applied sciences

MDPI

Article

A Charging and Discharging Data Privacy Protection Scheme for V2G Networks Based on Cloud–Fog-End

Baoyi Wang [1,2], Ziyan Shi [1,2,*] and Shaomin Zhang [1,3]

[1] Department of Computer, North China Electric Power University, Baoding 071003, China; wangbaoyi@126.com (B.W.); zhangshaomin@126.com (S.Z.)
[2] Hebei Key Laboratory of Knowledge Computing for Energy and Power, Baoding 071003, China
[3] Engineering Research Center of Intelligent Computing for Complex Energy Systems, Ministry of Education, Baoding 071003, China
* Correspondence: shiziyan106@163.com; Tel.: +86-176-2523-8757

Abstract: Due to the openness of the vehicle-to-grid (V2G) network, the upload of charging and discharging data faces severe security challenges such as eavesdropping, tampering, and forgery. These challenges can lead to privacy breaches, transmission delays, and service quality degradation. To address these issues, a V2G network architecture based on cloud–fog-end is designed, and a charging and discharging data privacy protection scheme is proposed. We employ a pseudonym mechanism to achieve the conditional privacy protection of electric vehicle (EV) users. We design a certificateless aggregate signcryption (CLASC) algorithm to guarantee the security of uploading the charging and discharging privacy data. The algorithm solves certificate management and key escrow issues, utilizes aggregate signature operations to save network bandwidth, and avoids complex computations like bilinear pairings and exponents. Additionally, the scheme delegates the aggregate verification process to the fog layer, thereby alleviating the computational burden on the cloud layer, decreasing transmission delays, and enhancing the efficiency and reliability of the V2G network. The analysis results indicate that the scheme not only meets the required security objectives, but also has lower computational and communication overheads, making it suitable for scenarios involving the charging and discharging of large-scale EVs in V2G networks.

Keywords: V2G network; cloud–fog-end; electric vehicle; privacy protection; pseudonyms; certificateless aggregate signcryption

Citation: Wang, B.; Shi, Z.; Zhang, S. A Charging and Discharging Data Privacy Protection Scheme for V2G Networks Based on Cloud–Fog-End. *Appl. Sci.* **2024**, *14*, 4096. https://doi.org/10.3390/app14104096

Academic Editor: Andreas Sumper

Received: 29 March 2024
Revised: 28 April 2024
Accepted: 8 May 2024
Published: 11 May 2024

1. Introduction

The vehicle-to-grid (V2G) network enables bidirectional communication and power exchange between electric vehicles (EVs) and the power grid [1]. EVs can serve as distributed energy storage systems [2], charging from the grid during low demand periods and discharging during peak demand, thus helping to alleviate grid load fluctuations and providing economic benefits to EV users [3]. With policy support and ongoing battery technology innovation, the EV industry is experiencing rapid development [4]. The International Energy Agency (IEA) estimates that the global number of EVs will reach 230 million by 2030 [5]. In the V2G network scenario, the charging pile (CP) uploads the charging and discharging data of EVs to the charging service operator (CSO) for processing, and then the CSO issues control commands to the CP to manage the EV. The data in the V2G network contain a large amount of private information, including the EV user's identity, license plate number, physical card number, charging pile number, charging and discharging quantity, and geographic location [6].

However, owing to the random nature of charging and discharging behavior and the openness of the communication network, there are severe security challenges when uploading the charging and discharging data [7], such as eavesdropping, tampering, forgery, node impersonation, and denial of service attacks, which can easily lead to privacy

breaches. If the EV user identity information is leaked, it could be used for commercial promotion by some companies or even for fraudulent activities by criminals, harming the interests of the users. Furthermore, if charging and discharging data are leaked, this could lead to the issuance of incorrect control commands by the CSO, severely impacting the normal operation of the V2G network. In addition, with the rapid growth of EVs and CPs, the communication overhead between CPs and the CSO will become very large, leading to heavy computational pressure on the CSO, transmission delays, decreased service quality, and increased privacy breach risks in V2G networks.

Cloud–based solutions have been proposed for this purpose [8], accompanied by additional challenges such as higher network bandwidth and latency owing to the distance between the EV and the cloud server. Fog computing [9] is considered an extension of cloud computing, shifting the storage, computation, and other functions of cloud computing from the center of the network to the edge. Fog nodes (FNs) can use batch validation to relieve computation and storage stress in the control center, as well as reduce data transmission distances, data transmission delays, and the cost of sending data [10]. To enable reliable, secure, and efficient services for the smart grids, fog computing can provide distributed computing services to users, supporting low-latency and location-aware services [11]. Therefore, fog computing architecture can also be used in V2G networks, leveraging the computing and storage resources of FNs to improve the efficiency and reliability of V2G networks [12].

In order to address the privacy protection issues of charging and discharging data in V2G networks, data-based privacy protection schemes were proposed in [13–15], mainly encrypting the data to make them difficult for attackers to obtain. However, these schemes involve complex operations and high computational overhead, which are not suitable for V2G network environments with limited resources. Identity-based privacy protection schemes were proposed in [16–19] that mainly protect the users' privacy by blurring the true identity of EV users. In [20], fog computing was combined with blockchain technology to achieve the security of the EV charging process. Xia et al. [21] proposed a charging identity authentication scheme based on fog computing that uses group signatures to protect the privacy of EV users. However, this scheme has issues with certificate management and key escrow. Moreover, the above schemes protect the identity of EV users, but charging and discharging data in V2G networks may still be eavesdropped on, so the EV users' identity and charging and discharging data must be protected at the same time.

Signcryption is a cryptographic primitive that combines signature and encryption to simultaneously achieve confidentiality and unforgeability in a single logical step [22]. Lu et al. [23] first proposed a scheme that combines signcryption and certificateless aggregate signatures, leading to the study of a large number of certificateless aggregate signcryption (CLASC) schemes. Aiming to achieve privacy protection in vehicular sensor network communications, Dai and Xu [24] proposed a CLASC scheme that satisfies confidentiality, unforgeability, forward secrecy, and conditional traceability. Zhang et al. [25] combined the consortium blockchain with the CLASC algorithm to achieve a lightweight and secure communication of real-time power information, but the use of blockchain in it increases the computational overhead.

However, more research is needed on CLASC schemes in fog computing environments. Cui et al. [26] designed a CLASC scheme for VANETs, but users can learn the master key through scalar operations, posing security risks. The CLASC scheme proposed by Basudan et al. [27] improved the security of the fog-based vehicular crowd-sensing road condition monitoring systems. Wang et al. [28] proposed a traceable road condition monitoring scheme based on cloud–fog, saving computational resources and network bandwidth. Dohare et al. [29] proposed a CLASC scheme for cloud–fog-based Industry 4.0, which achieves mutual identity authentication, public verifiability, data integrity, and confidentiality. However, all the above schemes are based on bilinear mappings, resulting in large computational and communication overhead.

In summary, the main contributions of this paper are as follows.

(1) In order to ensure the security and reliability of V2G networks during the charging and discharging data upload process for large-scale EVs, a cloud–fog-based V2G network architecture is designed, and a charging and discharging data privacy protection scheme is proposed.
(2) In the proposed scheme, we employ a pseudonym mechanism to achieve anonymity and the traceability of the EV users' identities, thus attaining conditional privacy protection. We also designed a CLASC algorithm that guarantees the security of uploading charging and discharging privacy data.
(3) The proposed scheme addresses certificate management and key escrow issues; employs aggregate operations to save network bandwidth; utilizes signature and encryption operations simultaneously to simplify computational steps; and avoids bilinear pairing and exponentiation and other complex operations.
(4) According to the cloud–fog-based V2G network architecture, the aggregate verification is processed by the fog layer, alleviating the computational burden on the CSO, reducing transmission delays, and improving the efficiency of the V2G network.
(5) The security analysis indicates that the proposed scheme not only meets the required security features, including conditional anonymity, confidentiality, and unforgeability, but can also resist common attacks such as impersonation, replay, and DDoS. The performance analysis demonstrates that the scheme exhibits high efficiency in both computation and communication, making it suitable for V2G network environments with limited resources.

