Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.036437
Article

Blockchain-Enabled Secure and Privacy-Preserving Data Aggregation for Fog-Based ITS

Siguang Chen[1,2,*], Li Yang[1,2], Yanhang Shi[1,2] and Qian Wang[1]

[1]Jiangsu Key Lab of Broadband Wireless Communication and Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing, 210003, China
[2]School of Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing, 210003, China
*Corresponding Author: Siguang Chen. Email: sgchen@njupt.edu.cn
Received: 30 September 2022; Accepted: 08 February 2023

Abstract: As an essential component of intelligent transportation systems (ITS), electric vehicles (EVs) can store massive amounts of electric power in their batteries and send power back to a charging station (CS) at peak hours to balance the power supply and generate profits. However, when the system collects the corresponding power data, several severe security and privacy issues are encountered. The identity and private injection data may be maliciously intercepted by network attackers and be tampered with to damage the services of ITS and smart grids. Existing approaches requiring high computational overhead render them unsuitable for the resource-constrained Internet of Things (IoT) environment. To address above problems, this paper proposes a blockchain-enabled secure and privacy-preserving data aggregation scheme for fog-based ITS. First, a fog computing and blockchain co-aware aggregation framework of power injection data is designed, which provides strong support for ITS to achieve secure and efficient power injection. Second, Paillier homomorphic encryption, the batch aggregation signature mechanism and a Bloom filter are effectively integrated with efficient aggregation of power injection data with security and privacy guarantees. In addition, the fine-grained homomorphic aggregation is designed for power injection data generated by all EVs, which provides solid data support for accurate power dispatching and supply management in ITS. Experiments show that the total computational cost is significantly reduced in the proposed scheme while providing security and privacy guarantees. The proposed scheme is more suitable for ITS with latency-sensitive applications and is also adapted to deploying devices with limited resources.

Keywords: Blockchain; fog computing; security; privacy-preserving; ITS

1 Introduction

Electric vehicles (EVs) are highly favored by governments worldwide and can significantly reduce the air pollution generated by fuel-driven vehicles (account for 17% of global CO2 emissions).

Research indicates that using EVs instead of traditional fuel-powered vehicles can reduce CO2 emissions by 70% [1]. With popularity of energy-saving and environmentally friendly EVs, they become an essential component of intelligent transportation systems (ITS). Due to the rapid development of ITS, vehicle-to-grid (V2G) is emerging as a promising service in ITS [2–4]. V2G provides mobile and distributed power for ITS and smart grid systems and reduces their dependence on nonrenewable energy. Furthermore, in V2G networks, as distributed energy storage elements, EVs can purchase power at valley hours and send power back to the smart grid at peak hours to achieve 'peak shaving and valley filling' to stabilize the power supply of ITS [5–7]. EVs can also generate profits by buying at a low price and selling at a high price [8]. This bidirectional power transmission produces many records, which can be analyzed by the control center to provide valuable services, such as charging/discharging scheduling, dynamic pricing, and optimal power dispatching [9–11].

However, these records can also cause a series of security and privacy issues, for example, the identities and locations of EVs and the amounts of charging and discharging [12,13]. These security and privacy issues are significant obstacles to the development of V2G in ITS. Especially when power is injected into the grid, information about power injection is highly sensitive. For example, network attackers may maliciously intercept relevant information, and power injection data may be tampered with to damage the ITS and smart grid services. In addition, each link from EVs to the control center is likely to be threatened by attacks. Measures should be taken to ensure the availability, integrity, confidentiality, and immutability of information in V2G networks.

Related studies have been conducted on the privacy and security issues of V2G networks. For example, from the perspective of identity authentication, Saxena et al. [14] presented a mutual authentication scheme for protecting the privacy of EV information by employing bilinear pairing technology. Still, its bilinear pairing implementation is costly. Abdallah et al. [15] constructed a secure authentication and privacy-preserving V2G connection scheme that leverages symmetric and public-private keys to authenticate identity. Tao et al. [16] investigated a lightweight protocol and developed capacity-based secure access authentication for the IoT that can efficiently satisfy security and privacy preservation requirements. Elliptic curve cryptography (ECC)-based schemes were also used in V2G networks. For example, Liang et al. [17] proposed a group authentication protocol by employing elliptic curve Diffie-Hellman and bilinear pairing, which effectively realizes security authentication in a V2G network. Fan et al. [18] developed a three-factor user authentication scheme, which significantly improves the robustness of the network by integrating one-way hash functions, bitwise exclusive OR (XOR) operations and ECC. Work [19] studied an anonymous key distribution scheme based on ECC, but its high computational overhead is unadaptable for the resource-constrained IoT environment.

The traditional centralized mechanism relies on a trusted third party to manage every energy transaction. Each transaction is vulnerable to a series of security threats in this scenario, such as single-point failure, denial of service attacks, and privacy leakage. Therefore, some schemes, such as [20–22], introduced blockchain technology for energy transactions because blockchains have the properties of decentralization, anonymity and immutability to provide an effective solution for V2G networks. Liu et al. [23] proposed a cross-domain identity authentication scheme based on a blockchain, which utilizes the encryption algorithm SM9 to guarantee the security and privacy required by V2G networks. Similarly, Kang et al. [24] addressed security and privacy issues in peer-to-peer (P2P) energy transactions by employing a consortium blockchain. Garg et al. [25] studied a combination mechanism of an ECC encryption algorithm and blockchain technology, which provides secure and anonymous energy transactions in V2G.

Although a secure identity authentication protocol can provide privacy preservation for users, fine-grained power consumption data are also sensitive due to their correlation with users' activities when EVs inject power into the grid. From the perspective of power consumption data, Tonyali et al. [26] developed a meter data confusion scheme for protecting consumer privacy by concealing meter data. However, it does not involve specified V2G networks. Mahmoud et al. [27] presented a power injection scheme for smart grid system that utilizes homomorphic encryption to aggregate power injection bids from the storage unit at the local gateway. Unfortunately, this scheme cannot ensure the privacy of power injection data. Accordingly, Zhang et al. [28] constructed a privacy-aware sensing data aggregation scheme for protecting the power injection information, but it involves many expensive bilinear pairing operations. Next, Zhang et al. [29] proposed a 5G-based communication and power injection scheme for privacy protection in V2G networks, which adopts the novel aggregation technology named 'hash-then-homomorphic' to further aggregate blinded bids in various time slots. The above schemes focus on the bidding prices of users but fail to consider the amounts of power that users can inject into the grid. Although they protect users' power injection and consumption data, security remains the major challenge of V2G networks.

