), 52–62 (2009)
7. Khurana, H., Hadley, M., Lu, N., Frincke, D.A.: Smart-grid security issues. IEEE Security & Privacy **8**(1), 81–85 (2010)
8. Bonomi, F., Milito, R., Zhu, J., Addepalli, S.: Fog computing and its role in the internet of things. In: Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, pp. 13–16 (2012)
9. Wu, T.-Y., Chen, C.-M., Sun, X., Liu, S., Lin, J.C.-W.: A countermeasure to sql injection attack for cloud environment. Wireless Personal Communications **96**(4), 5279–5293 (2017)
10. Xiong, H., Wang, Y., Li, W., Chen, C.-M.: Flexible, efficient, and secure access delegation in cloud computing. ACM Transactions on Management Information Systems (TMIS) **10**(1), 1–20 (2019)
11. Kumari, A., Kumar, V., Abbasi, M.Y., Kumari, S., Chaudhary, P., Chen, C.-M.: Csef: cloud-based secure and efficient framework for smart medical system using ecc. IEEE Access **8**, 107838–107852 (2020)
12. Stojmenovic, I., Wen, S.: The fog computing paradigm: Scenarios and security issues. In: 2014 Federated Conference on Computer Science and Information Systems, pp. 1–8 (2014). IEEE
13. Hu, P., Dhelim, S., Ning, H., Qiu, T.: Survey on fog computing: architecture, key technologies, applications and open issues. Journal of network and computer applications **98**, 27–42 (2017)
14. Hu, P., Ning, H., Qiu, T., Song, H., Wang, Y., Yao, X.: Security and privacy preservation scheme of face identification and resolution framework using fog computing in internet of things. IEEE Internet of Things Journal **4**(5), 1143–1155 (2017)
15. Chen, C.-M., Huang, Y., Wang, K.-H., Kumari, S., Wu, M.-E.: A secure authenticated and key exchange scheme for fog computing. Enterprise Information Systems, 1–16 (2020)
16. Deng, R., Lu, R., Lai, C., Luan, T.H.: Towards power consumption-delay tradeoff by workload allocation in cloud-fog computing. In: 2015 IEEE International Conference on Communications (ICC), pp. 3909–3914 (2015). IEEE
17. Okay, F.Y., Ozdemir, S.: A fog computing based smart grid model. In: 2016 International Symposium on Networks, Computers and Communications (ISNCC), pp. 1–6 (2016). IEEE
18. Hou, X., Li, Y., Chen, M., Wu, D., Jin, D., Chen, S.: Vehicular fog computing: A viewpoint of vehicles as the infrastructures. IEEE Transactions on Vehicular Technology **65**(6), 3860–3873 (2016)
19. Truong, N.B., Lee, G.M., Ghamri-Doudane, Y.: Software defined networking-based vehicular adhoc network with fog computing. In: 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), pp. 1202–1207 (2015). IEEE
20. Fouda, M.M., Fadlullah, Z.M., Kato, N., Lu, R., Shen, X.S.: A lightweight message authentication scheme for smart grid communications. IEEE Transactions on Smart grid **2**(4), 675–685 (2011)
21. Saputro, N., Akkaya, K., Uludag, S.: A survey of routing protocols for smart grid communications. Computer Networks **56**(11), 2742–2771 (2012)
22. Tsai, J.-L., Lo, N.-W.: Secure anonymous key distribution scheme for smart grid. IEEE transactions on smart grid **7**(2), 906–914 (2015)
23. Zhang, L., Tang, S., Luo, H.: Elliptic curve cryptography-based authentication with identity protection for smart grids. PloS one **11**(3), 0151253 (2016)
24. Mahmood, K., Chaudhry, S.A., Naqvi, H., Kumari, S., Li, X., Sangaiah, A.K.: An elliptic curve cryptography based lightweight authentication scheme for smart grid communication. Future Generation Computer Systems **81**, 557–565 (2018)
25. Hassan, R., Radman, G.: Survey on smart grid. In: Proceedings of the IEEE SoutheastCon 2010 (SoutheastCon), pp. 210–213 (2010). IEEE
26. Ericsson, G.N.: Cyber security and power system communication-essential parts of a smart grid infrastructure. IEEE Transactions on Power Delivery **25**(3), 1501–1507 (2010)
27. Kim, S., Kwon, E.Y., Kim, M., Cheon, J.H., Ju, S.-h., Lim, Y.-h., Choi, M.-s.: A secure smart-metering protocol over power-line communication. IEEE Transactions on Power Delivery **26**(4), 2370–2379 (2011)
28. Nazmudeen, M.S.H., Wan, A.T., Buhari, S.M.: Improved throughput for power line communication (plc) for smart meters using fog computing based data aggregation approach. In: 2016 IEEE International Smart Cities Conference (ISC2), pp. 1–4 (2016). IEEE
29. Chien, H.-Y., Chen, C.-H.: Mutual authentication protocol for rfid conforming to epc class 1 generation 2 standards. Computer Standards & Interfaces **29**(2), 254–259 (2007)
30. He, D., Kumar, N., Lee, J.-H., Sherratt, R.S.: Enhanced three-factor security protocol for consumer usb mass storage devices. IEEE Transactions on Consumer Electronics **60**(1), 30–37 (2014)
31. Moosavi, S.R., Gia, T.N., Nigussie, E., Rahmani, A.M., Virtanen, S., Tenhunen, H., Isoaho, J.: End-to-end security scheme for mobility enabled healthcare internet of things. Future Generation Computer Systems **64**, 108–124 (2016)
32. Wazid, M., Das, A.K., Kumar, N., Vasilakos, A.V.: Design of secure key management and user authentication scheme for fog computing services. Future Generation Computer Systems **91**, 475–492 (2019)