This paper is structured as follows: Section 2 introduces the system model, threat model, and safety objectives of this scheme. In Section 3, a CLASC scheme based on cloud–fog-end is proposed. In Section 4, the correctness and security of the proposed scheme is analyzed. In Section 5, the performance evaluation is conducted by analyzing computational and communication costs and comparing them with other related schemes. Section 6 concludes this paper and puts forward the future research directions.

2. Problem Formalization

2.1. System Model

The proposed V2G network communication architecture contains cloud, fog, and user layers, as shown in Figure 1.

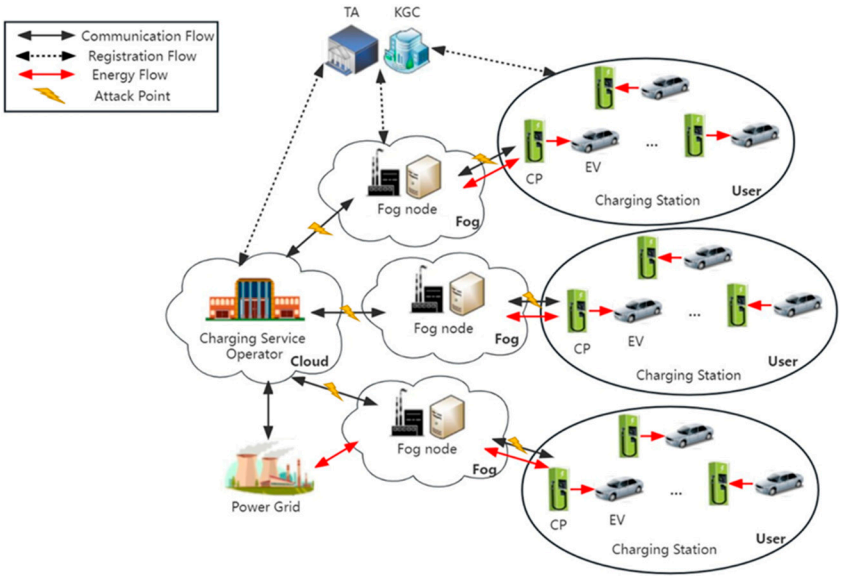

Figure 1. The communication architecture in the V2G network.

(1) TA: The TA is responsible for the registration of entities such as the EV, CP, and FN and tracking the real identities of EV users. The TA is a completely trustworthy entity.

(2) KGC: The KGC is responsible for generating public and private keys for entities such as the CP, FN, and CSO. The KGC is a partially trusted entity.
(3) CSO: The CSO, located in the cloud, is responsible for batch verification and decryption of the regional charging and discharging data reports uploaded by the FN, as well as processing the charging and discharging data. If the CSO detects abnormal charging and discharging data for the EV, it can request the TA to track the real identity of the EV user.
(4) FN: The FN is deployed at the level of the charging stations, with certain computing, communication, and storage capabilities. The FN is responsible for aggregating and verifying charging and discharging data reports, generating local regional charging and discharging data reports, and uploading them to the CSO, thereby avoiding the computational and communication overhead caused by direct data exchange between the CSO and each CP.
(5) CP: The CP is responsible for encrypting and signing the charging and discharging data of the EVs, generating charging and discharging data reports, and uploading them to the local FN. It is also responsible for providing power connections to the EV and charging or discharging the EV based on charging and discharging control commands issued by the CSO.
(6) EV: A vehicle with energy storage capacity, capable of bidirectional data communication and power transmission, is charged and discharged through the CP under the control of the FN, regulating the load on the power grid.

2.2. Threat Model

In this model, the connection between the TA and other entities is conducted through secure channels, while the connection to other public communication networks is not secure. According to the attack points marked in Figure 1, external adversaries may attempt to launch attacks such as eavesdropping, forgery, tampering, replay, and denial of service. Attackers may also attempt to impersonate legitimate EVs, CPs, or FNs. In this scheme, only the TA is a fully trusted entity, while the KGC is not fully trusted and may be vulnerable to malicious attacks or colluding with malicious attackers to cause a key leakage. The other entities, the CP, FN, and CSO, are honest and curious, and they honestly execute this plan, but also show curiosity about the EV's charging and discharging data or true identity.

Therefore, we consider two attackers in the threat model, namely, the external attacker A_I and the internal attacker A_{II}. A_I can query and tamper with the public key of any legitimate user but cannot obtain the master key; A_{II} represents a malicious KGC that can obtain the master key but cannot tamper with any user's public key.

2.3. Safety Objectives

To ensure the secure upload of large-scale charging and discharging privacy data in V2G networks, the proposed privacy protection scheme should meet the following security objectives:

(1) Conditional anonymity: The real identity of EV users must be kept confidential. However, if necessary, the TA can track the real identity of malicious EV users for accountability.
(2) Confidentiality: Charging and discharging data should be kept confidential to ensure that attackers cannot eavesdrop on plaintext data during communication.
(3) Unforgeability: Ensure that attackers cannot forge CP/FN uploaded charging and discharging data reports.
(4) Public verifiability: The signcryption can be verified through public information.
(5) Resistance to attack: In addition to the eavesdropping, forgery, and tampering mentioned above, the scheme must also resist impersonation attacks, replay attacks, and so on.

3. Implementation

The process of this scheme is shown in Figure 2, which includes system initialization, entity key generation, data report generation, aggregation verification and signature, aggregation verification and decryption, and identity tracking. Table 1 shows the symbols involved in the proposed scheme.

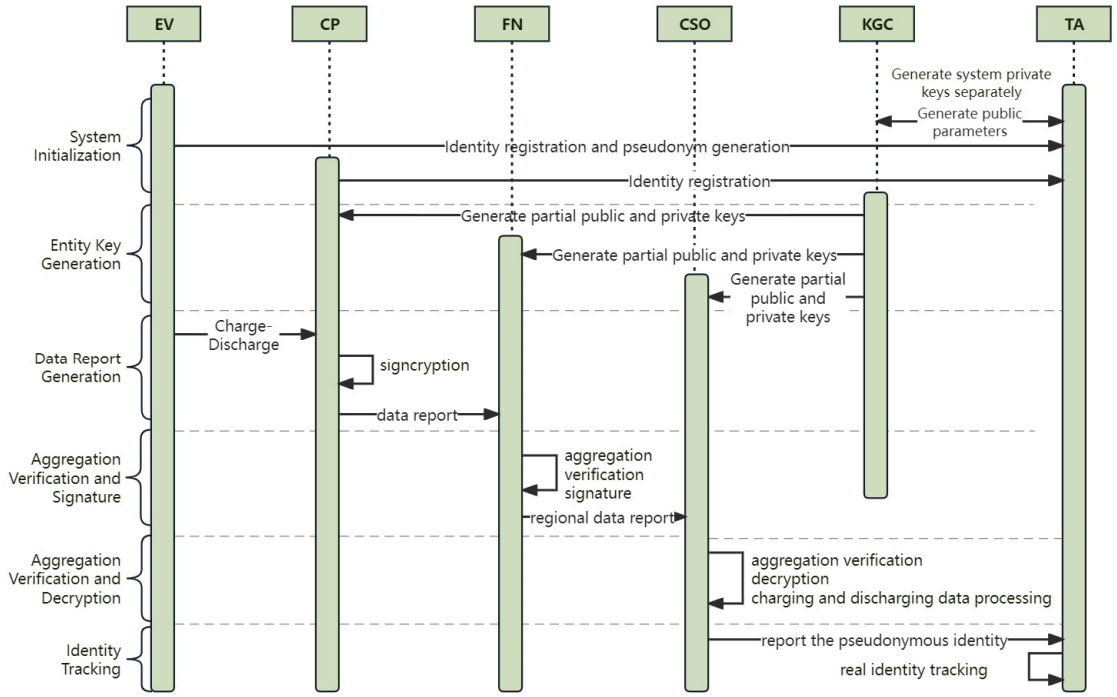

Figure 2. The process of the proposed scheme.