In contrast to the previously established solutions, from the perspective of security and privacy preservations of identity and private injection data, this paper proposes a blockchain-enabled secure and privacy-preserving data aggregation scheme for fog-based ITS by combining blockchain technology and fog computing. The significant contributions are summarized as follows:

- A fog computing and blockchain co-aware three-tier aggregation framework for power injection data is constructed, which provides firm support for the V2G network to realize secure, reliable and efficient power injection.
- A secure and privacy-preserving data aggregation mechanism is designed by jointly integrating Paillier homomorphic encryption, the batch aggregation signature mechanism and a Bloom filter, which can effectively ensure the security of power injection data and the identity privacy of V2G users.
- Fine-grained homomorphic aggregation of the power injection data generated by all EVs is realized, which provides accurate data support for flexible power dispatching and effectively stabilizes the power supply of the ITS.

Finally, extensive simulation results show that the proposed scheme has lower computational costs than the previously established schemes while providing security and privacy protections. It is more suitable for ITS with latency-sensitive applications and limited resources.

The remainder of this paper is structured as follows. Section 2 describes the fog computing and blockchain co-aware three-tier aggregation network model for ITS. Section 3 presents a blockchain-enabled secure and privacy-preserving data aggregation scheme. Then, a simple security analysis and detailed performance evaluation of the experimental results are conducted in Section 4. Finally, Section 5 presents the conclusions of this study.

2 Network Model

This section constructs a fog computing and blockchain co-aware three-tier aggregation network model for ITS, consisting of five types of entities: EVs, charging stations (CSs), fog nodes, a data center and a trusted authority (TA). Consider a city as an example. The city is divided into m subareas, and each subarea usually has w EVs. For simplicity, this paper uses symbol EV_{ij} $(0 \leq i \leq w, 0 \leq j \leq m)$ to represent the i^{th} EV in the j^{th} region. All $m * w$ EVs constitute the data sensing layer. Meanwhile, each

subarea employs a fog node which is responsible for collecting and aggregating power injection data from the sensing layer. The symbol Fog_j $(0 \leq j \leq m)$ represents the fog node deployed in subarea j, and all m fog nodes constitute the data aggregation layer, which is located at the edge of the ITS. Fig. 1 illustrates this framework.

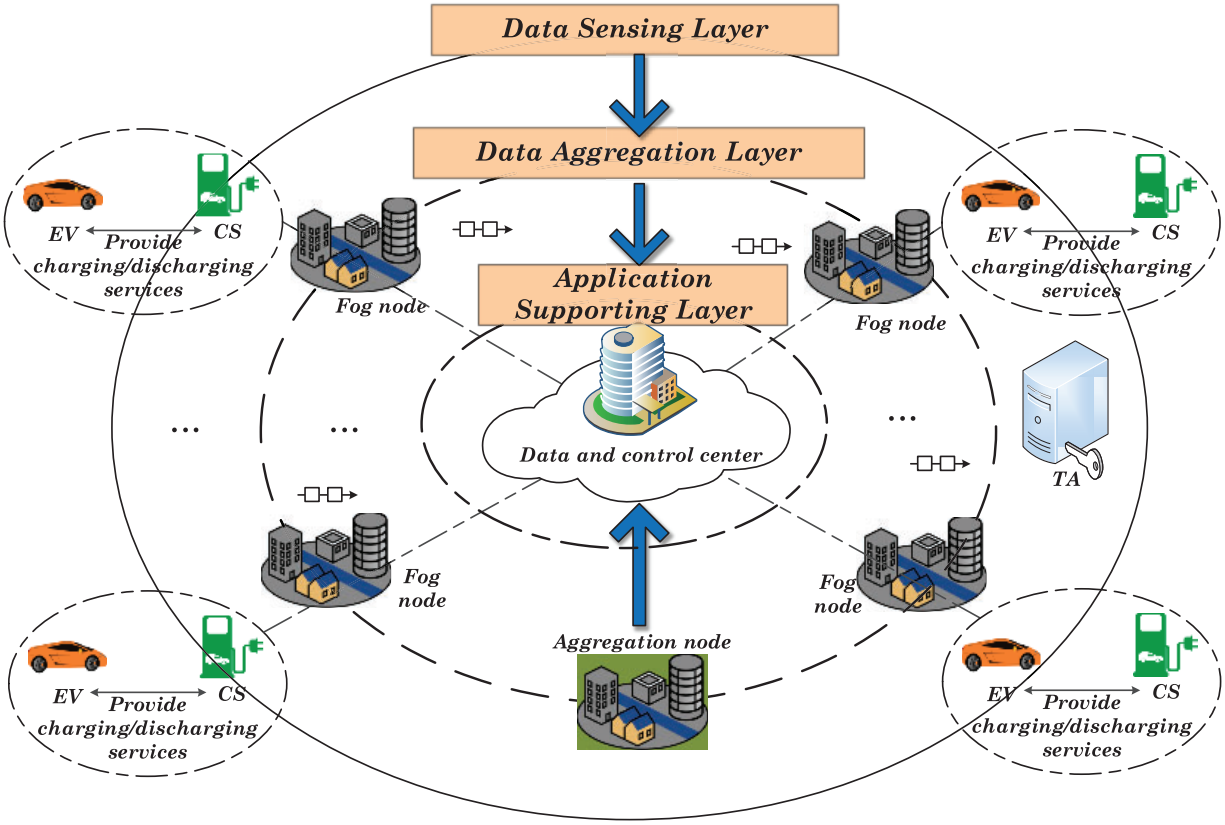

Figure 1: Three-tier aggregation framework for the ITS

Data Sensing Layer: The data sensing layer is located on the user side and includes EVs and CSs. EVs mainly refer to automobiles, motorcycles, ships, aircraft and other vehicles powered by batteries. CSs can provide charging or discharging services for vehicle batteries equipped with smart meters to record the corresponding data. In this layer, EVs can buy energy from the grid at a low price and inject the remaining power into the grid at a higher price to earn profits.

Data Aggregation Layer: The data aggregation layer is located on the edge side of the ITS. It is mainly composed of many fog nodes in all subareas. This layer selects an aggregation node in each time slot according to the remaining energy, and the remaining nodes are regarded as ordinary fog nodes. When the CS uploads the power injection data of an EV, the corresponding ordinary fog node is responsible for aggregating these data, generating a block, adding it to the blockchain through the consensus mechanism, and subsequently transmitting it to an aggregation node. The aggregation node is responsible for aggregating the received data from ordinary fog nodes (namely, for executing secondary aggregation of the power injection data) in this layer and encapsulating the data into a block with other relevant information. Then, the newly generated block is appended to the blockchain

through a consensus mechanism and uploaded to the application supporting layer, where it awaits the decryption and analysis operations on the data center.

Application Supporting Layer: The application supporting layer refers to the data and control center, which contains a cloud server and is mainly responsible for decrypting and analyzing the data uploaded from the lower layer.

