symmetry

MDPI

Article

A 3C Authentication: A Cross-Domain, Certificateless, and Consortium-Blockchain-Based Authentication Method for Vehicle-to-Grid Networks in a Smart Grid

Qianhao Miao [1], Tianyu Ren [2], Jiahan Dong [2], Yanjiao Chen [1] and Wenyuan Xu [1,*]

[1] College of Electrical Engineering, Zhejiang University, Hangzhou 310027, China; qhmiao@zju.edu.cn (Q.M.); chenyanjiao@zju.edu.cn (Y.C.)
[2] State Grid Beijing Electric Power Research Institute, Beijing 100075, China; rentianyu@bj.sgcc.com.cn (T.R.); dongjiahan@bj.sgcc.com.cn (J.D.)
* Correspondence: wyxu@zju.edu.cn

Abstract: As an important component of the smart grid, vehicle-to-grid (V2G) networks can deliver diverse auxiliary services and enhance the overall resilience of electrical power systems. However, V2G networks face two main challenges due to a large number of devices that connect to it. First, V2G networks suffer from serious security threats, such as doubtful authenticity and privacy leakage. Second, the efficiency will decrease significantly due to the massive requirements of authentication. To tackle these problems, this paper proposes a cross-domain authentication scheme for V2G networks based on consortium blockchain and certificateless signature technology. Featuring decentralized, open, and transparent transactions that cannot be tampered with, this scheme achieves good performance on both security and efficiency, which proves to be suitable for V2G scenarios in the smart grid.

Keywords: smart grid; vehicle-to-grid; cross-domain authentication; consortium blockchain; certificateless; signature

Citation: Miao, Q.; Ren, T.; Dong, J.; Chen, Y.; Xu, W. A 3C Authentication: A Cross-Domain, Certificateless, and Consortium-Blockchain-Based Authentication Method for Vehicle-to-Grid Networks in a Smart Grid. *Symmetry* **2024**, *16*, 336. https://doi.org/10.3390/sym16030336

Academic Editors: Yu Jiang, Jinbo Xiong, Shang Gao, Yuexiu Xing and Sergei D. Odintsov

Received: 27 January 2024
Revised: 27 February 2024
Accepted: 8 March 2024
Published: 11 March 2024

1. Introduction

As the new generation of power grids, smart grids serve as a bridge for exchanging information and allocating resources between customers and the grid, which can facilitate the mutual flow of energy between them. To further address the temporal and spatial imbalances of electricity in the power grid, as well as the issue of energy storage, smart grids have introduced the technology of vehicle-to-grid (V2G) [1]. This technology enables electric vehicles to supply electricity to the grid during peak hours, and to store energy from the grid during off-peak hours, which can achieve bidirectional energy flow between vehicles and the grid as well as mitigate the issue of power imbalance [2].

Despite the numerous benefits brought by V2G technology, the information is transmitted through public channels when bidirectional communication occurs between electric vehicles and the grid. However, this open network is susceptible to various cyber attacks, potentially allowing adversaries to fully capture user information [3], including not only the vehicle information like battery status and vehicle locations, but also the private information like home addresses and driving habits. After the privacy leakage, malicious attackers may sell user identity information for profit, or masquerade as legitimate users to freely access charging, discharging, or other services in V2G networks. In addition to the security issues, communication efficiency is also a concern in V2G networks. Users may, for various reasons such as business trips or tourism, drive electric vehicles to different regions. Since the vehicle's registration information is stored on local servers during enrollment, cross-domain authentication is inevitable when the vehicle requests charging or discharging services in a new area. The assistance of entities in the V2G network is

needed for message transmission between different regions, thereby increasing the communication burden. Therefore, due to the high mobility of vehicles in the V2G network, the massive demands for cross-domain authentication are likely to significantly reduce the communication efficiency.

It is obvious that security and efficiency are the two major focal points in the design of authentication protocols for V2G [4–7], and the privacy-preserving authentication schemes have been extensively researched. Raya and Hubaux [8] utilized anonymous certificates to conceal the real identity of users, and suggested that each vehicle node stores anonymous certificates to use different public–private key pairs in each authentication process. Similarly, Sun et al. [9] proposed an efficient vehicle communication anonymous authentication scheme based on certificates. Abdallah and Shen [10] proposed an authentication scheme based on bilinear mapping technology, which achieved mutual authentication in V2G networks, ensuring that the communicating peer entities have legitimate identities and can effectively resist spoofing attacks. Shen et al. [11] proposed a key protocol that enabled mutual authentication without revealing the user's real identity. Eiza et al. [12] studied the security and privacy issues in V2G networks using mobile IP communication, and proposed a mobile agent IPv6 protocol by employing blind signatures based on the RSA algorithm and incorporating built-in tag technology, which could also ensure the traceability for vehicles. Roman et al. [13] proposed a pairing-based authentication protocol to ensure the confidentiality of communication, protect the identity of vehicle users, and prevent vehicles from being tracked by malicious attackers. Park et al. [14] proposed a dynamic privacy-preserving and lightweight key negotiation protocol for V2G in SIoT, which was capable of resisting attacks such as impersonation, offline password guessing, man-in-the-middle, replay, and tracking. Su et al. [15] took the issues of an untrusted third-party in V2G networks into consideration, and proposed a lightweight authentication protocol using non-singular elliptic curves. Simultaneously, a secure two-party protocol was employed for the negotiation of the system master key between third-party entities and the dispatch center, preventing internal attacks. Secchi et al. [16] proposed a quadratic optimization algorithm to mitigate fluctuations in power supply and demand caused by increasing electric vehicles and photovoltaic penetration. Reddy et al. [17] suggested a lightweight protocol for key agreement and mutual authentication between entities operating in a V2G environment.

While the above schemes implement authentication between vehicles and grid servers in different ways, many of them use algorithms based on public-key cryptography, leading to higher communication and computation costs. In addition, the design of vehicle registration or authentication processes based on Public Key Infrastructure (PKI) is centralized. With the continuous increase in authenticated vehicles, this will not only easily result in issues such as increased load on authentication servers, prolonged authentication delays, and difficulties in certificate management and storage, but also pose problems of single points of failure and excessive power of trusted third-parties. To alleviate the aforementioned issues, certificateless authentication schemes in V2G are gradually coming into focus. In addition, blockchain, as a distributed database with excellent features including decentralization, immutability, and anonymity, is gradually being integrated into authentication frameworks in V2G. Aitzhan and Svetinovic [18] implemented secure transaction verification in the energy trading process based on blockchain technology and multi-signature mechanisms. Guan et al. [19] proposed an efficient data aggregation scheme based on blockchain for the privacy protection of user electricity consumption in smart grids. Garg et al. [20] proposed a blockchain-based hierarchical authentication mechanism, ensuring mutual authentication between vehicles, charging stations, and servers based on elliptic curve encryption algorithms. Wang et al. [21] proposed an efficient anonymous rewarding scheme based on blockchain, implementing security requirements in V2G networks through ring signatures and encryption algorithms. Ali et al. [22] proposed a certificateless public key signature scheme based on blockchain, but it involved a significant number of bilinear pairing operations for signature verification, leading to a decrease in system

performance. Patil et al. [23] proposed an authentication protocol based on blockchain technology and physical unclonable function technology, which utilized smart contracts in the blockchain to resist data tampering attacks.

Although the blockchain-based solutions mentioned above can achieve identity authentication between different entities in V2G networks and resist common cyber attacks, there are still issues such as low communication efficiency and privacy leakage during entity authentication. In other words, it is challenging to simultaneously balance high security and efficiency. Additionally, existing solutions address the issue of cross-domain authentication between vehicles and servers in multiple regions in practical applications to a lesser extent, and lack a systematic analysis of cross-domain authentication.