Table 1. Symbolic meanings of the proposed scheme.

Symbols	Meaning
V	System security parameter
q	Sufficient large prime number
s,t	System master key, $s \in Z_q^*$, $t \in Z_q^*$
P_{pub},T_{pub}	System public key, $P_{pub} = sP$, $T_{pub} = tP$
H_i	Secure hash function, i = 0, 1, 2, 3, 4
RID_i	Real identity of EV_i
PID_i	Pseudonym of EV_i
$\mathrm{ID}_{\mathrm{CP}i}$	Real identity of CP_i
$\mathrm{ID}_{\mathrm{FN}w}$	Real identity of FN_w

3.1. System Initialization

3.1.1. System Parameter Setting

Security parameter V is input, a group G is chosen on an elliptic curve, of prime order q, with generator P.

The TA initializes: a random number $t \in Z_q^*$ is chosen as the system master key, the public key $T_{pub} = tP$ is computed, and a hash function $H_0 : G \times \{0,1\}^* \rightarrow Z_q^*$ is chosen.

The KGC initializes: a random number $s \in Z_q^*$ is chosen as the system master key, the system public key $P_{pub} = sP$ is computed, and hash functions $H_1 : \{0,1\}^* \times G \times G \rightarrow Z_q^*$, $H_2 : \{0,1\}^* \times G \times G \times G \times \{0,1\}^* \rightarrow Z_q^*$, $H_3 : \{0,1\}^* \times \{0,1\}^* \times G \times G \times G \times G \rightarrow \{0,1\}^l$ are chosen, where l represents the length of plaintext or ciphertext messages.

The KGC and TA are independent of each other, each keeping their master keys s and t. They publish the system parameters $params = \{G, P, q,\ P_{pub}, T_{pub}, H_0, H_1, H_2, H_3\}$.

3.1.2. User Pseudonym Generation

When the EV, CP, FN, and CSO first join the V2G network, they must register with the TA using real identity information, and the TA then generates unique identifiers RID_i, ID_{CP_i}, ID_{FN_w}, $ID_{CSO} \in \{0,1\}^*$, respectively.

The process of generating pseudonyms is as follows: first, EV_i chooses a random number $t_i \in Z_q^*$, computes part of the pseudonym $PID_{i,1} = t_iP$, simultaneously computes $k_i = t_iT_{pub} \oplus RID_i$, and sends the request $(PID_{i,1}, k_i)$ to the TA to generate the pseudonym. Then, the TA verifies the identity of EV_i by computing the equation $RID_i = k_i \oplus tPID_{i,1}$; if the verification fails, the pseudonym generation request is discarded. Otherwise, the TA computes the other part of the pseudonym $PID_{i,2} = RID_i \oplus H_0(tPID_{i,1}, T_i)$ and secretly sends the complete pseudonym $PID_i = (PID_{i,1}, PPID_{i,2}, T_i)$ to EV_i, where T_i is the pseudonym's valid timestamp.

When EV_i's pseudonym expires, i.e., when the pseudonym's valid timestamp T_i is less than the current timestamp, EV_i must repeat the above process to request for the generation of a new pseudonym from the TA.

3.2. Entity Key Generation

CP_i selects a random number $x_i \in Z_q^*$ as the partial private key, calculates the partial public key $X_i = x_iP$, and then sends its identity ID_{CP_i} and partial public key X_i to the KGC.

After verifying the validity of CP_i's identity, the KGC selects a random number $y_i \in Z_q^*$, generates CP_i's partial public key $Y_i = y_iP$, calculates $h_{1i} = H_1(ID_{CPi},\ X_i,\ Y_i)$, calculates $d_i = y_i + s \cdot h_{1i}$ as CP_i's partial private key, and then sends $(Y_i,\ d_i)$ to CP_i through a secure channel.

After receiving the partial public and private keys, CP_i verifies the validity of the partial public and private keys through equation $d_iP = Y_i + H_1(ID_{CPi},\ X_i,\ Y_i) \cdot P_{pub}$ to prevent malicious KGC attacks. If the verification fails, it will be discarded directly, and part of the public and private keys will be requested again. Otherwise, CP_i's private key $SK_i = (x_i,\ d_i)$ and public key $PK_i = (X_i,\ Y_i)$.

The key generation process of FN_w and the CSO is similar to CP_i, after authentication using identity identifiers ID_{FN_w} and ID_{CSO}, and the generated key pairs are $SK_w = (x_w,\ d_w)$, $PK_w = (X_w,\ Y_w)$ and $SK_c = (x_c,\ d_c)$, $PK_c = (X_c,\ Y_c)$, respectively.

3.3. Data Report Generation

CP_i performs signcryption on the EV_i's charging and discharging data m_i. Subsequently, based on the signcrypted message δ_i, a data report P_i is generated and uploaded to the respective local FN_w.

The CP first selects a random number $r_i \in Z_q^*$ and generates C_i according to the following formula:

$$\begin{cases} R_i = r_iP \\ h_{1c} = H_1(ID_{CSO}, X_c, Y_c) \\ U_i = r_i(X_c + Y_c + P_{pub}h_{1c}) \\ h_{3i} = H_3(PID_i, ID_{CPi}, X_i, Y_i, U_i, R_i) \\ C_i = m_i \oplus h_{3i} \end{cases} \tag{1}$$

Then, according to the following formula, a signature s_i is generated, and the final signature message $\delta_i = (R_i, C_i, s_i)$ is obtained.

Subsequently, the charging and discharging data report P_i is uploaded to the local FN_w, where $P_i = \{PID_i, ID_{CP_i}, \delta_i, PK_i, TS_i\}$. Here, TS_i is a timestamp used to prevent replay attacks.

$$\begin{cases} h_{2i} = H_2(ID_{CP_i}, X_c, Y_c, R_i, C_i, TS_i) \\ s_i = d_i + x_ih_{2i} \end{cases} \tag{2}$$

3.4. Aggregation Verification and Signature

FN_w will receive multiple charging and discharging data reports uploaded by the local CP. If these signcryption messages are verified individually, it will significantly impact the response time of charging and discharging, which is unsuitable for large-scale EV charging and discharging scenarios. Additionally, if FN_w uploads these data reports directly to the CSO without aggregation, a significant communication and computation overhead will result. Therefore, in this scheme, FN_w chooses to perform an aggregated verification on these data reports, re-sign the aggregated signcryption messages, generate regional data reports, and then upload them to the CSO. This approach effectively utilizes the computational resources of the FN, reduces the computational burden on the CSO, and enhances the service quality of the V2G network. The specific process is as follows:

Firstly, after receiving the data report P_i uploaded by CP_i, FN_w performs initial verification by checking the validity of the timestamp TS_i in the data report and verifying the validity of the EV_i's pseudonym through the pseudonym timestamp T_i. If the verification fails, the data report P_i uploaded by CP_i is discarded.

Secondly, after successfully performing the initial verification on all n data reports P_i $(i = 1, 2, \ldots, n)$, FN_w proceeds to conduct aggregated verification according to the following formula. Upon successful verification, the aggregated signcryption message $\varphi_w = (R_1, R_2 \ldots R_n, C_1, C_2 \ldots C_n, S)$ is obtained.

$$\begin{cases} h_{1i} = H_1(PID_i, X_i, Y_i) \\ h_{2i} = H_2(ID_{CP_i}, X_c, Y_c, R_i, C_i, TS_i) \\ S = \sum_{i=1}^{n} s_i \\ SP = \sum\limits_{n=1}^{n} Y_i + P_{pub} \sum\limits_{n=1}^{n} h_{1i} + \sum\limits_{n=1}^{n} X_i h_{2i} \end{cases} \tag{3}$$

Thirdly, FN_w generates a signature sig_w using its private key based on the aggregated signcryption message φ_w according to the following formula.

$$\begin{cases} h_{2w} = H_2(ID_{FN_w}, X_c, Y_c, \varphi_w, TS_w) \\ sig_w = d_w + x_w h_{2w} \end{cases} \tag{4}$$

Finally, FN_w generates the data report $T_w = \{PID_i, ID_{CP_i}, ID_{FN_w}, \varphi_w, sig_w, PK_w, TS_w\}$ for its local region and uploads it to the CSO, where TS_w represents the current timestamp of the FN.