33. Chen, C.-M., Xu, L., Wang, K.-H., Liu, S., Wu, T.-Y.: Cryptanalysis and improvements on three-party-authenticated key agreement protocols based on chaotic maps. Journal of Internet Technology **19**(3), 679–687 (2018)
34. Fei, Y., Zhu, H., Vinh, P.C.: Security analysis of the access control solution of ndn using ban logic. Mobile Networks and Applications, 1–12 (2020)
35. Chen, C.-M., Li, C.-T., Liu, S., Wu, T.-Y., Pan, J.-S.: A provable secure private data delegation scheme for mountaineering events in emergency system. Ieee Access **5**, 3410–3422 (2017)
36. Li, C.-T., Lee, C.-C., Weng, C.-Y., Chen, C.-M.: Towards secure authenticating of cache in the reader for rfid-based iot systems. Peer-to-Peer Networking and Applications **11**(1), 198–208 (2018)
37. Kumari, S., Chaudhary, P., Chen, C.-M., Khan, M.K.: Questioning key compromise attack on ostad-sharif et al.'s authentication and session key generation scheme for healthcare applications. IEEE Access **7**, 39717–39720 (2019)
38. Wang, K.-H., Chen, C.-M., Fang, W., Wu, T.-Y.: On the security of a new ultra-lightweight authentication protocol in iot environment for rfid tags. The Journal of Supercomputing **74**(1), 65–70 (2018)
39. Sun, H.-M., Wang, K.-H., Chen, C.-M.: On the security of an efficient time-bound hierarchical key management scheme. IEEE transactions on dependable and secure computing **6**(2), 159–160 (2009)
40. Wu, T.-Y., Lee, Z., Obaidat, M.S., Kumari, S., Kumar, S., Chen, C.-M.: An authenticated key exchange protocol for multi-server architecture in 5g networks. IEEE Access **8**, 28096–28108 (2020)
41. Wang, Y.: Password protected smart card and memory stick authentication against off-line dictionary attacks. In: IFIP International Information Security Conference, pp. 489–500 (2012). Springer
42. Ghosh, D., Li, C., Yang, C.: A lightweight authentication protocol in smart grid. IJ Network Security **20**(3), 414–422 (2018)