Therefore, we propose a cross-domain authentication scheme for V2G networks in a smart grid system based on the consortium blockchain, UTXO mechanism, and certificateless signature technology that addresses common weaknesses of most existing authentication schemes. In addition, as an important security process in which both the client and the server verify each other's identities before establishing a connection, mutual authentication ensures that both parties involved are legitimate, reducing the risk of unauthorized access or data breaches. In a typical scenario, the client presents their credentials, and the server validates them. Then, these two entities exchange and repeat the symmetric procedures mentioned above. If both checks pass, the connection is established. In our proposed scheme, we achieve the mutual authentication between the electric vehicle (client) and the charging station (server) by implementing some kinds of cryptographic mechanisms like the message authentication code, negotiated session key, and hash value comparison. Through this method, we can verify the identity legitimacy of these two semi-trusted entities to maintain data security and message integrity in a symmetric manner. The main contributions of this paper can be summarized as follows:

- This paper formalizes the system model of V2G networks in a smart grid and elaborates the detailed process of cross-domain authentication in a systematic manner, which includes two scenarios of individual verification and aggregated verification.
- The proposed scheme can achieve message integrity, user anonymity, and unlinkability, as well as traceability through a theoretical analysis. In addition, our method is also capable of resisting common attacks including a replay attack, tampering attack, and impersonation attack, which can protect user privacy in an effective way.
- The proposed scheme precludes complex cryptographic operations like bilinear pairings and map-to-point hash operations, and reduces redundant computational overhead through aggregated signature verification, therefore achieving good performance on computation efficiency through evaluation.

The rest of the paper is organized according to the outline given as follows: Section 2 reviews the preliminaries related to the proposed scheme. Section 3 introduces the system model and security requirements in V2G networks. Section 4 gives the detailed steps of the proposed scheme. Section 5 provides a theoretical analysis of the scheme in terms of security. Section 6 presents the evaluation performance of the scheme on computation efficiency. Section 7 discusses the real-world application and the potential privacy threats. Finally, in Section 8, we make a conclusion about the proposed scheme.

2. Preliminaries

This section mainly introduces the background knowledge related to the proposed cross-domain authentication scheme.

2.1. Mathematical Assumptions

We utilize elliptic curve cryptography (ECC), which was first proposed in [24], to ensure the security of our authentication scheme. Based on ECC, Johnson et al. [25] later formalized the Elliptic Curve Digital Signature Algorithm (ECDSA). Based on an equivalent level of security, this algorithm can achieve higher security with shorter key lengths. Therefore, it has found widespread application in the field of cryptography. There

are several intractable problems in ECC, which are suitable for cryptographic purposes as there is no polynomial algorithm to solve them efficiently by brute force within probabilistic polynomial time.

1. *Elliptic Curve Discrete Logarithm (ECDL) Problem* [26]: Define an elliptic curve group G of order q whose generator is P, where q is a large prime number. With the unknown element $a \in Z_q^*$ in the finite field, while given $aP \in G$ and P, it is computationally difficult to solve for a within polynomial time.
2. *Elliptic Curve Computational Diffie–Hellman (ECCDH) Problem* [27]: Define an elliptic curve group G of order q whose generator is P, where q is a large prime number. With two unknown elements $a, b \in Z_q^*$ in the finite field, while given $aP \in G$ and $bP \in G$, it is computationally difficult to solve for $abP \in G$ within polynomial time.

2.2. Message Authentication Code

The message authentication code (MAC) is a key-related one-way hash function, also known as message checksum. It enables the authentication of the message source and integrity verification. It assumes that there are two communicating parties A and B, with a shared key K. When A sends a message to B, A calculates the MAC using the shared key and the sent message:

$$MAC = C(K, m) \tag{1}$$

where C is the MAC function, K is the shared key, and m is the message to be sent.

A sends both the message m and MAC together to B. Upon receiving them, B performs the same calculation using the shared key K to obtain MAC':

$$MAC' = C(K, m) \tag{2}$$

Then, B compares whether MAC' equals MAC:

$$MAC' \stackrel{?}{=} MAC \tag{3}$$

If Equation (3) does not hold, it indicates that the information has been corrupted or tampered during transmission. Otherwise, the received message contains integrity, and B trusts that the message originates from A.

2.3. Consortium Blockchain

Consortium blockchain represents a collaborative and distributed approach to blockchain technology, where multiple organizations come together to form a network with shared control over the blockchain. Unlike public blockchains that are open to anyone, or private blockchains that are restricted to a single entity, consortium blockchain strikes a balance by allowing a pre-selected group of trusted participants to validate transactions and maintain the distributed ledger [28]. This collaborative model offers several advantages. Firstly, it ensures a more efficient and scalable system as compared to public blockchains, where consensus mechanisms are often resource-intensive. Secondly, consortium blockchain enhances data security by limiting access to authorized entities, thereby reducing the risk of unauthorized tampering. Additionally, it addresses privacy concerns by providing a controlled environment where sensitive information can be shared securely among consortium members, making it a compelling choice for applications requiring a balance between transparency and confidentiality. Therefore, it is suitable under the interactive integration scenario of new energy vehicles and a power grid, so as to improve the security, information transparency, and authentication efficiency in distributed energy transactions [29,30].

2.4. UTXO Model

The Unspent Transaction Output (UTXO) model is a fundamental concept in blockchain technology, defining the manner in which transactions are tracked and verified within

a cryptocurrency network. In the UTXO model, each transaction generates a set of outputs, each representing an amount of cryptocurrency. These outputs, or UTXOs, serve as the inputs for subsequent transactions, creating a chain of ownership across the blockchain [31,32], as shown in Figure 1.

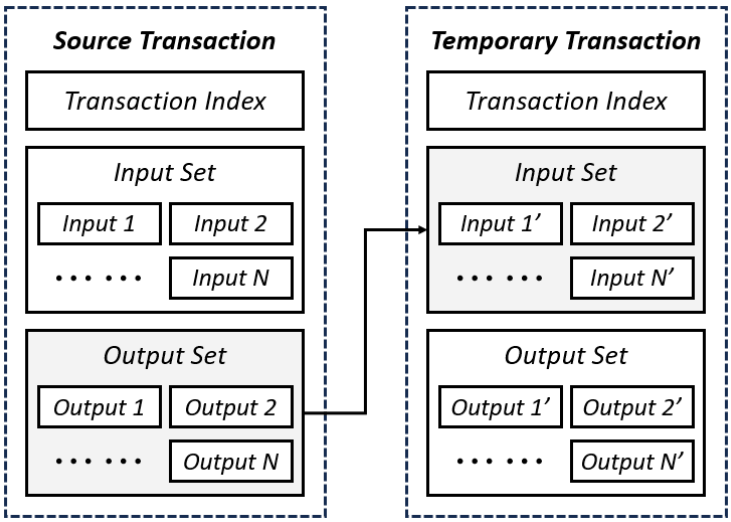

Figure 1. The UTXO model creates new inputs of temporary transaction based on the previous outputs of source transaction.

In detail, the *Output Set* mainly involves two parameters, V and $Hash_{pk}$, where V is the transaction value of the transaction object, usually represented in the form of cryptocurrency, and $Hash_{pk}$ represents the hash value of the user's public key. The *Input Set* of the current/temporary transaction is generated based on the *Output Set* of the previous/source transaction, which mainly involves four parameters, $Tran_{index}$, N_{out}, pk, and σ_{sk}, where $Tran_{index}$ represents the transaction index corresponding to this input, N_{out} is used to mark the position of the output corresponding to this input in the previous transaction, pk represents the user's public key corresponding to $Hash_{pk}$ in the *Output Set* of the previous transaction, and σ_{sk} is the signature generated based on the user's private key.