3.5. Aggregation Verification and Decryption

After receiving the regional data reports T_w uploaded by FN_w, the CSO first checks the validity of the timestamp TS_w. Subsequently, the CSO performs aggregated verification on the m valid regional data reports T_w $(w = 1, 2, \ldots, m)$ according to the following formula.

$$\begin{cases} h_{2w} = H_2(ID_{FN_w}, X_c, Y_c, \varphi_w, TS_w) \\ P \sum\limits_{m=1}^{m} sig_w = \sum\limits_{m=1}^{m} Y_w + P_{pub} \sum\limits_{n=1}^{m} h_{1w} + \sum\limits_{m=1}^{m} X_w h_{2w} \end{cases} \tag{5}$$

After successful aggregated verification, the CSO proceeds to decrypt and verify the regional data reports T_w $(w = 1, 2, \ldots, m)$ in sequence using the following formula, obtaining the complete charging and discharging data m_i uploaded by legitimate CP_i.

$$\begin{cases} U_i = (x_c + d_c) R_i \\ h_{3i} = H_3(PID_i, ID_{CP_i}, X_i, Y_i, U_i, R_i) \\ m_i = C_i \bigoplus h_{3i} \end{cases} \tag{6}$$

3.6. Identity Tracking

When the CSO processes the charging and discharging data of EVs and detects any abnormal charging and discharging data for the EV, it can request the TA to trace the real identity of EV_i. The process is as follows:

The CSO sends the pseudonym $PID_i = (PID_{i,1}, PID_{i,2}, T_i)$ to the TA for identity tracing. The TA calculates $RID_i = PID_{i,2} \oplus H_0(t\ PID_{i,1},\ T_i)$ to obtain the true identity of EV_i within constant time. Finally, the illegal EV user will be punished by the TA.

4. Correctness and Security Analysis

4.1. Correctness Analysis

The correctness of EV user pseudonym generation and identity tracking is proven as follows:

$$tPID_{i,1} = tt_iP = t_iT_{pub} \tag{7}$$

The FN can complete aggregation verification according to the following equation.

$$\begin{aligned} SP &= \sum_{i=1}^{n} s_iP \\ &= \sum_{i=1}^{n} (d_i + x_ih_{2i})P \\ &= \sum_{i=1}^{n} (y_i + s\ h_{1i} + x_ih_{2i})P \\ &= \sum_{i=1}^{n} Y_i + P_{pub}\sum_{i=1}^{n} h_{1i} + \sum_{i=1}^{n} X_ih_{2i} \end{aligned} \tag{8}$$

The CSO can complete aggregation verification and decryption successfully according to the following equation.

$$\begin{aligned} P\sum_{i=1}^{m} sig_w &= P\sum_{i=1}^{m} (d_w + x_wh_{2w}) \\ &= P\sum_{i=1}^{m} (y_w + s\ h_{1w} + x_wh_{2w}) \\ &= \sum_{i=1}^{m} Y_w + P_{pub}\sum_{i=1}^{m} h_w + \sum_{i=1}^{m} X_wh_{2w} \end{aligned} \tag{9}$$

$$\begin{aligned} U_i &= r_i(X_c + Y_c + P_{pub}h_{1c}) \\ &= r_i(x_cP + y_cP + sP\ h_{1c}) \\ &= r_i(x_c + y_c + s\ h_{1c})P \\ &= (x_c + d_c)R_i \end{aligned} \tag{10}$$

4.2. Security Analysis

The security features of the relevant representative schemes (i.e., Wang et al. [13], Zhang et al. [25], Basudan et al. [27], and Dohare et al. [29]) are compared as shown in Table 2, proving that our scheme's security surpasses those of the others. The symbol $\surd$ indicates compliance with the security feature, while $\times$ indicates non-compliance.

Table 2. Comparison of security features.

Scheme	Conditional Anonymity	Confidentiality	Unforgeability	Public Verifiability	Resistance to Replay Attacks	Resistance to DDoS Attacks
[13]	$\surd$	$\surd$	$\surd$	$\times$	$\surd$	$\times$
[25]	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\times$
[27]	$\times$	$\surd$	$\surd$	$\times$	$\times$	$\times$
[29]	$\times$	$\surd$	$\surd$	$\surd$	$\times$	$\surd$
Ours	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$

4.2.1. Conditional Anonymity

In this scheme, only the trusted entity TA knows the EV users' real identities, while other entities only know the pseudonymous identities. Apart from the TA, no one can deduce the real identity of the EV user from the pseudonymous identity. This ensures the conditional anonymity of the EV user.

4.2.2. Confidentiality

Theorem 1 (Confidentiality under Adversary A_I). *In the case of a stochastic prediction model and ECDHP difficulty, adversary A_I can win IND-CCA2 with a non-negligible advantage ε_{11}, then there exists a challenger C who can solve the ECDHP difficulty problem with at least a non-negligible probability $\left(1-\frac{q_{sk}}{2^k}\right)\left(1-\frac{q_3}{2^k}\right)\frac{\varepsilon_{11}}{en(q_s+q_{sk}+1)}$ in finite polynomial time, where e is the base of the natural logarithm and k is a safety parameter.*

Proof of Theorem 1. Challenger C is given an ECDHP challenge instance (P, aP, bP), where $a, b \in Z_q^*$ and its values are all unknown, and C's goal is to calculate the value abP by adversary A_I. □

Initial stage: C performs system initialization, generates system parameters, and sends them to adversary A_I, who cannot obtain system master key s. C randomly chooses ID_i^* as the challenger. In addition, C maintains five lists to record query data extracted by A_I from oracle H_1, H_2, H_3, partial private key, and public key, respectively. All lists are initialized to empty.

Query stage: Adversary A_I executes a polynomial bounded query as follows:

H_1 query: Challenger C maintains a list of $L_1 = (ID_{CP_i}, X_i, Y_i, h_{1i})$, and when receiving an H_1 query (ID_{CP_i}, X_i, Y_i) from adversary A_I, if the inquiry already exists in the list, returns the corresponding h_{1i} to A_I. Otherwise, challenger C randomly selects $h_{1i} \in Z_q^*$ and returns to A_I, and adds the item $(ID_{CP_i}, X_i, Y_i, h_{1i})$ to the list L_1.

H_2 query: Challenger C maintains a list of $L_2 = (ID_{CP_i}, X_c, Y_c, R_i, C_i, TS_i, h_{2i})$, and when receiving an H_2 query $(ID_{CP_i}, X_c, Y_c, R_i, C_i, TS_i)$ from adversary A_I, if the inquiry already exists in the list, returns the corresponding h_{2i} to A_I. Otherwise, challenger C randomly selects $h_{2i} \in Z_q^*$ and returns to A_I, and adds the item $(ID_{CP_i}, X_c, Y_c, R_i, C_i, TS_i, h_{2i})$ to the list L_2.

H_3 query: Challenger C maintains a list of $L_3 = (PID_i, ID_{CP_i}, X_i, Y_i, U_i, R_i, h_{3i})$, and when receiving an H_3 query $(PID_i, ID_{CP_i}, X_i, Y_i, U_i, R_i)$ from adversary A_I, if the inquiry already exists in the list, returns the corresponding h_{3i} to A_I. Otherwise, C randomly selects $h_{3i} \in Z_q^*$ and returns to A_I, and adds the item $(PID_i, ID_{CP_i}, X_i, Y_i, U_i, R_i, h_{3i})$ to the list L_3.