Figures

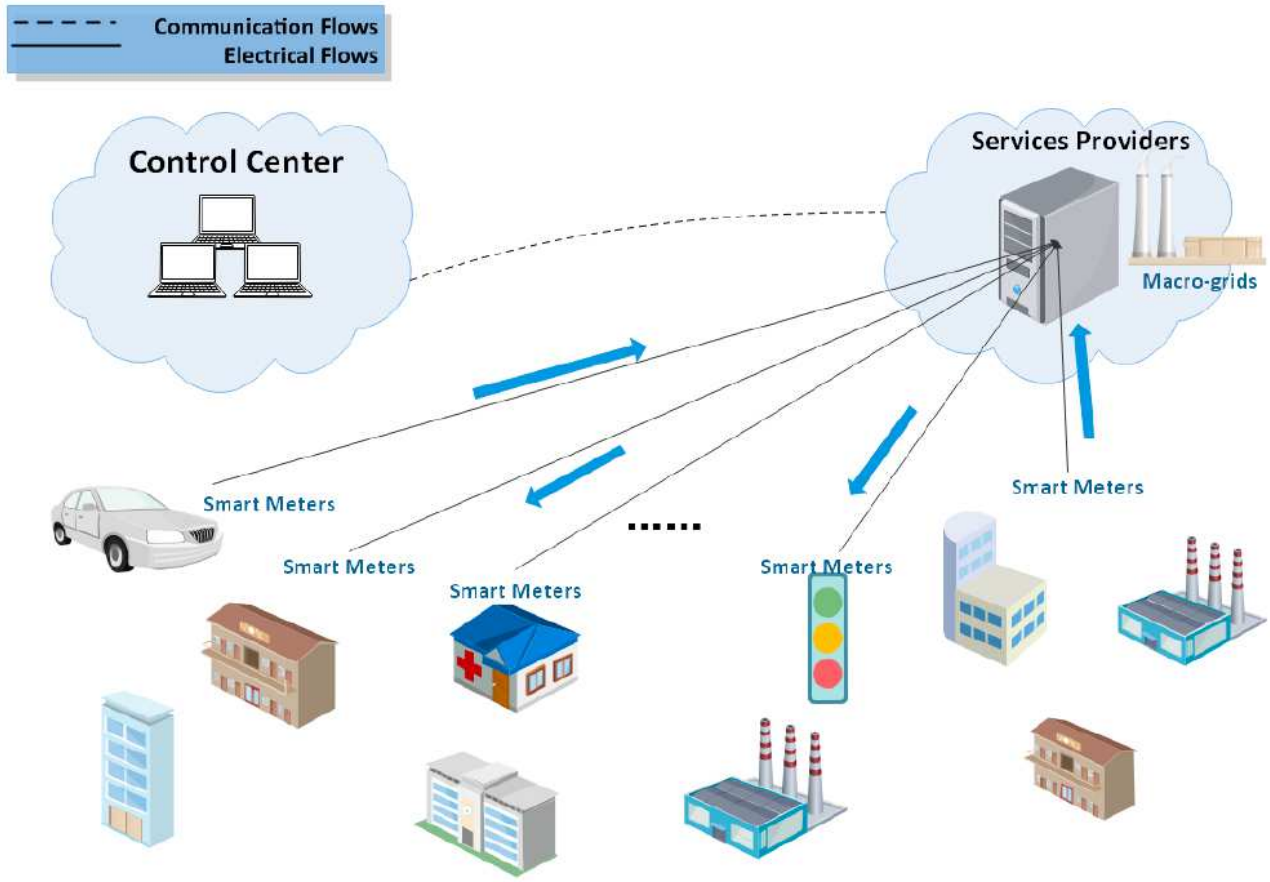

Figure 1

Smart grid infrastructure