In a consortium blockchain, the validators can confirm the legitimacy of a transaction by verifying the validity of the inputs in the current transaction. The specific verification details are as follows:

1. Validators use the hash function to compute the hash value $Hash_{in}$ of the public key pk contained in the input:

$$Hash_{in} = HashFunc(pk) \tag{4}$$

   where $HashFunc$ is the hash function and $Hash_{in}$ is the computation result of the hash value.
2. Validators compare whether the computed hash value $Hash_{in}$ equals the hash value $Hash_{pk}$ contained in the source transaction output corresponding to this input, to check for consistency:

$$Compare(Hash_{in}, Hash_{pk}) \stackrel{?}{=} True \tag{5}$$

3. Validators use the public key pk to verify the signature σ_{sk}:

$$Verify(\sigma_{sk}, pk) \stackrel{?}{=} True \tag{6}$$

After the aforementioned process of identity verification, users can engage in the specific transactions by utilizing the temporary transaction, and perform transaction aggregation and update with the assistance of consortium blockchain nodes. This transaction is subsequently stored by the trusted nodes in the consortium blockchain database, and redefined as the user's source transaction. Users can further generate another latest temporary transaction based on the output of the new source transaction.

Unlike the account-based model, where balances are associated with user accounts, UTXO tracks the specific units of cryptocurrency that have not been spent. This model enhances security and facilitates efficient transaction verification. When a user initiates a new transaction, they reference existing UTXOs as inputs and generate new UTXOs as outputs, ensuring a transparent and traceable record of ownership. The UTXO model contributes to the overall security, scalability, and privacy of blockchain networks, making it a crucial element in the design and functioning of various decentralized systems [33].

3. System Overview

This section first briefly introduces the architecture of the V2G network model and roles of each entity contained in this system, then further describes the security requirements, which are designed to meet in the proposed scheme.

3.1. System Model

The overall architecture of the V2G system model is shown in Figure 2. There are two trust domains, A and B, in this V2G network. Each domain has a grid server, GS, and several charging stations, CSs. In each trust domain, there are many electric vehicles, EVs, from the same domain interacting with the local charging stations, as circled in blue. However, if a vehicle from domain A has the charging or discharging requirements in domain B, as circled in red, then a process of cross-domain authentication is needed. In addition to the entities mentioned above, the system also includes a Key Generation Center (KGC) and a consortium blockchain. The following will introduce their roles, respectively:

1. *Electric Vehicle (EV)*: An EV is equipped with an On-Board Unit (OBU) and a battery. The OBU is responsible for handling the perception, computation, and communication tasks of the vehicle terminal and is capable of independently generating keys. Additionally, the OBU has tamper-resistant features, suitable for storing the vehicle's private information, such as registration information and keys. Assuming that EV_A from trust domain A has a charging or discharging requirement with CS_B in trust domain B, it must undergo authentication with GS_B before proceeding. After that, EV_A can facilitate bidirectional energy flow between the on-board battery and the power grid. Finally, EV_A engages in settlement transactions with the transaction center.
2. *Charging Station (CS)*: In this scheme, it is assumed that CS_B in trust domain B is distributed across specific areas as multiple roadside units (RSUs), and they are managed by the dispatch center in GS_B. Each CS_B can directly connect to the power grid and engage in energy exchange with EV_A. In this scheme, CS_B serves as an information transfer medium between EV_A and GS_B, primarily responsible for aggregating signatures from multiple EV_A, and simultaneously forwarding information from EV_A or GS_B. In this paper, it is assumed that CS_B is a semi-trusted entity.
3. *Grid Server (GS)*: A GS plays an important role in the V2G system, which mainly consists of an authentication server, a dispatch center, and a transaction settlement center. In this scheme, GS_A is responsible for the registration of EV_A and stores the mapping between the real identity information and the pseudonym of EV_A. GS_B has the capability to generate keys and its authentication server is responsible for the authentication of aggregated signatures from multiple EV_A. The dispatch center manages CS_B and controls the flow of electrical energy in the power grid. After the charging or discharging process of EV_A is completed, the transaction settlement center

is responsible for order settlement and management of transaction information. In this paper, it is assumed that GS_A and GS_B are fully trusted entities.

4. *Key Generation Center (KGC)*: A *KGC* is responsible for generating and publishing system public parameters. Additionally, a *KGC* possesses the system public key, which is used to generate encrypted pseudonyms during the registration phase of EV_A. Furthermore, a *KGC* is also responsible for handling an authentication error reported by the *GS*. In this paper, it is assumed that *KGC* is a fully trusted entity.
5. *Consortium Blockchain*: The member nodes of the consortium blockchain include GS_A and GS_B, both of which have the authority to view and update the contents of blocks in the chain. In this scheme, the consortium blockchain stores transaction information lists and revocation lists of vehicles. The information in these two lists is grouped according to the domain identifier of the vehicles. When the *GS* looks up vehicle information, it first uses the domain identifier as an index to locate the group of the vehicle. Then, it performs a fine-grained search within the group based on the vehicle's identity information and public key. The registration information of EV_A is uploaded to the transaction information list by GS_A. In case of malicious behavior by EV_A, GS_B can notify GS_A to perform identity tracing and revocation procedures for EV_A.

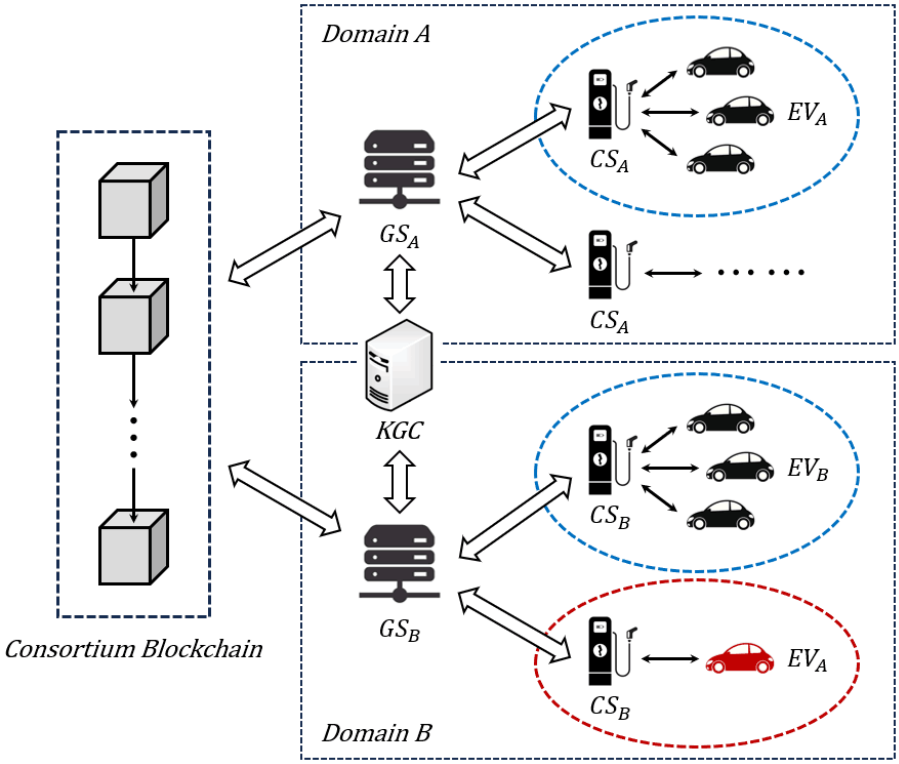

Figure 2. The overall architecture of the V2G system model, where the blue circle indicates intra-domain authentication, while the red circle indicates cross-domain authentication.