Partial private key query: When challenger C receives a query from A_1 about ID_i, if $ID_i = ID_i^*$, the simulation operation is terminated. If $ID_i \neq ID_i^*$, challenger C queries the list L_{psk}, and if there is a corresponding item, returns (d_i, Y_i) to A_I. Otherwise, challenger C randomly selects $a_i, b_i \in Z_q^*$, so that $d_i = a_i$, $H_1(ID_i, X_i, Y_i) = b_i$, then $Y_i = a_iP - b_iP_{pub}$. C adds (ID_i, X_i, Y_i, b_i) and (ID_i, d_i, Y_i) to L_1 and L_{psk}, respectively, and returns (d_i, Y_i) to A_I.

Create user query: When challenger C receives a query from A_I about ID_i, C queries the list L_{user}. If the list L_{user} contains $(ID_i, x_i, d_i, X_i, Y_i)$, C returns $PK_i = (X_i, Y_i)$ to A_I; Otherwise, if $ID_i = ID_i^*$, challenger C randomly selects $x_i \in Z_q^*$, so that $X_i = x_iP$, calculates $Y_i = (1 - h_{1i})P_{pub}$, adds $(ID_i, x_i, \perp, X_i, Y_i)$ to the list L_{user}, and returns $PK_i = t(X_i, Y_i)$ to A_I. If $ID_i \neq ID_i^*$, C randomly selects $x_i, y_i \in Z_q^*$, calculates $X_i = x_iP$, $Y_i = y_iP - h_{1i}P_{pub}$, then adds $(ID_i, x_i, \perp, X_i, Y_i)$ to the list of L_{user}, and returns $PK_i = (X_i, Y_i)$ to A_I.

Secret value query: When challenger C receives a query about ID_i, if $ID_i = ID_i^*$, the simulation operation is terminated. Otherwise, C queries the list L_{user}, and if there is a corresponding item, returns the secret value x_i to A_I. If it does not exist, C executes the creation of a user inquiry to generate $(ID_i, x_i, \perp, X_i, Y_i)$ and adds it to L_{user}, then returns x_i to A_I.

Change public key query: When challenger C receives a query (ID_i, X_i', Y_i') from A_I, that is, A_I wants to replace the old $PK_i = (X_i, Y_i)$ with a new $PK_i' = (X_i', Y_i')$, assuming that C has already submitted a create user query. Then, C obtains $(ID_i, x_i, d_i, X_i, Y_i)$ from the list L_{user}, updates X_i to X_i', updates Y_i to Y_i', and sets $x_i = \perp, d_i = \perp$, thus $(PID_i, x_i, d_i, X_i, Y_i)$ in L_{user} has been updated to $(PID_i, \perp, \perp, X_i', Y_i')$.

Signcryption query: When challenger C receives a query (m_i, ID_i, ID_c) from A_I, where m_i is a plaintext message, ID_i is the sender, and ID_c is the receiver, the following processing is performed: If $ID_i \neq ID_i^*$, C obtains the receiver's $PK_w = (X_c, Y_c)$ and the sender's $SK_i = (x_i, d_i)$ by querying L_{user}, and then performs the signcryption operation on m_i according to the scheme. If $ID_i = ID_i^*$, C queries L_1 to obtain $h_{1c} = H_1(ID_c, X_c, Y_c)$, queries L_{user} to obtain the receiver's private key $SK_c = (x_c, d_c)$, selects the random number $r_i \in Z_q^*$, calculates $R_i = r_i P$, $U_i = (x_c + d_c t)R_i$, $C_i = m_i \oplus H_3(PID_i, ID_{CP_i}, X_i, Y_i, U_i, R_i)$. $(PID_i, ID_{CP_i}, X_i, Y_i, U_i, R_i)$ and $(ID_{CP_i}, X_c, Y_c, R_i, C_i, TS_i)$ will be stored in list L_3 and list L_2, respectively. Finally, C calculates s_i to make $s_i P = Y_i + P_{pub} h_{1i} + X_i h_{2i}$ true and return $\delta_i = (R_i, C_i, s_i)$.

Aggregation verification query: When challenger C receives an aggregation signature query $(ID_1, ID_2 \ldots ID_n, m_1, m_2 \ldots m_n, ID_c)$ from A_I, C performs aggregation verification according to the scheme, verifying whether the equation $SP = \sum_{i=1}^{n} Y_i + P_{pub} \sum_{i=1}^{n} h_{1i} + \sum_{i=1}^{n} X_i h_{2i}$ is true. If it is true, C returns the aggregation signcryption $\varphi = (R_1, R_2 \ldots R_n, C_1, C_2 \ldots C_n, S)$.

Unsigncryption query: Challenger C receives an unsigncryption query $(ID_1, ID_2 \ldots ID_n, \varphi, ID_w)$ from A_I, and if $ID_i \neq ID_i^*$, C will perform the unsigncryption operation according to the scheme and return the plaintext message m_i. Otherwise, the game will be terminated. If $ID_i = ID_i^*$ or the public key of ID_i is replaced, C queries the lists L_2 and L_3. If there are corresponding tuples, it returns the plaintext message m_i; otherwise, the simulation stops.

Challenge stage: Adversary A_I randomly selects two messages of equal length, m_0 and m_1, and randomly selects two identities, ID_i^* and ID_r^*, where ID_r^* is the challenge identity. C selects $d \in (0, 1)$ and performs a signcryption query for m_d to obtain an aggregation signcryption $\varphi^* = (R_i^*, C_i^*, S^*)$ and sends it to A_I. After receiving φ^*, A_I continues to initiate a series of polynomial bounded queries, but A_I cannot perform partial private key queries of ID_r^* and the unsigncryption queries of φ^*.

Guessing stage: Through the various queries in the first stage, adversary A_I outputs d' as their own guess about d. If the guess is correct, challenger C outputs $d_i^* R_i^* = b(Y_i^* + h_{1i} P_{pub}) = b[(1 - h_{1i})P_{pub} + h_{1i} P_{pub}] = bP_{pub} = abP$ as the solution to ECDHP, where $R_i^* = bP$. From this, C solves the ECDHP problem with A_I.

Theorem 2 (Confidentiality under Adversary A_{II}). *In the case of a stochastic prediction model and ECDHP difficulty, adversary A_{II} can win IND-CCA2 with a non-negligible advantage ε_{12}, there exists C who can solve the ECDHP difficulty problem with at least a non-negligible probability* $\left(1 - \frac{q_{sk}}{2^k}\right)\left(1 - \frac{q_3}{2^k}\right)\frac{\varepsilon_{12}}{en(q_s + q_{sk} + 1)}$.

Proof of Theorem 2. Challenger C is given a random ECDHP challenge instance (P, aP, bP), where $a, b \in Z_q^*$ and its values are all unknown, and C's goal is to calculate the value abP by the adversary A_{II}. The proof process is similar to Theorem 1. □

In summary, challenger C has the ability to solve ECDHP, but this contradicts ECDHP, so the scheme satisfies confidentiality under the attack of adversaries A_I and A_{II}.

4.2.3. Unforgeability

Theorem 3 (Unforgeability under Adversary A_I). *In the case of a stochastic prediction model and ECDLP difficulty, adversary A_I can win EUF-CMA with a non-negligible advantage ε_{21}, there exists C who can solve ECDLP difficulty problem with at least a non-negligible probability* $\left(1 - \frac{q_{sk}}{2^k}\right)\frac{\varepsilon_{21}}{en(q_s + q_{sk} + 1)}$.

Proof of Theorem 3. Challenger C is given a random ECDLP challenge instance (P, aP), where $a \in Z_q^*$ and its value is unknown, and C's goal is to calculate the value a by the adversary A_I. □

Initial stage: Challenger C performs system initialization, generates system parameters, and sends them to adversary A_I, who cannot obtain system master key s.