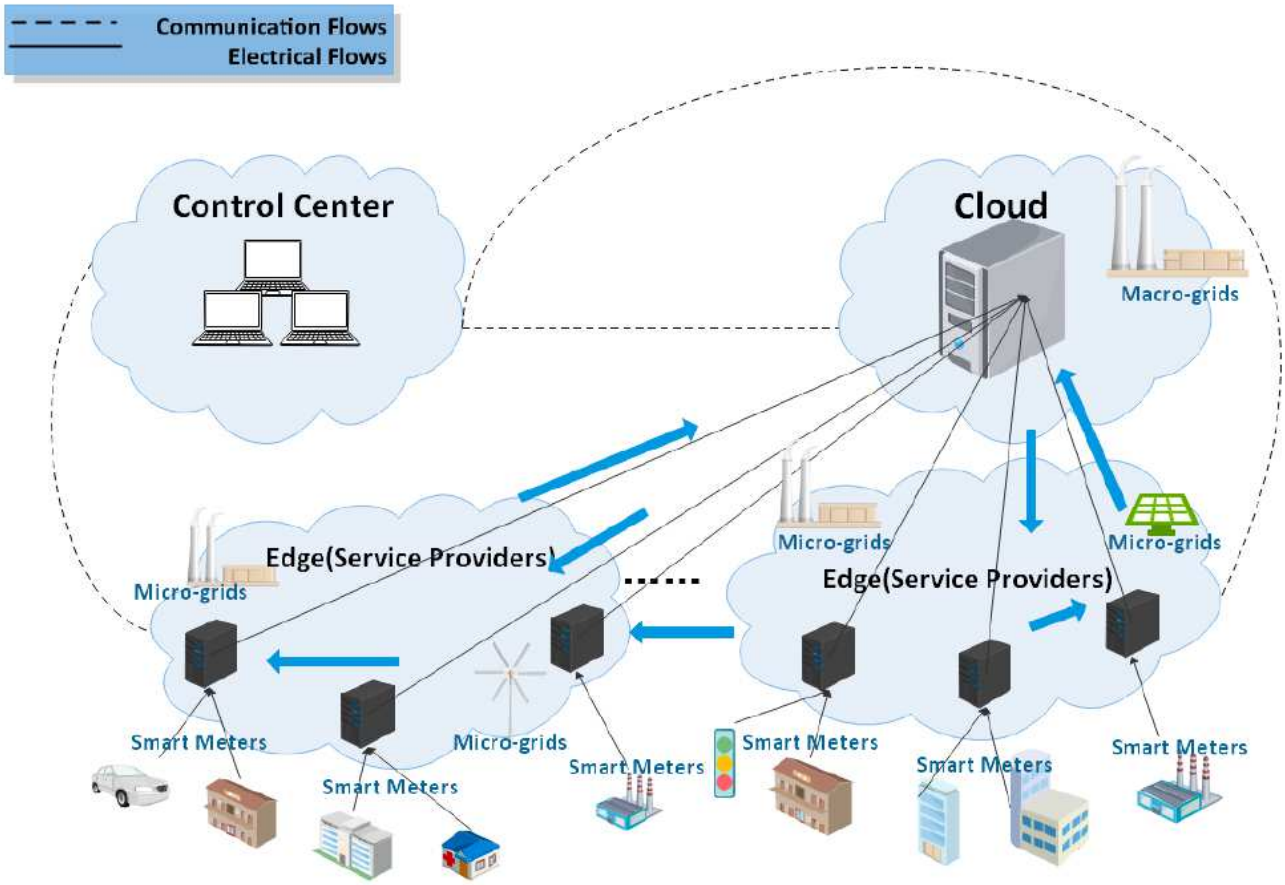

Figure 2

edge based Smart grid infrastructure

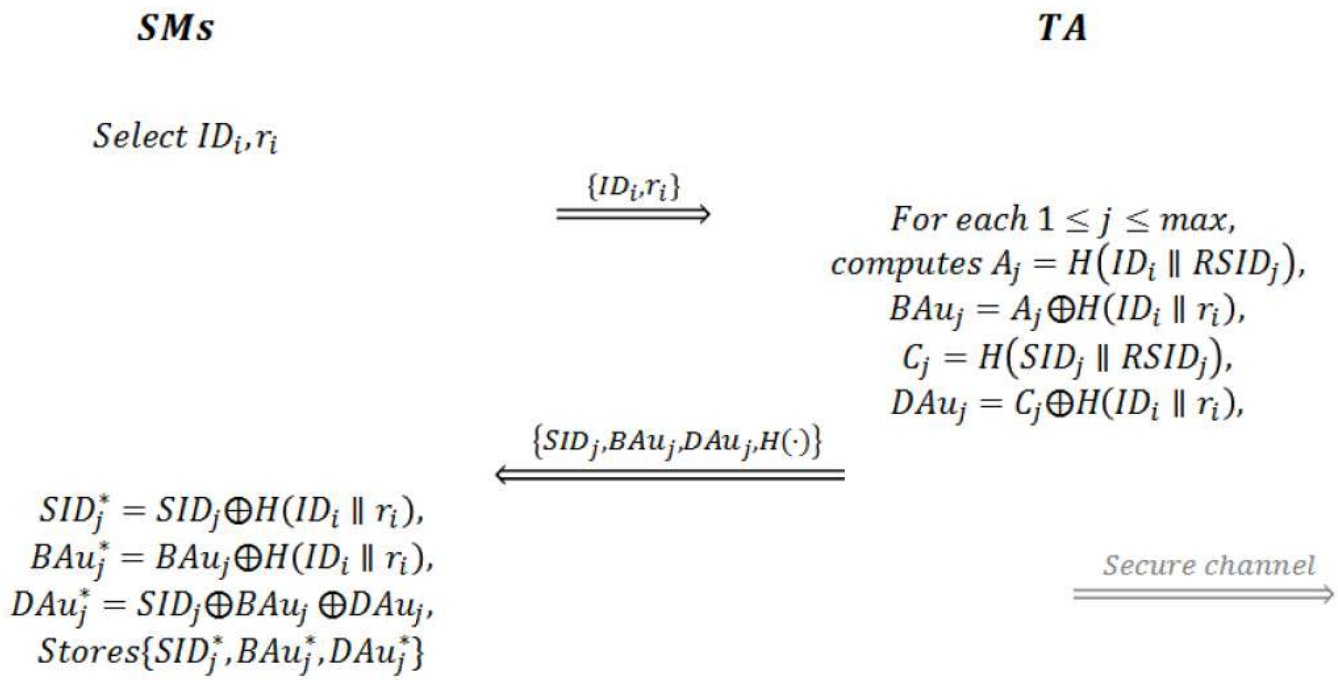