TA: TA is mainly used to generate and allocate public parameters and keys for entities. At the same time, it generates a Bloom filter by collecting the pseudonyms of EVs and fog nodes and sends the Bloom filter to the corresponding entities.

The network model mainly considers the following three types of network threats: (1) threats on fog nodes and the data center: fog nodes and the cloud server are considered honest but curious. Namely, they strictly abide by the service protocol, but at the same time, they will attempt to learn all detailed information from received data. In addition, fog nodes and the cloud server are easy to capture; (2) threats on the communication link: there is a potential hazard that an attacker may obtain private user data through eavesdropping on communication links; (3) threats of active attacks: an attacker may damage the authenticity and integrity of transmission data by launching active attacks (such as tampering, forgery, or replay).

3 Blockchain-Enabled Secure and Privacy-Preserving Data Aggregation

This section presents a blockchain-enabled secure and privacy-preserving data aggregation scheme for fog-based ITS. This scheme includes five parts: system initialization, power injection request, EV-chain generation, fog-chain generation and application support.

3.1 System Initialization

The TA is used to perform system initialization, including two procedures: the generation and distribution of parameters and the registration of devices.

3.1.1 The Generation and Distribution of Parameters

In the generation phase, the TA selects security parameter k and generates $\left\{a_i, p_i^{tra}, f_i^l, t_i^{tra}\right\}$according to $gen\,(k)$. Then the TA selects the security parameter k_1 to calculate two safe large primes $|p| = |q| = |k_1|$. The prime numbers have to be different from each other. Accordingly, it can obtain the public and private keys of Paillier homomorphic encryption by calculating $n = pq$ and $\lambda = \text{lcm}\,(p-1, q-1)$, respectively. Similarly, the TA randomly selects a random integer $r \in Z_n^*$ and calculates $s = r^n \bmod n^2$. Setting $g = n + 1$ and the function $L\,(u) = (u-1)/n$ is obtained. Furthermore, the TA chooses a secure cryptographic hash function for the signature of private data: $h\colon \{0,1\}^* \to \{0,1\}^l$, and selects two secure cryptographic hash functions: $H_1\colon \{0,1\}^* \to Z_q^*$ and $H_2\colon G_1 \to Z_q^*$.

Each EV chooses a random secure key α_{ij} and calculates β_{ij} that satisfies $\alpha_{ij} = \beta_{ij} \bmod n^2$. This public key is used to calculate the EV's pseudonym $Pseu_{ij}$ ($Pseu_{ij} = \alpha_{ij} \bmod n^2$). Similarly, the fog node selects a random secure key α_j and calculates β_j with $\alpha_j = \beta_j{}^{-1} \bmod n^2$ to represent the fog's pseudonym $Pseu_j = \alpha_j \bmod n^2$. The cloud server in the data center selects a random secure key α and calculates β with $\alpha = \beta^{-1} \bmod n^2$.

Finally, after the generation of system parameters $\left(\lambda, n, k_{ij}, k_j, s, h, \alpha_{ij}, \alpha_j, \alpha, \beta_{ij}, \beta_j, \beta\right)$, the public parameters $(n, h, q_1, P_0, G_1, G_2, e)$ are released online, and others are distributed to the corresponding entities. For example, the keys $\left(k_{ij}, s, \alpha_{ij}, \beta_j, \beta\right)$, $\left(k_j, \alpha_j, \beta_{ij}, \beta\right)$ and (λ, β_j) are allocated to the electric vehicle EV_{ij}, fog node fog_j and data center, respectively.

3.1.2 Device Registration

The EVs need to be registered, and the registration process is illustrated in Fig. 2.

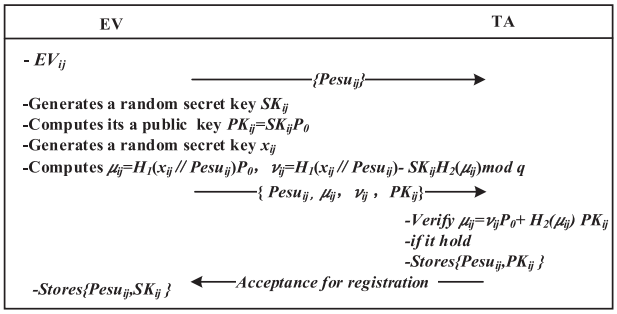

Figure 2: Registration process of EVs

Step-1: First, in a certain subarea, every new EV will generate information m_{ij} through its built-in algorithm, and m_{ij} mainly contains the EV_{ij}, vehicle information, positioning information, etc. Remarkably, the registration information is uniquely identified.

Step-2: Then, in the such subarea, EV_{ij} will select a random element SK_{ij} as its secret key and calculate $PK_{ij} = SK_{ij}P_0$ as its public key.

Step-3: Subsequently, EV_{ij} selects a random element $x_{ij} \in Z_q^*$, calculates $\mu_{ij} = H_1\left(x_{ij}||Pseu_{ij}\right) P_0$ and $\nu_{ij} = H_2\left(x_{ij}||Pseu_{ij}\right) - SK_{ij}H_2\left(\mu_{ij}\right) \mod q$.

Step-4: Next, the EV sends the parameters $\left(Pseu_{ij}, \mu_{ij}, \nu_{ij}, PK_{ij}\right)$ to the TA, the TA will verify $\mu_{ij} = \nu_{ij}P_0 + H_2\left(\mu_{ij}\right) PK_{ij}$ to ensure PK_{ij} is correct after receiving the parameters. If it passes the verification, EV_{ij} will store the parameter $\left(Pseu_{ij}||SK_{ij}\right)$, and simultaneously TA stores the parameter $\left(Pseu_{ij}||PK_{ij}\right)$. Otherwise, it will refuse this registration.

Step-5: Finally, the TA creates a Bloom filter based on the stored parameter $\left(Pseu_{ij}||PK_{ij}\right)$ for each subarea. Specifically, the TA sets a θ-bit string at the data sensing layer, then calculates the hash value of all pseudonyms in the same area. Next, it specifies the value of the string element to 1 when its index value is equal to $H\left(Pseu_{ij}\right) \mod \theta$. Finally, to achieve anonymous identity authentication, TA sends the generated Bloom filter to the fog_j and EV_{ij} in the subarea.

Similarly, the fog node also needs to be registered, and the process is the same as that for the EV. Specifically, the fog node generates parameters $\left(Pseu_j, \mu_j, \nu_j, PK_j\right)$ similarly and sends them to the TA. After receiving these parameters, the TA needs to determine whether the equation $\mu_j = \nu_j P_0 + H_2\left(\mu_j\right) PK_j$ holds. If so, it stores the parameter $\left(Pseu_j||PK_j\right)$, and the fog node stores the parameter $\left(Pseu_j||SK_j\right)$. After that, the TA creates a Bloom filter for the data aggregation layer by using the collected parameters, which are the same as the data sensing layer. Finally, the TA sends the generated Bloom filter to all fog nodes in the layer and the data center.