Forgery stage: When adversary A_I submits aggregate signcryption $\varphi^* = (R_1^*, R_2^* \ldots R_n^*, C_1^*, C_2^* \ldots C_n^*, S^*)$ of n users' message m_i^*, Equation (11) holds if aggregate verification is valid. C forges an aggregate signcryption $\varphi' = (R_1^*, R_2^* \ldots R_n^*, C_1^*, C_2^* \ldots C_n^*, S')$ in the same way and sends it to A_I.

$$S^*P = \sum_{i=1}^{n} Y_i^* + P_{pub} \sum_{i=1}^{n} h_{1i}^* + \sum_{i=1}^{n} X_i^* h_{2i}^* \tag{11}$$

If A_I receives φ' and the aggregate signcryption verification passes, then Equation (12) is obtained by repeating the above steps and selecting a different H_1 according to a bifurcation lemma.

$$S'P = \sum_{i=1}^{n} Y_i^* + P_{pub} \sum_{i=1}^{j-1} h_{1i}^* + P_{pub} h'_{1j} + P_{pub} \sum_{i=j+1}^{n} h_{1i}^* + \sum_{i=1}^{n} X_i^* h_{2i}^* \tag{12}$$

After subtracting Equation (11) from Equation (12) to obtain the derived Formula (13), C outputs $a = \frac{S'-S^*}{h'_{1j}-h_{1j}^*}$ as the solution to the ECDLP problem. Therefore, C solves the ECDLP problem by A_I.

$$(S' - S^*)P = P_{pub} \sum_{i=1}^{j-1} h_{1i}^* + P_{pub} h'_{1j} + P_{pub} \sum_{i=j+1}^{n} h_{1i}^* - P_{pub} \sum_{i=1}^{n} h_{1i}^* = (h'_{1j} - h_{1j}^*) P_{pub} = (h'_{1j} - h_{1j}^*) aP \tag{13}$$

Theorem 4 (Unforgeability under Adversary A_{II}). *In the case of a stochastic prediction model and ECDLP difficulty, adversary A_{II} can win EUF-CMA with a non-negligible advantage ε_{22}, there exists C who can solve the ECDLP difficulty problem with at least a non-negligible probability* $\left(1 - \frac{q_{sk}}{2^k}\right) \frac{\varepsilon_{22}}{en(q_s+q_{sk}+1)}$.

Proof of Theorem 4. Challenger C is given a random ECDHP challenge instance (P, aP), where $a \in Z_q^*$ and its value is unknown, and C's goal is to calculate the value a by the adversary A_{II}. The proof process is similar to Theorem 3 and will not be repeated here. □

In summary, challenger C has the ability to solve ECDLP, but this contradicts ECDLP, so the scheme satisfies unforgeability under the attack of adversaries A_I and A_{II}.

4.2.4. Public Verifiability

Aggregate verification operations rely on the public parameters, without requiring the private keys of the sender CP/FN/CSO or the receiver. Any entity can verify the validity of the data reports.

4.2.5. Resistance to Impersonation Attacks

Through aggregate verification, the FN/CSO can confirm the legitimacy of the sender CP/FN's identity, making it impossible for attackers to forge the sender's legitimate signature or impersonate the sender's valid identity.

4.2.6. Resistance to Replay Attacks

This attack involves retransmitting previously eavesdropped data reports to the receiver without modifying the content. The CP/FN adds timestamps before sending mes-

sages to indicate their timeliness. By checking the timestamps, the receiver FN/CSO can resist replay attacks.

4.2.7. Resistance to Distributed Denial of Service (DDoS) Attacks

The communication architecture of this scheme is based on cloud–fog collaboration, where introducing fog devices between the cloud and user layers enables distributed computation and storage of fog computing, mitigating issues such as vulnerable transmission distance and susceptibility to DDoS attacks in traditional cloud computing.

5. Performance Evaluation

In this section, we evaluate the performance of the proposed scheme. Taking the example of m FNs, where each fog node region contains n EVs, we analyze the computational and communication overheads of the proposed scheme. Furthermore, we also compare our scheme with other related CLASC schemes (i.e., Basudan et al. [27], Wang et al. [28], Dai et al. [24], and Zhang et al. [25]). The first two schemes are based on fog computing architecture, while the latter two are based on traditional architecture.

5.1. Computation Cost

In our experiments, we conducted the operations on our personal computer configured with an Intel Core I5-4210H 1.80GHz processor, 8GB RAM, and Windows 10. We utilized the Pairing-Based Cryptography (PBC) library to perform these operations, and the running times for different operations are shown in Table 3. The table only includes the computationally expensive operations, among which the bilinear pairing operation is significantly more costly than the multiplication operation.

Table 3. Execution times of related operations.

Symbol	The Run Time of Operation (ms)	Meaning
T_{p}	4.2846	Time for a bilinear pairing operation
T_{pm}	0.4720	Time for point multiplication operation in elliptic curve
T_{sm}	0.2530	Time for scalar multiplication operation
T_{h}	3.8643	Time for map to point hash

T_p is calculated by using bilinear pairs $e : G_1 \times G_1 \rightarrow G_2$. G_1 of order q is generated by elliptic curves $\mathrm{E}(F_{\mathrm{p}'})$: $y^2 = x^3 + x$ defined over finite field $F_{p'}$. T_{pm} is calculated from elliptic curves $E(F_p) : y^2 = x^3 + ax + b$ defined over finite fields F_p.Table 4 shows the computation cost comparison among CLASC schemes. In this scheme, each CP requires $3T_{pm} + T_{sm}$ for signcryption of m_i to generate the charging and discharging data report P_i. Therefore, the computation cost for each CP is $3T_{pm} + T_{sm}$. Each FN requires $(n+2)T_{pm}$ for aggregating and verifying n received data reports $P_i (n = 1, 2, \ldots, n)$, and T_{sm} is used for generating the corresponding signature. Hence, the total computation cost for each FN is $(n+2)T_{pm} + T_{sm}$. The CSO requires $(m+2)T_{pm}$ for aggregating and verifying m received area data reports $T_w (w = 1, 2, \ldots, m)$ and mnT_{pm} for decrypting mn encrypted messages. The total computation cost of the CSO is $(mn + m + 2)T_{pm}$.

Table 4. Computation cost comparison among CLASC schemes.

Scheme	Signcryption (CP)	Verification (FN)	Verification and Unsigncryption (CSO)
[27]	$2T_h + 7T_{pm}$	$2nT_h + 4nT_p + nT_{pm}$	$2mnT_h + (mn+3)T_p + 3mnT_{pm}$
[28]	$2T_h + 6T_{pm}$	$(n+1)T_h + 3nT_p + 2nT_{pm}$	$(mn+1)T_h + (mn+2)T_p + 4mnT_{pm}$
[24]	$3T_{pm} + 4T_{sm}$	—	$(3mn+1)T_{pm}$
[25]	$5T_{pm} + 2T_{sm}$	$(3n+1)T_{pm}$	$2mnT_{pm}$
Ours	$3T_{pm} + T_{sm}$	$(n+2)T_{pm} + T_{sm}$	$(mn+m+2)T_{pm}$

Figure 3 shows the computation costs of CP signcryption among the relevant CLASC schemes. The computation cost of FN verification compared with other schemes is shown in Figure 4. We can see that neither [24,25] nor our scheme involves bilinear mapping operations, resulting in higher execution efficiency compared to [27,28]. With the increase in the number of charging and discharging EVs, the computation overhead of the FN becomes lower and lower compared with other schemes. When $n = 120$, charging and discharging data reports are uploaded in the local area, and the total aggregation verification time of the FN is 57.837 ms, which is a very effective and reasonable time assumption for FNs with certain computing abilities.

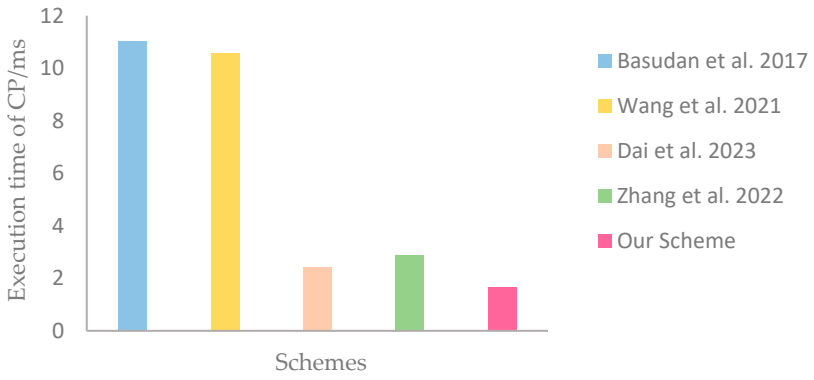

Figure 3. Comparison of computation costs for signcryption (CP) [24,25,27,28].