Figure 3

Smart meter registration phase

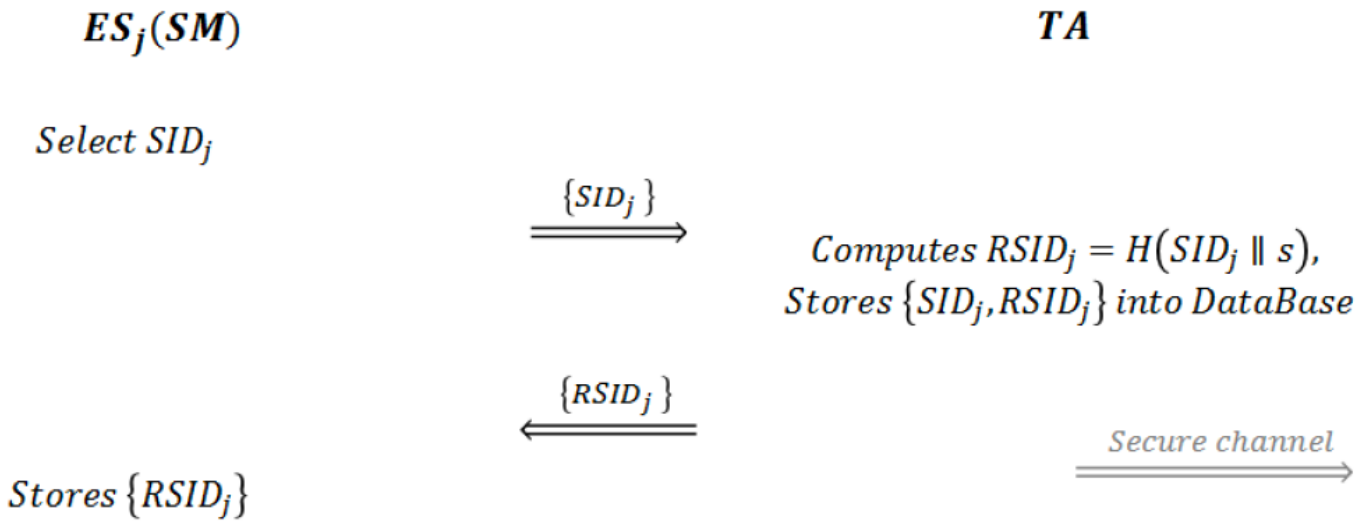