3.2 Power Injection Request

During peak hours of power consumption, the cloud server in the data and control center will perform the following operations.

Step-1: First, the cloud server will select a random element $\xi \in Z_q^*$ and obtain the signature MAC with the current timestamp T_s.

$$MAC = h\left(T_s||\xi\right)^{\alpha}. \tag{1}$$

Then, it obtains the power injection request packet $< Power - req - fog >= \{\xi||MAC||T_s||P_c\}$, in which MAC is used to verify the identity of the cloud server and P_c is the power price of the current slot. The cloud server sends the power injection request packet $< Power - req - fog >$ to the fog node at the data aggregation layer.

Step-2: After the fog node fog_j receives the packet $< Power - req - fog >$, it checks whether the timestamp T_s is within the validity period. If yes, the fog node fog_j further checks the authenticity of the signed MAC.

$$MAC^{\beta} = h\left(\xi||T_s\right) \bmod n^2. \tag{2}$$

According to the equation $\alpha = \beta^{-1} \bmod n^2$, the received signature is valid if the above equation holds. Next, the fog node in another subarea at the data aggregation layer will generate packet $< Power - req - fog >$ and broadcast it to EVs in its region.

Step-3: To protect the identity information of the fog node fog_j, the fog node generates the signature MAC_j by combining the pseudonym $Pseu_j$ generated during the registration phase and the current timestamp T_s.

$$MAC_j = h\left(T_s||Pseu_j\right)^{\alpha_j}. \tag{3}$$

Then, it obtains the packet $< Power - req - EV >= \left\{Pseu_j||MAC_j||T_s||P_w\right\}$, where MAC_j *is* used to verify the authenticity and integrity of the packet. Subsequently, the fog node broadcasts the packet $< Power - req - EV >$ to the EV_{ij}.

Step-4: After receiving the data packet $< Power - req - EV >$, EV_{ij} is ready to participate in the power injection; this packet also indicates the amount of power that can be injected into the grid. Precisely, similar to the fog node phase, the EV_{ij} checks the validity of the timestamp T_s, if the time is still within the validity period, EV_{ij} further verifies the authenticity of MAC_j.

$$MAC_j^{\beta_j} = h\left(Pseu_j||T_s\right) \bmod n^2. \tag{4}$$

If the above equation holds, the packet's source is legal, and the EV prepares for a power injection operation. Next, the generation processes of EV chains and fog chains are illustrated in Fig. 3.

3.3 Generation of an EV Chain

The electric vehicle EV_{ij} prepares for power injection. This process is described in detail as follows. For example, in subarea j, the amount of power to be injected by an EV_i is d_{ij}. The EV uploads these data to the data aggregation layer, so the ITS can know the EV's injection power to conduct the flexible dispatching and pricing of the power. This process is realized through the generation of EV-chain. The generation of EV-chain consists of three procedures: the generation of transaction, the creation of the EV-block and the generation of the EV-chain. The details are presented as follows.

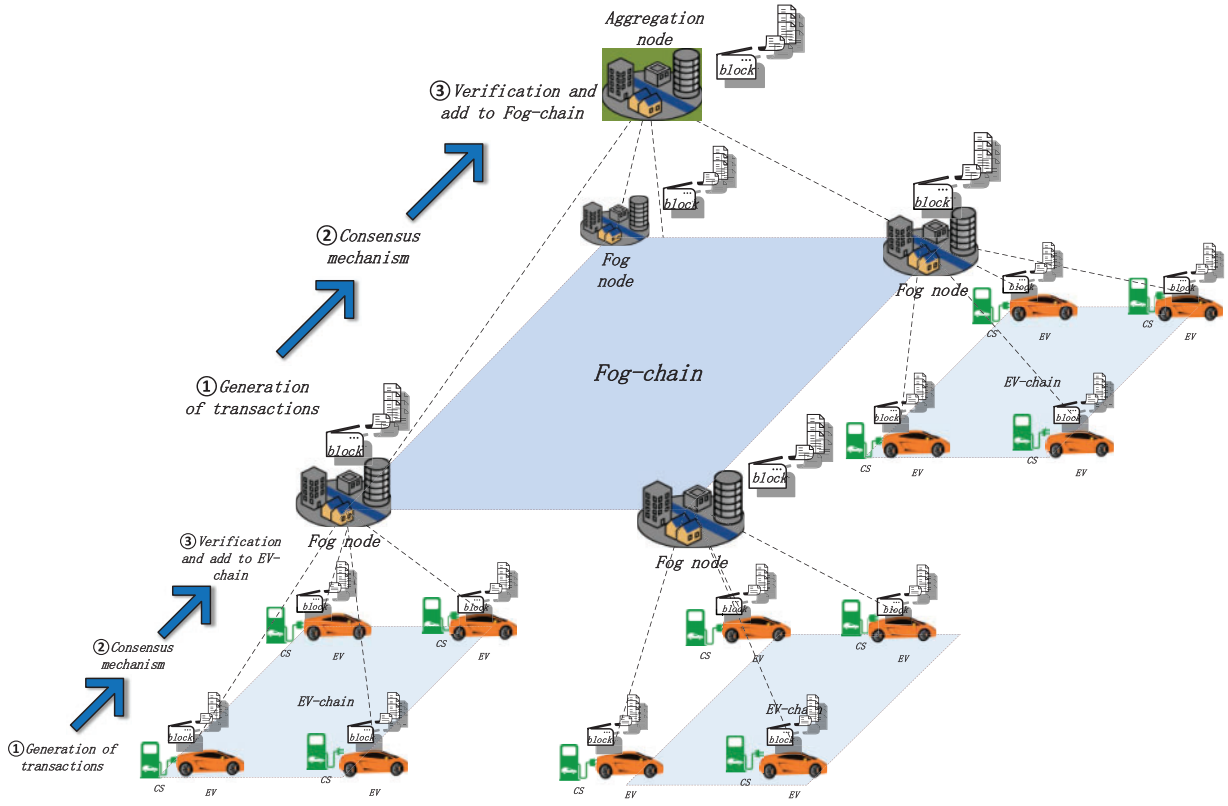

Figure 3: Generation processes of an EV chain and a fog chain

3.3.1 Generation of a Transaction

Step-1*: *Generation of injection power ciphertext. The uploaded injection data of EV_{ij} may expose the personal privacy of the user, so the injected power d_{ij} must be encrypted. The encrypted power injection data C_{ij} can be obtained by using the extended Paillier homomorphic encryption.

$$C_{ij} = \left(1 + d_{ij}n\right) \cdot s. \tag{5}$$

Step-2*: *Generation of ciphertext signature. This signature is mainly used to verify the integrity and authenticity of the ciphertext.

$$y_{ij} = h\left(T_s || Pseu_{ij}\right), \tag{6}$$

$$MAC_{ij} = h\left(C_{ij} || y_{ij}\right)^{\alpha_{ij}}. \tag{7}$$

Step-3*: *Verification of EV's pseudonym and timestamp. After the fog_j receives the reports $\left(Pseu_{ij}, C_{ij}, MAC_{ij}, T_s\right)$, it checks the validity of timestamp T_s. If T_s is valid, then fog_j further checks whether the pseudonym of the EV_{ij} is legal. This step is primarily completed through a Bloom filter.