3.2. Security Requirements

Rajasekaran et al. [34] conducted a thorough categorization of various potential security threats in V2G networks. Based on their work, our proposed scheme is designed to achieve the following security requirements:

1. *Message integrity*: When any information issued by entities in the system is intentionally tampered with during transmission, the receiver can discover and reject the message.

2. *Anonymity*: Vehicles participate in the authentication process without revealing their real identities, meaning that the real identities of the vehicles are kept confidential from any entity other than the registration grid server.
3. *Unlinkability*: Adversaries cannot link multiple messages sent by the same entity, meaning that adversaries cannot deduce the real identity of entities from the obtained information.
4. *Traceability*: When a vehicle engages in malicious behavior during the authentication process, the system has the ability to trace and disclose the ownership of this vehicle.
5. *Resistance to attacks*: The proposed scheme should be able to resist common attacks, such as the replay attack, the tampering attack, and the impersonation attack.

4. Proposed Authentication Scheme

This section mainly describes the several processes of the proposed scheme, which consists of six parts: the initialization phase, registration phase, new transaction generation, transaction authentication, transaction phase, and revocation phase. In this section, we assume that the vehicle EV_A registered in trust domain A applies for charging or discharging operations in trust domain B. We take this scenario as an example to illustrate the proposed cross-domain authentication scheme for V2G networks in detail.

The symbols and their meanings involved in this scheme are explained in Table 1.

Table 1. The definitions of the relevant symbols.

Notation	Description
KGC	Key Generation Center
EV_A	Electric vehicle from trust domain A
CS_B	Charging station in trust domain B
GS_A, GS_B	Grid server of trust domain A or B
E	The elliptic curve: $y^2 = x^3 + Ax + B \ mod \ p$
G	The additive cyclic group
P	The generator of G
p, q	The two large prime numbers
H_i	The i-th hash function
Z_q^*	q-Order integer multiplication cyclic group
$\{SK_x, PK_x\}$	The private key and public key of entity x [1]
ID_{pseudo}	The encrypted pseudonym of EV_A
ID_{EV_A}	The identity information of EV_A
F_A	The domain identifier of EV_A
$K_{CS_B-GS_B}$	The session key between CS_B and GS_B
t	The timestamp
M_y	The message from entity y [2]
σ_{EV_A}	The signature information of EV_A
$Tran_{EV_A}$	The transaction information of EV_A
MAC	Message authentication code
$\oplus$	XOR operator
$\|\|$	Concatenation operator

[1] Entity x can be KGC, GS_A, GS_B, or EV_A. [2] Entity y can be EV_A or CS_B.

4.1. Initialization Phase

4.1.1. KGC Initialization

The KGC first initializes the system parameters. Define an elliptic curve over a finite field— $E : y^2 = x^3 + Ax + B \ mod \ p$, where $A, B \in Z_p$ satisfy $4A^3 + 27B^2 \neq 0$. The points on E and the points at infinity form a cyclic group, G, of order q, whose generator is denoted as P. p and q are two large prime numbers. KGC randomly selects three secure hash functions—$H_1 : G \rightarrow Z_q^*$, $H_2 : \{0,1\}^* \rightarrow Z_q^*$, $H_3 : \{0,1\}^* \rightarrow Z_q^*$.

4.1.2. System Key Generation

1. *KGC key generation*: The *KGC* selects a random number, $SK_{KGC} \in Z_q^*$, as the system private key, and calculates $PK_{KGC} = SK_{KGC} \cdot P$ as the system public key.
2. *GS key generation*: GS_A in trust domain A selects a random number, $SK_{GS_A} \in Z_q^*$, as the system private key, and calculates $PK_{GS_A} = SK_{GS_A} \cdot P$ as the system public key. Similarly, GS_B in trust domain B selects a random number, $SK_{GS_B} \in Z_q^*$, as the system private key, and calculates $PK_{GS_B} = SK_{GS_B} \cdot P$ as the system public key.
3. *System parameter publication*: All system parameters $\{q, G, P, H_1, H_2, H_3, PK_{KGC}, PK_{GS_A}, PK_{GS_B}\}$ are public and uploaded to consortium blockchain.

4.2. Registration Phase

4.2.1. Vehicle Pseudonym Registration

1. *Pseudo-identity generation*: The real identity of EV_A, such as the vehicle unique identity or device identification code, is denoted as $ID_{real} \in Z_q^*$. EV_A selects a random number, $r \in Z_q^*$, generates pseudo-identity $ID'_{pseudo} = r \cdot P$, and transmits $\{ID_{real}, ID'_{pseudo}\}$ to GS_A via a secure channel.
2. *Pseudo-identity encryption*: GS_A verifies the legitimacy of ID_{real}, checks the uniqueness of ID'_{pseudo} in the local database, and calculates encrypted pseudonym $ID_{pseudo} = ID_{real} \oplus H_1(SK_{GS_A}, PK_{KGC}, F_A)$, where SK_{GS_A} is the private key of GS_A, PK_{KGC} is the public key of *KGC*, and F_A is the domain identifier of EV_A.
3. *Identity information storage*: GS_A stores the registration information of EV_A $\{ID_{real}, ID'_{pseudo}, ID_{pseudo}, F_A\}$ in the local database, and transmits identity information $ID_{EV_A} = \{ID'_{pseudo}, ID_{pseudo}, F_A\}$ to EV_A via a secure channel. EV_A stores the identity information ID_{EV_A} in its tamper-resistant OBU.

4.2.2. Transaction Initialization

GS_A issues authentication tokens to EV_A through a registration transaction as the source transaction, to initialize the user's authentication permissions. The initialized source transaction is uploaded to the consortium blockchain database by GS_A.

4.2.3. Vehicle Key Generation

EV_A selects a random number, $SK_{EV_A} \in Z_q^*$, as the private key, and calculates $PK_{EV_A} = SK_{EV_A} \cdot P$ as the public key. EV_A then stores $\{SK_{EV_A}, PK_{EV_A}\}$ in its tamper-resistant OBU.

4.2.4. Session Key Negotiation

CS_B sends its identifier ID_{CS_B} to GS_B. After registration verification, GS_B selects a secure key, $K_{CS_B-GS_B}$, as the session key between CS_B and GS_B, stores $\{ID_{CS_B}, K_{CS_B-GS_B}\}$ in the local database, and transmits $K_{CS_B-GS_B}$ to CS_B via a secure channel. Upon receiving the session key, CS_B stores it in its secure storage device.

4.3. New Transaction Generation

4.3.1. Transaction Index Construction

When EV_A initiates a charging or discharging request to CS_B in trust domain B, EV_A calculates the transaction index $Tran_{index} = H_2(ID_{pseudo}, PK_{GS_B}, t)$ through encryption, where ID_{pseudo} is the encrypted pseudonym of EV_A, PK_{GS_B} is the public key of GS_B, and t is the timestamp.

4.3.2. New Input Construction

Based on the $Output$ information in the source transaction, EV_A generates the $Input'$ information for the new transaction.

4.4. Individual Transaction Authentication

Firstly, we discuss the individual transaction authentication scenario, where we assume that there is only single vehicle EV_A from trust domain A having the charging or discharging request in trust domain B. Later, in Section 4.5, we will discuss the scenario with multiple vehicles.