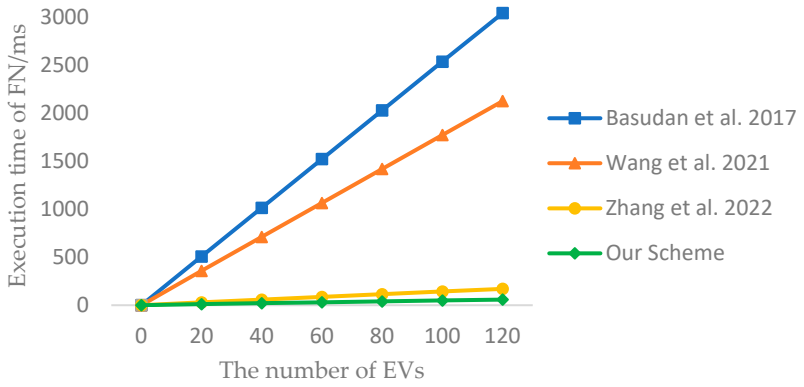

Figure 4. Comparison of computation costs for verification (FN) [25,27,28].

Figure 5 shows the computation cost of the CSO in this scheme with the increase in the number of FNs m and the number of EVs in the local region n. Assuming that the number of FNs $m = 1$, the comparison of computation costs of the CSO is shown in Figure 6. It can be seen that the advantages of our scheme are gradually reflected in the increase in the number of FNs and the number of EVs in the local region. In summary, the CLASC scheme designed in this paper has lower computational overhead compared to several other schemes, resulting in higher execution efficiency.

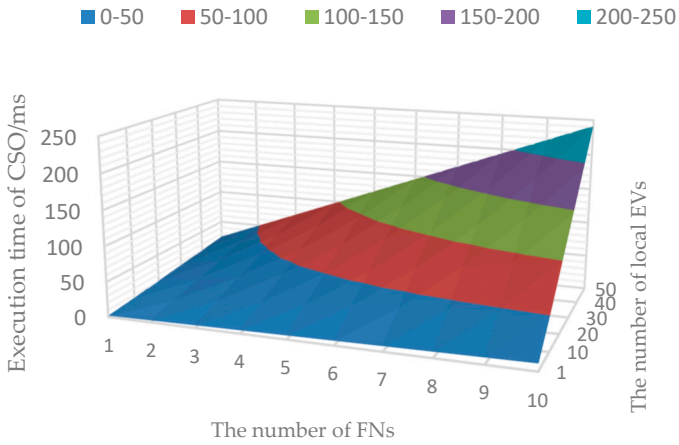

Figure 5. Computation cost of the CSO in this scheme.

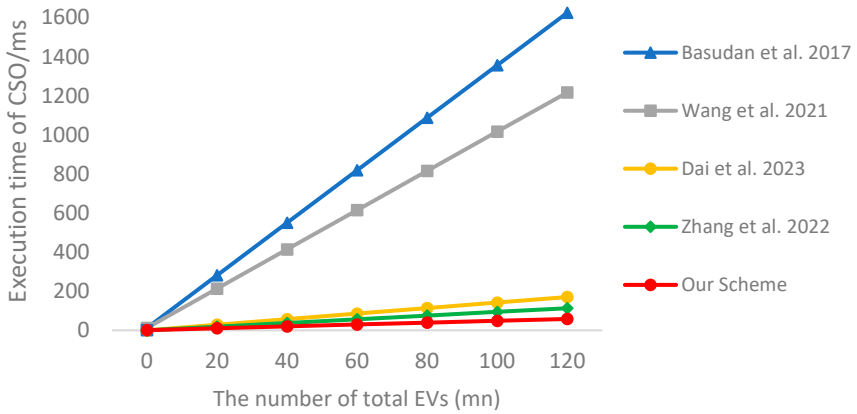

Figure 6. Comparison of computation costs for the CSO [24,25,27,28].

5.2. Communication Cost

In order to analyze the communication overhead of each scheme under the same conditions, we assume that the elements in Z_q^* are the same length as the messages, both 160 bits, $|Z_q^*| = 160$ bits, $|m| = 160$ bits. The bilinear pairing group is $|G_1| = 2|p'| = 1024$ bits, and the elliptic curve group is $|G| = 2|p| = 320$ bits.

Table 5 compares the signcryption output length as the communication cost with other schemes. It can be seen that our scheme and [24] have the lowest communication overhead, both of which are 640 bits. Compared with [27], our scheme is reduced by $(2208 - 640)/2208 = 71.01\%$; compared with [28], it is reduced by $(4416 - 640)/4416 = 85.51\%$; compared with [25], it is reduced by $(960 - 640)/960 = 33.33\%$. Therefore, our scheme has lower communication overhead than the others and further saves the bandwidth of V2G networks.

Table 5. Communication cost comparison among CLASC schemes.

Scheme	Length of Signcryption (Bits)	Length of Aggregate Signcryption (Bits)
[27]	$2\|G_1\| + \|m\| = 2208$	$(n+1)\|G_1\| + n\|m\| = 1184n + 1024$
[28]	$\left\|Z_q^*\right\| + 4\|G_1\| + \|m\| = 4416$	$n\left\|Z_q^*\right\| + \mathrm{n}\|m\| + (2\mathrm{n}+2)\|G_1\| = 2368n + 2048$
[24]	$\left\|Z_q^*\right\| + \|G\| + \|m\| = 640$	$\left\|Z_q^*\right\| + n\|G\| + n\|m\| = 480\mathrm{n} + 160$
[25]	$\left\|Z_q^*\right\| + 2\|G\| + \|m\| = 960$	$\left\|Z_q^*\right\| + 2n\|G\| + n\|m\| = 800\mathrm{n} + 160$
Ours	$\left\|Z_q^*\right\| + \|G\| + \|m\| = 640$	$\left\|Z_q^*\right\| + n\|G\| + n\|m\| = 480\mathrm{n} + 160$

6. Conclusions

The V2G network faces serious security issues and formidable privacy protection challenges. To ensure the security and efficiency of large-scale charging and discharging data transmission in the V2G network, a cloud–fog-based V2G network architecture is designed, and a charging and discharging data privacy protection scheme is proposed. The proposed scheme achieves anonymity and the traceability of EV users' identities through a pseudonym mechanism. The designed CLASC algorithm guarantees the security of uploading charging and discharging privacy data.

The security analysis determined that the proposed scheme not only meets the required security features, including conditional anonymity, confidentiality, and unforgeability, but can also resist common attacks such as impersonation, replay, and distributed denial of service (DDoS). The experimental analysis indicated that the proposed scheme has high efficiency in both computation and communication, making it suitable for V2G network environments with limited resources. Further research can address the challenge of balancing security and efficiency and achieving rapid identity authentication between EVs and FNs before charging and discharging data transmission.