***Step-4*: *Verification of the ciphertext signature*.** If $Pseu_{ij}$ and the current timestamp T_s are both valid, then it verifies the signature MAC_{ij} of the ciphertext by batch verification.

$$\prod_{i=1}^{w} MAC_{ij}^{\beta_{ij}} = \prod_{i=1}^{w} h\left(h\left(T_s || Pseu_{ij}\right) || C_{ij}\right) \bmod n^2. \tag{8}$$

***Step-5*: *Generation of fine-grained aggregated ciphertext*.** The fog node fog_j aggregates the injection power of all w EVs in the subarea j to obtain the fine-grained aggregated ciphertext C_j.

$$C_j = \prod_{i=1}^{w} C_{ij} \bmod n^2. \tag{9}$$

***Step-6*: *Generation of aggregated ciphertext signature*.**

$$y_j = h\left(T_s || Pseu_j\right), \tag{10}$$

$$MAC_j = h\left(C_j || y_j\right)^{\alpha_j}. \tag{11}$$

***Step-7*: *Transaction generation*.** After the above operations are completed, it generates the transaction information $T_j = \left\{C_j, Pseu_j, T_s, MAC_j\right\}$.

3.3.2 Creation of an EV Block

The fog node records the generated transaction $T_j = \left\{C_j, Pseu_j, T_s, MAC_j\right\}$ in a new block and broadcasts it for information authentication in the subarea j. This new block also contains Markel root, the hash value of previous block, and the hash value of current block. The value of Markel root is calculated by hashing the aggregated injection power ciphertext and the related pseudonym in the Markel tree. The calculation of the hash value of current block always involves the previous block, which indicates that once a new block is added to the EV chain, the content of this block is difficult to tamper with successfully because once a block has been tampered with maliciously, the subsequent blocks will be affected, which can be easily identified.

3.3.3 EV-Chain Generation

After the fog node creates a new EV block, this new block will be broadcast to all EVs in this subarea. These EVs will verify the records in the new block, and each EV only verifies the data with which it is associated to save computing resources. If the new block passes the verification, it will broadcast the verification result to other EVs in the same subarea. It is assumed that the number of malicious EVs is less than $w/3$ in the network scenario. Therefore, the new block is regarded as a valid block after passing the verification of $2w/3 + 1$ EVs or more EVs; then, it will be added to the EV chain.

3.4 Generation of the Fog-chain

When the system generates the EV chain, the fog node transmits the EV chain to the aggregation node for secondary aggregation at the data aggregation layer. Subsequently, the aggregation node generates a new block based on the result of the secondary aggregation and adds it to the fog chain. The specific process is similar to the generation of the EV chain.

3.4.1 Generation of the Transaction

***Step-1: Verification of information*.** The aggregation node queries the transaction information $T_j = \left\{C_j, Pseu_j, T_s, MAC_j\right\}$ from the EV-chain. First, to verify the $Pseu_j$ and timestamp T_p, the method is similar to that in the previous subsection. If it passes the pseudonym and timestamp verifications,

the ciphertext signature MAC_j is further verified to ensure the authenticity and integrity of ciphertext $\{C_1, C_2, \ldots, C_m\}$.

$$\prod_{j=1}^{m} MAC_j^{\beta_j} = \prod_{j=1}^{m} h\left(h\left(T_s||Pseu_j\right)||C_j\right) \bmod n^2. \tag{12}$$

Step-2: Generation of coarse-grained aggregated ciphertext. After the aggregation node completes the above verifications, it will perform a secondary aggregation to collect the injection power from m subareas.

$$C = \prod_{j=1}^{m} C_j \bmod n^2. \tag{13}$$

Step-3: Generation of the ciphertext signature. After the coarse-grained aggregated ciphertext is generated, the aggregation node signs the aggregated ciphertext C, as expressed below.

$$y = h\left(T_s||Pseu_j\right), \tag{14}$$

$$MAC = h\left(C||y\right)^{\alpha_j}. \tag{15}$$

Step-4: ***Transaction generation.*** After the above operations are completed, the aggregation node will generate the transaction $T = \left\{C, Pseu_j, T_s, MAC\right\}$.

3.4.2 Addition to the Fog-chain

The aggregation node at the data aggregation layer records the transaction in a new block and broadcasts this new block to other fog nodes for information authentication. Similar to the creation of the EV chain, a new fog block at the aggregation node mainly includes transactions, timestamps, pseudonyms, Merkle roots, and the hash values of the previous block and the current block. After the execution of the consensus mechanism, the verified block is added to the fog chain, and the aggregation node sends the newly generated fog chain to the data center at the application supporting layer for further processing.

3.5 Application Supporting

At the application supporting layer, the cloud server reads the transaction information T of the received fog chain and verifies the identity of the aggregation node. If the node's identity is correct, it further checks the signature of the ciphertext. If it also passes the signature verification, the Paillier decryption algorithm is used to decrypt the aggregated ciphertext, and the steps are as follows.

Step-1: Verification of information. After the cloud server queries the transaction information $T = \left\{C, Pseu_j, T_s, MAC\right\}$ from the fog chain, it verifies $Pseu_j$ and timestamp T_s, the method is similar to that in the previous subsection; if the verifications of pseudonym and timestamp are both passed, and then the ciphertext signature is verified to ensure the authenticity and integrity of ciphertext C.

$$MAC^{\beta_j} = h\left(h\left(T_s||Pseu_j\right)||C\right) \bmod n^2. \tag{16}$$

Step-2: Decryption of the aggregated ciphertext. If all the verifications are passed, the decryption operation of the Paillier algorithm is applied to decrypt the aggregated ciphertext using the private key, and it can obtain the total amount of plaintext injection power of the whole area. Meanwhile, the amount of injection power of each subarea can be derived by employing Horner's rule; namely, the fine-grained aggregation result can be recovered. Based on these coarse-grained and fine-grained

results, the ITS can flexibly regulate the power supply during peak hours to maintain the supply-demand balance. It can also realize peak shaving and valley filling by a historical data-based time-of-use pricing mechanism. Thus, these data can provide strong support for various application services in ITS.