4.4.1. Vehicle Signature

1. *Vehicle signature generation*: EV_A selects a random number, $d \in Z_q^*$, calculates $R = d \cdot P$ and $H_{EV_A} = H_3(M_{EV_A}, ID_{EV_A}, PK_{EV_A}, R, t)$, and then calculates the partial signature information $S_{EV_A} = (H_{EV_A} \cdot d + SK_{EV_A})\ mod\ p$, from which EV_A can receive its complete signature information $\sigma_{EV_A} = \{R, S_{EV_A}\}$, where M_{EV_A} is the message of a charging or discharging request to be signed, $ID_{EV_A} = \{ID'_{pseudo}, ID_{pseudo}, F_A\}$ is the identity information of EV_A, PK_{EV_A} is the public key of EV_A, SK_{EV_A} is the private key of EV_A, and t is the timestamp.
2. *Vehicle signature transmission*: EV_A sends request message set $MSet_{EV_A} = \{ID_{EV_A}, M_{EV_A}, Tran_{EV_A}, PK_{EV_A}, t\}$ to CS_B, where the complete signature information σ_{EV_A} is included in $Input'$ of the transaction information $Tran_{EV_A}$. CS_B first checks whether the timestamp t meets the real-time requirement. If not, the message is discarded. Then, CS_B generates request message $M_{CS_B} = Encrypt(K_{CS_B-GS_B}, MSet_{EV_A})$ and further calculates $T_{CS_B} = MAC_{K_{CS_B-GS_B}}(M_{CS_B}||n)$, where $K_{CS_B-GS_B}$ is the negotiated session key between CS_B and GS_B, $Encrypt$ is the agreed-upon encryption algorithm, MAC is the agreed-upon message authentication code verification mechanism, and $n \in Z_q^*$ is the random number selected by CS_B. Finally, CS_B transmits $\{M_{CS_B}, T_{CS_B}, n, t\}$ to GS_B.

4.4.2. Signature Verification

1. *Identity authenticity and message integrity verification*: GS_B first checks whether the timestamp t meets the real-time requirement. If not, the message is discarded. Then, GS_B calculates $T'_{CS_B} = MAC_{K_{CS_B-GS_B}}(M_{CS_B}||n)$ and compares whether T'_{CS_B} equals T_{CS_B}. If not, the identity authentication of CS_B has not passed, and GS_B will report the error to the KGC.
2. *Signature validity verification:* GS_B *receives plaintext* $MSet_{EV_A} = Decrypt(K_{CS_B-GS_B}, M_{CS_B})$ through decryption, and then calculates $H'_{EV_A} = H_3(M_{EV_A}, ID_{EV_A}, PK_{EV_A}, R, t)$ based on the obtained information $\{ID_{EV_A}, M_{EV_A}, \sigma_{EV_A}, PK_{EV_A}, t\}$. If the equation $S_{EV_A} \cdot P = H'_{EV_A} \cdot R + PK_{EV_A}$ is satisfied, the signature validity verification is successful.

4.4.3. Source Transaction Retrieval

Based on the transaction information $Tran_{EV_A}$, GS_B retrieves the source transaction information of EV_A by querying the consortium blockchain database.

4.4.4. Hash Value Comparison

GS_B calculates the hash value of the vehicle public key PK_{EV_A} included in the $Input'$ of the temporary transaction $Hash_{in} = H_1(PK_{EV_A})$ through the hash function H_1, and compares this hash value with the hash value $Hash_{out}$ in the $Output$ of the source transaction, which corresponds to this temporary transaction. If these two hash values are equal, the hash verification is successful.

If the above conditions are met, the transaction authentication is successful, and GS_B finally accepts the charging or discharging request message. The whole pipeline of transaction authentication is shown in Figure 3.

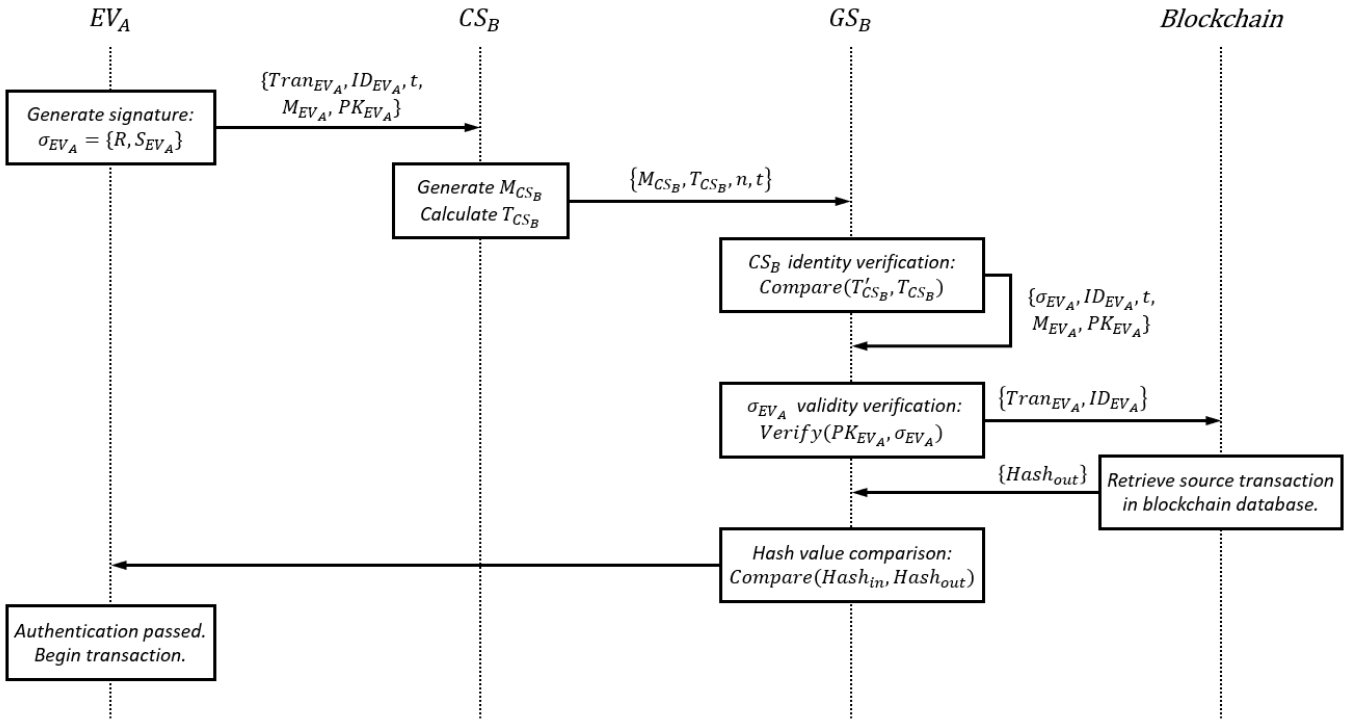

Figure 3. The pipeline of transaction authentication.

4.5. Aggregated Transaction Authentication

In Section 4.4, we have discussed the scenario of individual transaction authentication. In addition, if there are multiple vehicles from trust domain A $\{EV_{A-1}, EV_{A-2}, \ldots, EV_{A-n}\}$ having the charging or discharging request in trust domain B, we can leverage the method of an aggregated signature and aggregated verification to enhance the efficiency of transaction authentication. It should be noted that Sections 4.4 and 4.5 are just two parallel cases. In the actual cross-domain authentication pipeline, only one of them needs to be adopted based on the specific situation.

Assume the identity information of n vehicles $\{EV_{A-1}, EV_{A-2}, \ldots, EV_{A-n}\}$ is $\{ID_{EV_{A-1}}, ID_{EV_{A-2}}, \ldots, ID_{EV_{A-n}}\}$, with their private keys and public keys being, respectively, $\{SK_{EV_{A-1}}, SK_{EV_{A-2}}, \ldots, SK_{EV_{A-n}}\}$ and $\{PK_{EV_{A-1}}, PK_{EV_{A-2}}, \ldots, PK_{EV_{A-n}}\}$. The messages to be signed are $\{M_{EV_{A-1}}, M_{EV_{A-2}}, \ldots, M_{EV_{A-n}}\}$. Below, we will take the example of the i-th vehicle, EV_{A-i}, to illustrate the process of aggregated transaction authentication involving multiple vehicles, where $i \in \{1, 2, \ldots, n\}$.