Figure 4

Edge node registration phase

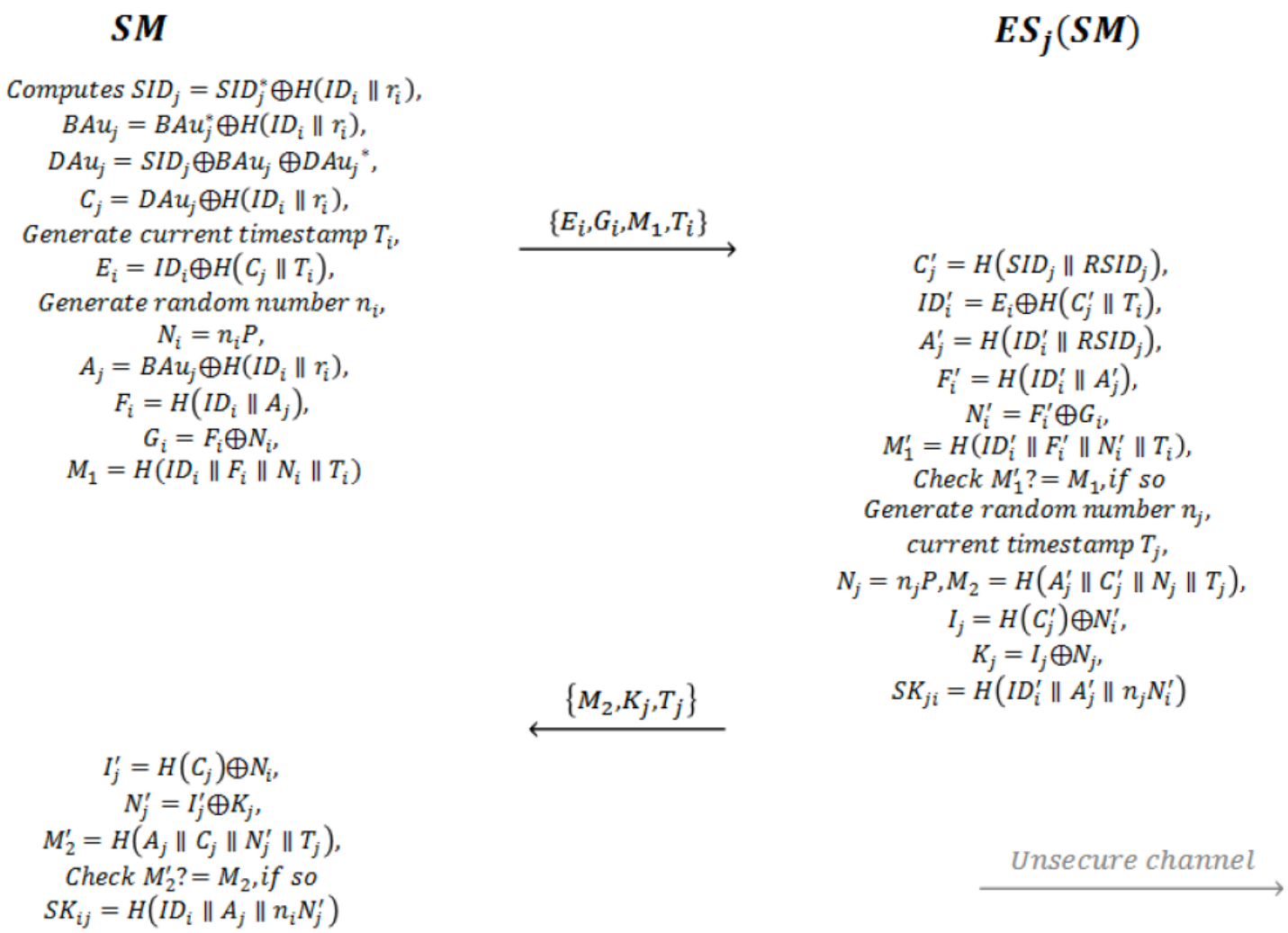

Figure 5