4 Performance Evaluation

The focus of this section is to analyze the security and privacy of the developed scheme and evaluate its performance while guaranteeing the security and privacy of collected data.

As described in Section 2, there are three main types of network threats: honest-but-curious processing nodes, link eavesdropping and active attacks. To resist first two types of threats, Paillier homomorphic encryption is utilized to encrypt the power injection data to counteract eavesdropping attacks and prevent information from leaking to honest-but-curious fog and cloud nodes. Although the cloud server can derive the aggregation result of each subarea, it cannot recover the power injection data of each EV. Consequently, the proposed scheme can guarantee confidentiality for EV power injection data. Intending to resist active attacks from attackers, the signature mechanism with a timestamp in the proposed scheme guarantees the integrity and validity of private data. Furthermore, the Bloom filter can ensure the identity anonymity and authenticity of valid nodes. As a result, the developed scheme guarantees the integrity and validity of private data and provides identity protection for EVs and fog nodes.

To analyze the performance of the proposed scheme, this paper compares the computational costs of this scheme with two previously established schemes: privacy-aware data aggregation (PADA) [28] and efficient privacy-preserving communication and power injection (ePPCP) [29]. To facilitate the explanation, with the same definitions in reference [30], it represents T_{E1}, T_{E2}, T_M and T_P as the exponential operation in $z_{n^2}^*$, the exponential operations, multiplication operations and bilinear pairings $\mathbb{G}$, respectively. In our simulation scenario, it assumes that there is one control center in the application supporting layer. The number of fog nodes is 50, and the number of EVs in each subarea ranges from 0 to 1000. The parameters are randomly generated within their ranges. The experiment is performed 1000 times to evaluate the average value of the simulation results. The code is implemented using the pairing-based cryptography (PBC) library, and all the simulations are performed on a laptop with an Intel Core i5-7200U 2.5 GHz CPU and 8.00 GB RAM. Table 1 lists their time costs in the execution process.

Table 1: Operations and time costs

Notations	Descriptions	Time cost (ms)
T_{E1}	Exponentiation operation in $z_{n^2}^*$	1.60
T_{E2}	Exponentiation operation in $\mathbb{G}$	1.62
T_M	Multiplication operation in $\mathbb{G}$	0.06
T_P	Pairing operation in $\mathbb{G}$	17.70

In the PADA scheme, the generation of a power request packet $< Power - req - UC >$ requires a computational cost of $2T_M$; in the privacy aggregation stage, the required computational cost is $(2w+1)\,T_P + (w+2)\,T_M + T_{E2} + 2T_{E1}$. After receiving the packet $< Power - req - UC >$, the gateway needs $2T_P + T_M$ to verify the packet and generate a new packet. Meanwhile, a computational cost of $wT_P + (w+2)\,T_M + T_{E1}$ is required during the privacy aggregation phase. Next, each storage unit

incurs a computational cost of $2T_p + 4T_M + T_{E2} + (k+2)\,T_{E1}$, where k represents the number of time slots. Since this scheme focuses only on the power supply of a specified time slot, it sets $k = 1$ for comparison. Therefore, the computational costs at the utility company (UC), gateway (GW) and power storage unit (PSU) are $(2w+1)\,T_P + (w+4)\,T_M + T_{E2} + 2T_{E1}$, $(w+2)\,T_P + (w+3)\,T_M + T_{E1}$ and $2wT_P + 4wT_M + wT_{E2} + 3wtT_{E1}$, respectively.

In the ePPCP scheme, to generate power requests, a computational cost of T_{E1} is incurred. In the privacy aggregation phase, the computational cost is $(3w+6)\,T_{E1}$. After receiving the data packet $< Power - req - UC >$, the gateway incurs a computational cost of $3T_{E1}$ to verify the packet. In the privacy aggregation stage, the gateway incurs the computational cost of $(5w+2)\,T_{E1}$. To successfully bid, each PSU incurs a computational cost of $(k+8)\,T_{E1}$. Then, the computational costs at UC, GW, and PSU are $(3w+7)\,T_{E1}$, $(5w+5)\,T_{E1}$ and $(k+8)\,T_{E1}$, respectively.

To generate the power request $< Power - req - fog >$, the cloud server incurs a computational cost of T_{E1}; while in the privacy aggregation stage, the required computational cost is $T_M + T_{E1}$. Here the cloud server is equivalent to the UC of the above solutions, and the scheme transfers this part to be implemented in the cloud server. After receiving the data packet $< Power - req - fog >$, the fog node needs the cost of $2wT_{E1}$ to verify the packet and generate a new signature. During the aggregation phase, the fog node incurs the cost of $2wT_M + wT_{E1}$. Here the fog node is equivalent to the gateway in the above solutions. Subsequently, the data packet is received at the EV. A cost of wT_{E1} is required to verify the packet, and the EV incurs a cost of $2wT_M + wT_{E1}$ to protect the privacy of EV. Here the EV is equivalent to the PSU in above schemes. The computational costs at the cloud server, fog node, and EV are $wT_M + (w+1)\,T_{E1}$, $2wT_M + 3wT_{E1}$ and $w\,(2T_M + 2T_{E1})$, respectively. The cost comparisons are compared in Table 2. It can be observed that the developed scheme outperforms the other two benchmark schemes.

Table 2: The computational costs of different schemes

	PADA	ePPCP	Our scheme
UC	$(2w+1)\,T_P + (w+4)\,T_M + T_{E2} + 2T_{E1}$	$(3w+7)T_{E1}$	$wT_M + (w+1)\,T_{E1}$
GW	$(w+2)\,T_P + (w+3)\,T_M + T_{E1}$	$(5w+5)T_{E1}$	$2wT_M + 3wT_{E1}$
PSU	$w\,(2T_P + 4T_M + T_{E2} + 3T_{E1})$	$9wT_{E1}$	$wT_M + 2wT_{E1}$

Through the above analysis, the developed solution has advantages over computational costs, but these advantages are not unconditional. For example, to achieve secure and reliable power injection, this paper adopts blockchain technology which usually associates with specific memory and bandwidth consumptions, but this trade-off is acceptable and well worth for the security and privacy preservation improvements. To show the above results more intuitively, the following figures are shown.

As shown in Fig. 4, the computational cost at the UC is directly proportional to the number of EVs. Compared with PADA and ePPCP, the computational cost at the UC of the proposed scheme is lower, and as the number of EVs increases, this advantage is enhanced. This is mainly because PADA uses expensive bilinear pairing calculations to generate packets, and ePPCP uses many exponential operations. Compared to PADA and ePPCP, the proposed scheme effectively avoids these operations, thereby reducing computational costs.