4.5.1. Vehicle Aggregated Signature

1. *Vehicle signature generation*: EV_{A-i} selects a random number, $d_i \in Z_q^*$, calculates $R_i = d_i \cdot P$ and $H_{EV_{A-i}} = H_3(M_{EV_{A-i}}, ID_{EV_{A-i}}, PK_{EV_{A-i}}, R_i, t_i)$, and then calculates the partial signature information $S_{EV_{A-i}} = (H_{EV_{A-i}} \cdot d_i + SK_{EV_{A-i}})\ mod\ p$, from which EV_{A-i} can receive its complete signature information $\sigma_{EV_{A-i}} = \{R_i, S_{EV_{A-i}}\}$. Similarly, the remaining vehicles generate their signatures $\{\sigma_{EV_{A-1}}, \sigma_{EV_{A-2}}, \ldots, \sigma_{EV_{A-n}}\}$, respectively, in the same manner.
2. *Vehicle signature aggregation and transmission*: EV_{A-i} sends request message set $MSet_{EV_{A-i}} = \{ID_{EV_{A-i}}, M_{EV_{A-i}}, Tran_{EV_{A-i}}, PK_{EV_{A-i}}, t_i\}$ to CS_B, where the complete signature information $\sigma_{EV_{A-i}}$ is included in $Input'$ of the transaction information $Tran_{EV_{A-i}}$. Similarly, the remaining vehicles send their request message sets $\{MSet_{EV_{A-1}}, MSet_{EV_{A-2}}, \ldots, MSet_{EV_{A-n}}\}$ to CS_B, respectively, in the same manner. CS_B first checks whether the timestamp t_i in each request message set meets the real-time requirement. If not, the message is discarded. Then, CS_B aggregates the signature information from each vehicle $\{\sigma_{EV_{A-1}}, \sigma_{EV_{A-2}}, \ldots, \sigma_{EV_{A-n}}\}$ and receives

$\sigma = \{R, S\}$, where $R = \{R_1, R_2, \ldots, R_n\}$ and $S = \sum_{i=1}^{n} S_{EV_{A-i}}$. Later, CS_B gathers request message sets and aggregated partial signature information S, and receives $MSet = \{MSet_{EV_{A-1}}, MSet_{EV_{A-2}}, \ldots, MSet_{EV_{A-n}}, S\}$. After that, CS_B generates request message $M_{CS_B} = Encrypt(K_{CS_B-GS_B}, MSet)$ and further calculates $T_{CS_B} = MAC_{K_{CS_B-GS_B}}(M_{CS_B}||n)$, where $K_{CS_B-GS_B}$ is the negotiated session key between CS_B and GS_B, $Encrypt$ is the agreed-upon encryption algorithm, MAC is the agreed-upon message authentication code verification mechanism, and $n \in Z_q^*$ is the random number selected by CS_B. Finally, CS_B transmits $\{M_{CS_B}, T_{CS_B}, n, t\}$ to GS_B.

4.5.2. Aggregated Signature Verification

1. *Identity authenticity and message integrity verification*: GS_B first checks whether the timestamp t meets the real-time requirement. If not, the message is discarded. Then, GS_B calculates $T'_{CS_B} = MAC_{K_{CS_B-GS_B}}(M_{CS_B}||n)$ and compares whether T'_{CS_B} equals T_{CS_B}. If not, the identity authentication of CS_B has not passed, and GS_B will report the error to KGC.
2. *Aggregated signature validity verification: GS_B receives plaintext $MSet = Decrypt(K_{CS_B-GS_B}$, $M_{CS_B})$* through decryption, and then calculates $H'_{EV_{A-i}} = H_3(M_{EV_{A-i}}, ID_{EV_{A-i}}$, $PK_{EV_{A-i}}, R_i, t_i)$ based on the obtained information of each vehicle $\{ID_{EV_{A-i}}, M_{EV_{A-i}}$, $\sigma_{EV_{A-i}}, PK_{EV_{A-i}}, t_i\}$. If the equation $S \cdot P = \sum_{i=1}^{n} H'_{EV_{A-i}} \cdot R_i + \sum_{i=1}^{n} PK_{EV_{A-i}}$ is satisfied, the aggregated signature validity verification is successful, where S is the aggregated partial signature information.

4.5.3. Source Transaction Retrieval

Based on the transaction information $Tran_{EV_{A-i}}$ of each vehicle, GS_B retrieves the source transaction information of EV_{A-i} by querying the consortium blockchain database.

4.5.4. Hash Value Comparison

GS_B calculates the hash value of each vehicle public key $PK_{EV_{A-i}}$ included in the $Input'$ of the temporary transaction $Hash_{in} = H_1(PK_{EV_{A-i}})$ through the hash function H_1, and compares this hash value with the hash value $Hash_{out}$ in the $Output$ of the source transaction, which corresponds to this temporary transaction. If these two hash values are equal, the hash verification is successful.

If the above conditions are met, the transaction authentication of EV_{A-i} is successful, and GS_B finally accepts the charging or discharging request message.

4.6. Transaction Phase

4.6.1. Temporary Transaction Generation

1. *Authentication token issuance*: GS_B generates a temporary transaction locally, issuing authentication tokens with a quantity of V to EV_A, which is successfully authenticated. Correspondingly, if the user EV_A fails authentication or transmits malicious request messages, a certain quantity of authentication tokens is deducted according to the severity of its threat. Subsequently, GS_B constructs the output of this temporary transaction with the parameters H_{PK} and V, where H_{PK} is the hash value of the public key PK_{EV_A} of EV_A.
2. *Transaction information transmission and service provision*: GS_B sends the temporary transaction information to the charging station CS_B within the region and uploads this temporary transaction information to the consortium blockchain database. Then, CS_B broadcasts the temporary transaction within the management area and provides charging or discharging services to EV_A.

4.6.2. Transaction Aggregation

The user EV_A starts a transaction aggregation, and constructs the $Input$ of this aggregated transaction based on other earlier transactions containing authentication tokens. Then, EV_A constructs the $Output$ of this aggregated transaction based on the local key pair

$\{SK_{EV_A}, PK_{EV_A}\}$, which is stored in its tamper-resistant OBU. Subsequently, EV_A sends the aggregated transaction to GS_B for transaction verification.

4.6.3. Transaction Update

After the temporary transaction generated by user EV_A is aggregated and passes verification by the grid server, this aggregated transaction will be defined as the latest source transaction. Then, GS_B will accordingly update the transaction information $Tran_{EV_A}$ in the consortium blockchain database. After that, based on the updated source transaction, user EV_A can further generate a new temporary transaction for the next cross-domain authentication process and begin a new transaction process of authentication token aggregation.

4.7. Revocation Phase

In special circumstances, such as when the authentication fails or EV_A engages in malicious behavior, it is necessary to reveal the real identity of the vehicle from the pseudonym and then perform vehicle revocation. In the consortium blockchain nodes, the revocation list of vehicles is stored in the form of tuples, such as $<F_A, (ID_{EV_A}, PK_{EV_A}, t_{start}, t_{end})>$, where F_A is the domain identifier of EV_A, $ID_{EV_A} = \{ID'_{pseudo}, ID_{pseudo}, F_A\}$ is the identity information of EV_A, PK_{EV_A} is the public key of EV_A, t_{start} is the start time of vehicle revocation, and t_{end} is the end time of vehicle revocation, when the vehicle can resume normal operation. The process of vehicle identity traceback and revocation is as follows:

4.7.1. Real Identity Traceback

GS_B notifies GS_A through the smart contract in the consortium blockchain. Upon receiving the notification, GS_A reveals the real identity of EV_A by calculating $ID'_{real} = ID_{pseudo} \oplus H_1(SK_{GS_A}, PK_{KGC}, F_A)$. Then, GS_A compares whether ID'_{real} equals the real identity ID_{real} stored in the local database. If not, GS_A will report the error to KGC. Otherwise, GS_A starts vehicle revocation.