Login and authentication phase

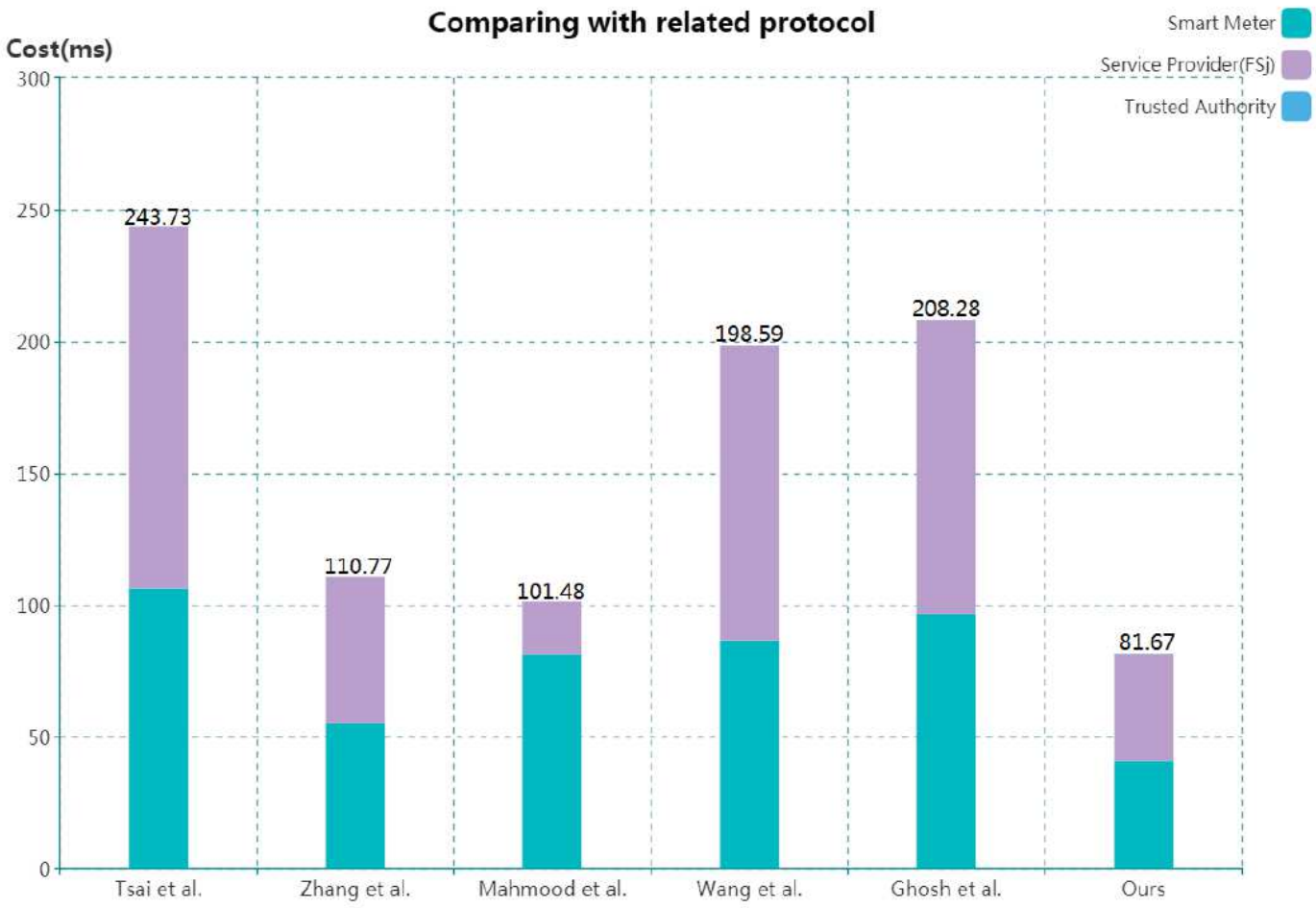

Figure 6

Comparison with related protocol