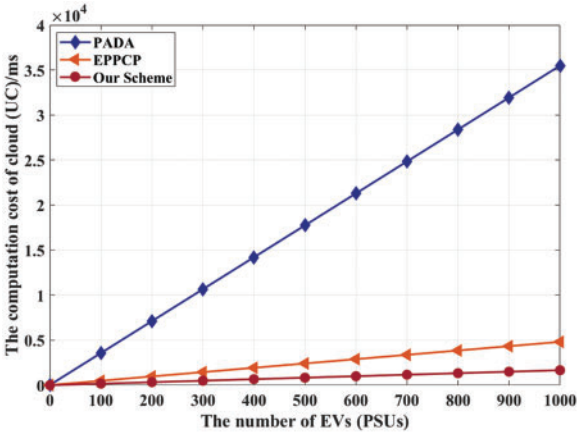

Figure 4: Comparison of computational costs of the cloud (UC)

As shown in Fig. 5, the computational cost at the fog (GW) is also proportional to the number of EVs. Meanwhile, compared with those of schemes PADA and ePPCP, the computational cost of the proposed scheme is lower, and this advantage is strengthened with the increase in the number of EVs. This is mainly because PADA uses pairing calculations to verify the data packets generated by the UC in the privacy aggregation phase, the PADA scheme also uses pairing operations and many multiplication operations. At the same time, ePPCP uses many exponential operations to verify data packets and aggregate private data. Compared to PADA and ePPCP, the proposed scheme effectively avoids pairing operations and simplifies the operations as much as possible; thus, it can reduce the computational costs significantly.

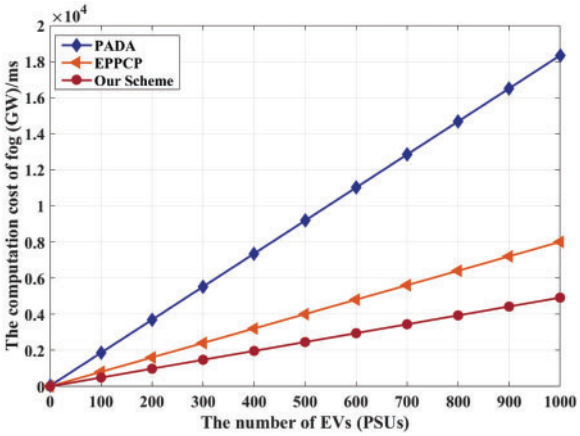

Figure 5: Comparison of computational costs of the fog (GW)

Results similar to those in the above two figures are presented in Fig. 6, which shows the computational cost of the proposed scheme is lower than those of the other two schemes at the EV. PADA utilizes many bilinear pairing operations to protect private data and verify the integrity and authenticity of packets broadcasted by the gateway. Meanwhile, in ePPCP, many exponential operations are used, which consume a substantial amount of time.

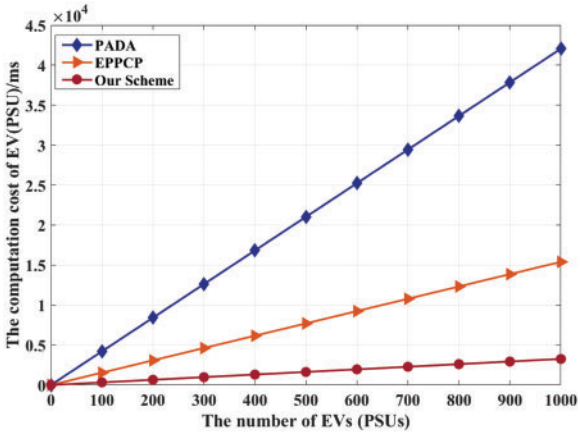

Figure 6: Comparison of computational costs of the EV (PSU)

As depicted in Fig. 7, the total computational cost is significantly reduced in our proposed scheme while providing security and privacy protections. The proposed scheme is more suitable for ITS with latency-sensitive applications and is also adapted to deploying devices with limited resources.

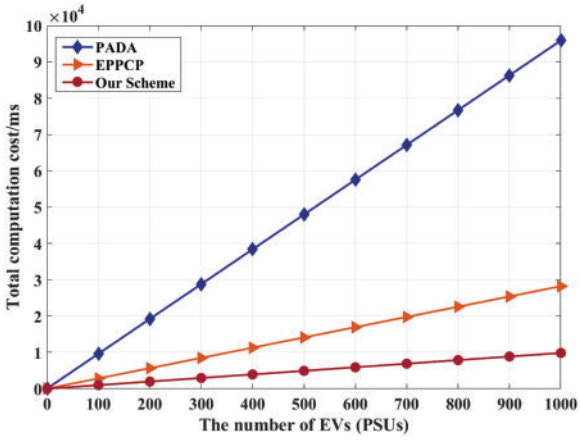

Figure 7: Total computational costs comparison

5 Conclusions

Existing privacy protection schemes tend to focus on the security of identity, while ignoring that fine-grained power consumption data are also sensitive. And many privacy protection schemes of power consumption data require high computational costs. To achieve the efficient aggregation of EV injection data with security and privacy guarantees, this paper proposes a blockchain-enabled secure and privacy-preserving data aggregation scheme for fog-based ITS. The proposed scheme constructs a secure and efficient aggregation framework by combining fog computing and blockchain technology. Then, it uses the Bloom filter and lightweight signature mechanism to build a secure and anonymous registration and authentication mechanism for preventing forgery attacks from malicious nodes. Furthermore, Paillier homomorphic encryption is implemented to encrypt the power injection data to ensure the confidentiality of the data. Finally, the performance evaluation shows that the

proposed scheme has a lower computational cost with security and privacy guarantees. It is assumed that all EVs are benign and do not consider the impact of malicious EVs uploading fake private data on system performance. In the future, this paper will combine blockchain and smart contract technology to defend against malicious poisoning attacks.

Funding Statement: The authors received Funding for this study from the National Natural Science Foundation of China (No. 61971235), the China Postdoctoral Science Foundation (No. 2018M630590), the Jiangsu Planned Projects for Postdoctoral Research Funds (No. 2021K501C), the 333 High-level Talents Training Project of Jiangsu Province, and the 1311 Talents Plan of NJUPT.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] J. Dixon, W. Bukhsh, C. Edmunds and K. Bell, "Scheduling electric vehicle charging to minimise carbon emissions and wind curtailment," *Renewable Energy*, vol. 161, no. 4, pp. 1072–1091, 2020.

[2] L. Calearo, M. Marinelli and C. Ziras, "A review of data sources for electric vehicle integration studies," *Renewable and Sustainable Energy Reviews*, vol. 151, no. 3, pp. 1–18, 2021.

[3] J. Van Mierlo, M. Berecibar, M. El Baghdadi, C. De Cauwer, M. Messagie *et al.,* "Beyond the state of the art of electric vehicles: A fact-based paper of the current and prospective electric vehicle technologies," *World Electric Vehicle Journal*, vol. 12, no. 1, pp. 1–26, 2021.