4.7.2. Vehicle Revocation

GS_A designates the current time as the start time of vehicle revocation, noted as $t_{start} = t_{current}$. Then, GS_A defines the duration of vehicle revocation $t_{duration}$ based on the severity level of the malicious behavior. Therefore, the end time of vehicle revocation is $t_{end} = t_{start} + t_{duration}$. Finally, GS_A uploads the information $<F_A, (ID_{EV_A}, PK_{EV_A}, t_{start}, t_{end})>$ to the revocation list in consortium blockchain.

5. Security Analysis

This section provides a theoretical analysis about the security and privacy-preserving features that the proposed scheme meets, which specifically includes message integrity, anonymity, unlinkability, traceability, and resistance to common attacks, as mentioned earlier in Section 3.2.

5.1. Message Integrity

In the proposed scheme, there are mainly two processes of message integrity verification. We will discuss them in detail, respectively, in the following parts:

1. *Message integrity from CS_B*: In the system model, we assume that the charging stations are semi-trusted entities. Therefore, it is necessary to verify the identity legitimacy of CS_B before GS_B receives messages from it. After CS_B aggregates messages from EV_A and generates request message M_{CS_B}, CS_B further calculates $T_{CS_B} = MAC_{K_{CS_B-GS_B}}(M_{CS_B}||n)$ used as an intermediate variable for identity verification, and then sends both M_{CS_B} and T_{CS_B} to GS_B. Even if the malicious adversaries manipulate the messages sent by CS_B, they cannot receive the negotiated session key $K_{CS_B-GS_B}$, which is stored in the secure storage device of CS_B. In this way, GS_B can verify the identity legitimacy of CS_B as well as the message integrity through comparing T_{CS_B} and T'_{CS_B}.

2. *Message integrity from EV_A: After GS_B obtains the information* $\{ID_{EV_{A-i}}, M_{EV_{A-i}}, \sigma_{EV_{A-i}}, PK_{EV_{A-i}}, t_i\}$ from CS_B, GS_B performs signature verification by checking whether the equation $S_{EV_{A-i}} \cdot P = H'_{EV_{A-i}} \cdot R_i + PK_{EV_{A-i}}$ is satisfied. After that, GS_B accomplishes the transaction authentication process combined with the hash value comparison. If the messages from EV_A are tampered with or lost during the communication process, the signature verification fails, and GS_B can discard the messages.

5.2. Anonymity

In this scheme, the real identity ID_{real} of EV_A is stored in the local database of GS_A. During the communication process, EV_A utilizes the encrypted pseudonym ID_{pseudo}. The only way for the adversary to receive the real identity of EV_A is to calculate $ID_{real} = ID_{pseudo} \oplus H_1(SK_{GS_A}, PK_{KGC}, F_A)$. However, the adversary cannot receive the private key SK_{GS_A} of GS_A, for the reason that calculating SK_{GS_A} through $PK_{GS_A} = SK_{GS_A} \cdot P$ involves solving the discrete logarithm problem. In addition, it is also difficult for the adversary to receive the encrypted pseudonym ID_{pseudo} of EV_A through the transaction index. Due to the high security level of member nodes GS_A and GS_B, the adversary is not able to receive the authority to view the contents of the consortium blockchain database. Therefore, the anonymity of user identity is protected.

5.3. Unlinkability

During the communication process, the signature information generated by EV_A is $\sigma_{EV_A} = \{R, S_{EV_A}\}$, where $R = d \cdot P$, and $d \in Z_q^*$ is the random number chosen by EV_A. Therefore, multiple messages sent by EV_A actually appear as random to external entities. Even if adversaries obtain several messages sent by EV_A, they cannot trace to the real identity ID_{real} due to the anonymity of EV_A and the randomness of messages. In addition, the transaction index is generated through encryption based on the timestamp t, so that adversaries cannot link multiple different transactions to the same user.

5.4. Traceability

During the vehicle registration phase in this scheme, GS_A conceals the real identity of EV_A by calculating the pseudonym $ID_{pseudo} = ID_{real} \oplus H_1(SK_{GS_A}, PK_{KGC}, F_A)$, and the registration information $\{ID_{real}, ID'_{pseudo}, ID_{pseudo}, F_A\}$ is stored in the local database of GS_A. However, in special circumstances, such as when the authentication fails or EV_A engages in malicious behavior, it is necessary to reveal the real identity of the vehicle from the pseudonym and then perform vehicle revocation. As illustrated in Section 4.7, the process of vehicle identity traceback and revocation is as follows:

1. *Real identity traceback*: GS_B notifies GS_A through the smart contract in the consortium blockchain. Upon receiving the notification, GS_A reveals the real identity of EV_A by calculating $ID'_{real} = ID_{pseudo} \oplus H_1(SK_{GS_A}, PK_{KGC}, F_A)$. Then, GS_A compares whether ID'_{real} equals the real identity ID_{real} stored in the local database. If not, GS_A will report the error to the *KGC*. Otherwise, GS_A starts vehicle revocation.
2. *Vehicle revocation*: GS_A designates the current time as the start time of vehicle revocation, noted as $t_{start} = t_{current}$. Then, GS_A defines the duration of vehicle revocation $t_{duration}$ based on the severity level of the malicious behavior. Therefore, the end time of vehicle revocation is $t_{end} = t_{start} + t_{duration}$. Finally, GS_A uploads the information $<F_A, (ID_{EV_A}, PK_{EV_A}, t_{start}, t_{end})>$ to the revocation list in consortium blockchain.

5.5. Resistance to Attacks

1. *Resist replay attack*: Firstly, messages sent by both EV_A and CS_B contain the timestamp t. The receiver must verify whether t meets the real-time requirements. Once t is deemed invalid, the message is discarded, which can effectively resist replay attacks. Secondly, in the transaction phase, the authenticated transaction will be updated and stored as the latest source transaction in the consortium blockchain database. If the adversary generates a temporary transaction based on a previously invalidated source

transaction, it will not pass authentication. Therefore, the proposed scheme is capable of resisting replay attacks.

2. *Resist tampering attack*: In the signature verification phase, any modification to messages sent by CS_B will be detected after calculating $T'_{CS_B} = MAC_{K_{CS_B-GS_B}}(M_{CS_B}||n)$ and comparing whether T'_{CS_B} equals T_{CS_B}, and any modification to messages sent by EV_A will be detected during the verification of $S_{EV_{A-i}} \cdot P = H'_{EV_{A-i}} \cdot R_i + PK_{EV_{A-i}}$. Therefore, the proposed scheme is capable of resisting tampering attacks.
3. *Resist impersonation attack*: The impersonation or spoofing attack aims to steal authentication credentials to gain unauthorized service access. In this scheme, assuming that the signature information $\sigma_{EV_{A-i}} = \{R_i, S_{EV_{A-i}}\}$ is verifiable, it is impossible for the adversary to obtain the private key $SK_{EV_{A-i}}$ of EV_{A-i} among public parameters based on the assumption of the discrete logarithm difficulty. Therefore, the proposed scheme is capable of resisting impersonation attacks.