Author Contributions: Conceptualization, B.W. and Z.S.; methodology, S.Z.; simulation and analysis, Z.S.; writing—original draft preparation, Z.S.; supervision, B.W. and S.Z.; project administration, B.W.; funding acquisition, S.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Fundamental Research Funds for the Central Universities (2018ZD06).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article material and further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zhong, W.; Yu, R.; Xie, S.; Zhang, Y.; Yau, D.K.Y. On Stability and Robustness of Demand Response in V2G Mobile Energy Networks. *IEEE Trans. Smart Grid* **2018**, *9*, 3203–3212. [CrossRef]
2. Tushar, W.; Yuen, C.; Huang, S.; Smith, D.B.; Poor, H.V. Cost Minimization of Charging Stations with Photovoltaics: An Approach with EV Classification. *IEEE Trans. Intell. Transport. Syst.* **2016**, *17*, 156–169. [CrossRef]
3. Lo Franco, F.; Mandrioli, R.; Ricco, M.; Monteiro, V.; Monteiro, L.F.; Afonso, J.L.; Grandi, G. Electric Vehicles Charging Management System for Optimal Exploitation of Photovoltaic Energy Sources Considering Vehicle-to-Vehicle Mode. *Front. Energy Res.* **2021**, *9*, 716389. [CrossRef]
4. Le Goff Latimier, R.; Multon, B.; Ben Ahmed, H.; Baraer, F.; Acquitter, M. Stochastic Optimization of an Electric Vehicle Fleet Charging with Uncertain Photovoltaic Production. In Proceedings of the 2015 International Conference on Renewable Energy Research and Applications (ICRERA), Palermo, Italy, 22–25 November 2015; pp. 721–726.
5. Lazaroiu, C.; Roscia, M.; Saadatmandi, S. Finite Element Methodologies Application in EV's Charging Infrastructure Planning. In Proceedings of the 2020 International Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM), Sorrento, Italy, 24–26 June 2020; pp. 369–374.
6. Han, W.; Xiao, Y. Privacy Preservation for V2G Networks in Smart Grid: A Survey. *Comput. Commun.* **2016**, *91*, 17–28. [CrossRef]
7. Saxena, N.; Grijalva, S.; Chukwuka, V.; Vasilakos, A.V. Network Security and Privacy Challenges in Smart Vehicle-to-Grid. *IEEE Wirel. Commun.* **2017**, *24*, 88–98. [CrossRef]
8. Sureshkumar, V.; Mugunthan, S.; Amin, R. An Enhanced Mutually Authenticated Security Protocol with Key Establishment for Cloud Enabled Smart Vehicle to Grid Network. *Peer-to-Peer Netw. Appl.* **2022**, *15*, 2347–2363. [CrossRef]
9. Mukherjee, M.; Kumar, S.; Mavromoustakis, C.X.; Mastorakis, G.; Matam, R.; Kumar, V.; Zhang, Q. Latency-Driven Parallel Task Data Offloading in Fog Computing Networks for Industrial Applications. *IEEE Trans. Ind. Inform.* **2020**, *16*, 6050–6058. [CrossRef]
10. Zhu, L.; Li, M.; Zhang, Z.; Xu, C.; Zhang, R.; Du, X.; Guizani, N. Privacy-Preserving Authentication and Data Aggregation for Fog-Based Smart Grid. *IEEE Commun. Mag.* **2019**, *57*, 80–85. [CrossRef]
11. Gu, K.; Wu, N.; Yin, B.; Jia, W. Secure Data Sequence Query Framework Based on Multiple Fogs. *IEEE Trans. Emerg. Top. Comput.* **2021**, *9*, 1883–1900. [CrossRef]
12. Wu, T.-Y.; Guo, X.; Yang, L.; Meng, Q.; Chen, C.-M. A Lightweight Authenticated Key Agreement Protocol Using Fog Nodes in Social Internet of Vehicles. *Mob. Inf. Syst.* **2021**, *2021*, 3277113. [CrossRef]

13. Wang, X.; Liu, Y.; Choo, K.-K.R. Fault-Tolerant Multisubset Aggregation Scheme for Smart Grid. *IEEE Trans. Ind. Inform.* **2021**, *17*, 4065–4072. [CrossRef]
14. Chen, L.; Zhou, J.; Chen, Y.; Cao, Z.; Dong, X.; Choo, K.-K.R. PADP: Efficient Privacy-Preserving Data Aggregation and Dynamic Pricing for Vehicle-to-Grid Networks. *IEEE Internet Things J.* **2021**, *8*, 7863–7873. [CrossRef]
15. Yang, Q.; Li, D.; An, D.; Yu, W.; Fu, X.; Yang, X.; Zhao, W. Towards Incentive for Electrical Vehicles Demand Response with Location Privacy Guaranteeing in Microgrids. *IEEE Trans. Dependable Secur. Comput.* **2022**, *19*, 131–148. [CrossRef]
16. Abdallah, A.; Shen, X.S. Lightweight Authentication and Privacy-Preserving Scheme for V2G Connections. *IEEE Trans. Veh. Technol.* **2017**, *66*, 2615–2629. [CrossRef]
17. Zhang, Y.; Zou, J.; Guo, R. Efficient Privacy-Preserving Authentication for V2G Networks. *Peer-to-Peer Netw. Appl.* **2021**, *14*, 1366–1378. [CrossRef]
18. Xu, C.; Wu, H.; Liu, H.; Li, X.; Liu, L.; Wang, P. An Intelligent Scheduling Access Privacy Protection Model of Electric Vehicle Based on 5G-V2X. *Sci. Program.* **2021**, *2021*, 1198794. [CrossRef]
19. Yu, S.; Park, K. PUF-Based Robust and Anonymous Authentication and Key Establishment Scheme for V2G Networks. *IEEE Internet Things J.* **2024**, *11*, 15450–15464. [CrossRef]
20. Li, H.; Han, D.; Tang, M. A Privacy-Preserving Charging Scheme for Electric Vehicles Using Blockchain and Fog Computing. *IEEE Syst. J.* **2021**, *15*, 3189–3200. [CrossRef]
21. Xia, Z.; Fang, Z.; Gu, K.; Wang, J.; Tan, J.; Wang, G. Effective Charging Identity Authentication Scheme Based on Fog Computing in V2G Networks. *J. Inf. Secur. Appl.* **2021**, *58*, 102649. [CrossRef]
22. Zheng, Y. Digital Signcryption or How to Achieve Cost (Signature & Encryption) ≪ Cost (Signature)+ Cost (Encryption). In Proceedings of the Advances in Cryptology—CRYPTO'97: 17th Annual International Cryptology Conference, Santa Barbara, CA, USA, 17–21 August 1997; Springer: Berlin/Heidelberg, Germany, 1997; pp. 165–179.
23. Lu, H.; Xie, Q. An Efficient Certificateless Aggregate Signcryption Scheme from Pairings. In Proceedings of the 2011 International Conference on Electronics, Communications and Control (ICECC), Ningbo, China, 9–11 September 2011; pp. 132–135.
24. Dai, C.; Xu, Z. Pairing-Free Certificateless Aggregate Signcryption Scheme for Vehicular Sensor Networks. *IEEE Internet Things J.* **2023**, *10*, 5063–5072. [CrossRef]
25. Zhang, S.; Ma, M.; Wang, B. A Lightweight Privacy Preserving Scheme of Charging and Discharging for Electric Vehicles Based on Consortium Blockchain in Charging Service Company. *Int. J. Electr. Power Energy Syst.* **2022**, *143*, 1084–1095. [CrossRef]
26. Cui, M.; Han, D.; Wang, J. An Efficient and Safe Road Condition Monitoring Authentication Scheme Based on Fog Computing. *IEEE Internet Things J.* **2019**, *6*, 9076–9084. [CrossRef]
27. Basudan, S.; Lin, X.; Sankaranarayanan, K. A Privacy-Preserving Vehicular Crowdsensing-Based Road Surface Condition Monitoring System Using Fog Computing. *IEEE Internet Things J.* **2017**, *4*, 772–782. [CrossRef]
28. Wang, W.; Wu, L.; Qu, W.; Liu, Z.; Wang, H. Privacy-Preserving Cloud-Fog-Based Traceable Road Condition Monitoring in VANET. *Int. J. Netw. Manag.* **2021**, *31*, e2096. [CrossRef]
29. Dohare, I.; Singh, K.; Ahmadian, A.; Mohan, S.; Kumar Reddy, M.P. Certificateless Aggregated Signcryption Scheme (CLASS) for Cloud-Fog Centric Industry 4.0. *IEEE Trans. Ind. Inf.* **2022**, *18*, 6349–6357. [CrossRef]