[4] M. R. Patel, A. P. Shah, K. J. Chudasama and G. J. Jadhav, "A review of EV converters performance during V2G /G2V mode of operation," in *Proc. INCET*, Belgaum, India, pp. 1–7, 2022.

[5] M. Boni, T. Ch, S. Alamanda, B. V. S. G. Arasada and A. Maria, "An efficient and secure anonymous authentication scheme for V2G networks," in *Proc. ICDCS*, Coimbatore, India, pp. 432–436, 2022.

[6] G. Sharma, A. M. Joshi and S. P. Mohanty, "An efficient physically unclonable function based authentication scheme for V2G network," in *Proc. ISES*, Jaipur, India, pp. 421–425, 2021.

[7] S. Ahmed, S. Shamshad, Z. Ghaffar, K. Mahmood, N. Kumar *et al.,* "Signcryption based authenticated and key exchange protocol for EI-based V2G environment," *IEEE Transactions on Smart Grid*, vol. 12, no. 6, pp. 5290–5298, 2021.

[8] F. Wang, L. Jiao, K. Zhu, K. Zhu and L. Zhang, "Online edge computing demand response via deadline-aware V2G discharging auctions," *IEEE Transactions on Mobile Computing*, pp. 1–14, 2022. https://doi.org/10.1109/TMC.2022.3208420

[9] B. Bibak and H. Tekiner-Moğulkoç, "A comprehensive analysis of vehicle to grid (V2G) systems and scholarly literature on the application of such systems," *Renewable Energy Focus*, vol. 36, no. February (2016), pp. 1–20, 2021.

[10] S. Goyal, S. Bhushan, Y. Kumar, A. H. S. Rana, M. R. Bhutta *et al.,* "An optimized framework for energy-resource allocation in a cloud environment based on the whale optimization algorithm," *Sensors*, vol. 21, no. 5, pp. 1–24, 2021.

[11] S. Rani, D. Koundal, Kavita, M. F. Ijaz, M. Elhoseny *et al.,* "An optimized framework for WSN routing in the context of industry 4. 0," *Sensors*, vol. 21, no. 19, pp. 1–15, 2021.

[12] F. Gao, L. Zhu, M. Shen, K. Sharif, Z. Wan *et al.,* "A blockchain-based privacy-preserving payment mechanism for vehicle-to-grid networks," *IEEE Network*, vol. 32, no. 6, pp. 184–192, 2018.

[13] W. Han and Y. Xiao, "Privacy preservation for V2G networks in smart grid: A survey," *Computer Communications*, vol. 91, no. 1, pp. 17–28, 2016.

[14] N. Saxena and B. J. Choi, "Authentication scheme for flexible charging and discharging of mobile vehicles in the V2G networks," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 7, pp. 1438–1452, 2016.

[15] A. Abdallah and X. S. Shen, "Lightweight authentication and privacy preserving scheme for V2G connections," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 3, pp. 2615–2629, 2017.
[16] M. Tao, K. Ota, M. Dong and Z. Qian, "AccessAuth: Capacity-aware security access authentication in federated-IoT-enabled V2G networks," *Journal of Parallel and Distributed Computing*, vol. 118, no. 4, pp. 107–117, 2018.
[17] G. Liang, S. R. Weller, F. Luo, J. Zhao and Z. Dong, "Distributed blockchain-based data protection framework for modern power systems against cyber attacks," *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 3162–3173, 2019.
[18] M. Fan and X. Zhang, "Consortium blockchain based data aggregation and regulation mechanism for smart grid," *IEEE Access*, vol. 7, pp. 35929–35940, 2019.
[19] Z. Guan, G. Si, X. Zhang, L. Wu, N. Guizani *et al.,* "Privacy-preserving and efficient aggregation based on blockchain for power grid communications in smart communities," *IEEE Communications Magazine*, vol. 56, no. 7, pp. 82–88, 2018.
[20] L. F. A. Roman, P. R. L. Gondim and J. Lloret, "Pairing-based authentication protocol for V2G networks in smart grid," *Ad Hoc Networks*, vol. 90, no. 3, pp. 1–16, 2019.
[21] M. Wazid, A. K. Das, N. Kumar and J. J. P. C. Rodrigues, "Secure three factor user authentication scheme for renewable energy based smart grid environment," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 3144–3153, 2017.
[22] D. Abbasinezhad-Mood and M. Nikooghadam, "An anonymous ECC-based self-certified key distribution scheme for the smart grid," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 10, pp. 7996–8004, 2018.
[23] D. Liu, D. Li, X. Liu, L. Ma, H. Yu *et al.,* "Research on a cross-domain authentication scheme based on consortium blockchain in V2G networks of smart grid," in *Proc. EI2*, Beijing, China, pp. 1–5, 2018.
[24] J. Kang, R. Yu, X. Huang, S. Maharjan, Y. Zhang *et al.,* "Enabling localized peer-to-peer electricity trading among plug-in hybrid electric vehicles using consortium blockchains," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 3154–3164, 2017.
[25] S. Garg, K. Kaur, G. Kaddoum, F. Gagnon, J. J. P. C. Rodrigues *et al.,* "An efficient blockchain-based hierarchical authentication mechanism for energy trading in V2G environment," in *Proc. ICC Workshops*, Shanghai, China, pp. 1–6, 2019.
[26] S. Tonyali, O. Cakmak, K. Akkaya, M. M. E. A. Mahmoud, I. Guvenc *et al.,* "Secure data obfuscation scheme to enable privacy-preserving state estimation in smart grid AMI networks," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 709–719, 2016.
[27] M. M. E. A. Mahmoud, N. Saputro, P. K. Akula and K. Akkaya, "Privacy-preserving power injection over a hybrid AMI/LTE smart grid network," *IEEE Internet of Things Journal*, vol. 4, no. 4, pp. 870–880, 2017.
[28] Y. Zhang, D. Zheng, Q. Zhao, C. Lai and F. Ren, "PADA: Privacy-aware data aggregation with efficient communication for power injection in 5G smart grid slice," in *Proc. NaNA*, Kathmandu, Nepal, pp. 11–16, 2017.
[29] Y. Zhang, J. Li, D. Zheng, P. Li and Y. Tian, "Privacy-preserving communication and power injection over vehicle networks and 5G smart grid slice," *Journal of Network and Computer Applications*, vol. 122, no. 6, pp. 50–60, 2018.
[30] S. Chen, L. Yang, C. Zhao, V. Varadarajan and K. Wang, "Double-blockchain assisted secure and anonymous data aggregation for fog-enabled smart grid," *Engineering*, vol. 8, no. 1, pp. 159–169, 2022.