6. Efficiency Evaluation

This section presents the implementation details of efficiency evaluation as well as the performance analysis of the proposed scheme. We compare this work with related research on the fields in terms of computation cost performance.

6.1. Implementation

We will evaluate the performance of the proposed work by comparing its computation cost with that of other related works, using the method outlined in [35]. This work adopts a certificateless signature scheme based on bilinear pairing, which is constructed as $G \times G \rightarrow G_T$. Here, we consider that G is an additive cyclic group defined on a super-singular elliptic curve $\overline{E} : y^2 = x^3 + x \ mode \ \overline{p}$, and G_T is a multiplicative cyclic group, where the generator $\overline{P}$ of $\overline{E}$ is generated by a large prime number, $\overline{q}$, of 160 bits, and $\overline{p}$ is a large prime number of 512 bits.

To assess the computational overhead of different cryptographic operations, we conduct a simulation experiment on the Ubuntu 20.04 system, where the processor is configured as Intel(R) Core(TM) i7-10700 CPU @ 2.90 GHz, and the memory is 32 GB. We utilize the MIRACL cryptographic library using the C programming language in the simulation experiment. To eliminate the impact of errors during the experimental process, we perform each cryptographic operation 1000 times and took the average.

The notations for various cryptographic operations are as follows:

- T_{bp} denotes the execution time for the bilinear pairing operation defined as $e(P, Q)$, where $P, Q \in G$.
- $T_{bp.m}$ denotes the execution time for the scalar multiplication operation $x \cdot P$ in the bilinear pairing operation defined as $e(P, Q)$, where $P, Q \in G$ and $x \in Z_q^*$.
- $T_{bp.a}$ denotes the execution time for the point addition operation $P + Q$ in the bilinear pairing operation defined as $e(P, Q)$, where $P, Q \in G$.
- T_{mpt} denotes the execution time for the map-to-point hash function operation in the bilinear pairing operation defined as $e(P, Q)$, where $P, Q \in G$.
- $T_{e.m}$ denotes the execution time for the scalar multiplication operation $x \cdot P$ in ECC, where $P \in G$ and $x \in Z_q^*$.
- $T_{e.a}$ denotes the execution time for the point addition operation $P + Q$ in ECC, where $P, Q \in G$.
- T_h denotes the execution time for one hash function operation in ECC.

The average execution times of these cryptographic operations are shown in Table 2.

Table 2. The execution time of cryptographic operations.

Operations	Execution Times (ms)
T_{bp}	4.1892
$T_{bp.m}$	1.6993
$T_{bp.a}$	0.0071
T_{mpt}	4.3960
$T_{e.m}$	0.4415
$T_{e.a}$	0.0018
T_h	0.0001

6.2. Efficiency Analysis

We mainly focus on the computational overhead of the process of signature verification, while the operations that are very light like the addition operation in Z_q^* and the multiplication operation in Z_q^* will not be considered. By using the computation execution times for various dominant time-consuming cryptographic operations summarized in Table 2, we carry out an efficiency analysis of our proposed scheme compared with two related works: one [35] is based on the bilinear pairing operation, and the other [36] is a certificateless scheme based on ECC. We conduct the computation analysis in terms of the three phases of the signature, individual verification, and aggregate verification. The observation is clear that our proposed scheme has better computation performance compared to related works from Table 3.

Table 3. Comparison of computation costs for related signature verification schemes in ***ms***.

Scheme	Signature	Individual Verification	Aggregate Verification
Mei et al. [35]	$4T_{bp.m} + 2T_{bp.a} + 2T_{mpt} + T_h \approx 15.604$ ms	$4T_{bp} + 2T_{bp.m} + T_h \approx 20.156$ ms	$4T_{bp} + 2nT_{bp.m} + (2n-2)T_{bp.a} + nT_h \approx 3.413n + 16.743$ ms
Vallent et al. [36]	$T_{e.m} + T_h \approx 0.442$ ms	$3T_{e.m} + 3T_{e.a} + 2T_h \approx 1.330$ ms	$(2n+1)T_{e.m} + (4n-1)T_{e.a} + 2nT_h \approx 0.890n + 0.440$ ms
Our scheme	$T_{e.m} + T_h \approx 0.442$ ms	$2T_{e.m} + T_{e.a} + T_h \approx 0.885$ ms	$(n+1)T_{e.m} + (2n-1)T_{e.a} + nT_h \approx 0.445n + 0.434$ ms

In our ECC-based scheme, to generate a signature, a vehicle needs to calculate $R = d \cdot P$ and $H_{EV_A} = H_3(M_{EV_A}, ID_{EV_A}, PK_{EV_A}, R, t)$. This means that the computation cost for the signature is one scalar multiplication operation over an elliptic curve and one hash function operation in ECC, that is to say, $T_{e.m} + T_h \approx 0.442$ ms. In individual verification, $H'_{EV_A} = H_3(M_{EV_A}, ID_{EV_A}, PK_{EV_A}, R, t)$ needs to be calculated and the equation $S_{EV_A} \cdot P = H'_{EV_A} \cdot R + PK_{EV_A}$ needs to be verified. This means that two scalar multiplication operations, one point addition operation and one hash function operation, in ECC are required, that is to say, $2T_{e.m} + T_{e.a} + T_h \approx 0.885$ ms. In aggregate verification, $H'_{EV_{A-i}} = H_3(M_{EV_{A-i}}, ID_{EV_{A-i}}, PK_{EV_{A-i}}, R_i, t_i)$ for n vehicles needs to be calculated and the equation $S \cdot P = \sum_{i=1}^{n} H'_{EV_{A-i}} \cdot R_i + \sum_{i=1}^{n} PK_{EV_{A-i}}$ needs to be verified. This means that $(n+1)$ scalar multiplication operations, $(2n-1)$ point addition operations, and n hash function operations in ECC are required, that is to say, $(n+1)T_{e.m} + (2n-1)T_{e.a} + nT_h \approx 0.445n + 0.434$ ms.

In a similar manner, the computation costs for the other two related schemes can be calculated. In [35], four scalar multiplication operations, two point addition operations, two map-to-point hash function operations in the bilinear pairing operation, and one hash function operation ($4T_{bp.m} + 2T_{bp.a} + 2T_{mpt} + T_h \approx 15.604$ ms) are required for the signature; four bilinear pairing operations, two scalar multiplication operations, and one hash function operation ($4T_{bp} + 2T_{bp.m} + T_h \approx 20.156$ ms) are required for individual verification; and four bilinear pairing operations, $2n$ scalar multiplication operations, $(2n-2)$ point addition operations, and n hash function operations ($4T_{bp} + 2nT_{bp.m} + (2n-2)T_{bp.a} + nT_h \approx$

$3.413n + 16.743$ ms) are required for aggregate verification. In [36], one scalar multiplication operation and one hash function operation in ECC ($T_{e.m} + T_h \approx 0.442$ ms) are required for the signature; three scalar multiplication operations, three point addition operations, and two hash function operations in ECC ($3T_{e.m} + 3T_{e.a} + 2T_h \approx 1.330$ ms) are required for individual verification; and $(2n + 1)$ scalar multiplication operations, $(4n - 1)$ point addition operations, and $2n$ hash function operations in ECC ($(2n + 1)T_{e.m} + (4n - 1)T_{e.a} + 2nT_h \approx 0.890n + 0.440$ ms) are required for aggregate verification.

The visual representation of execution time comparison in the signature and individual verification is shown in Figure 4. We can assemble the computation load generated in message signing and individual verifying for a single signature, assuming equal computation capabilities for signing and verifying for simplicity's sake. The overall load for Mei et al. [35] comes up to $(15.604 + 20.156)$ ms = 35.760 ms, while for Vallent et al. [36], the overall load is $(0.442 + 1.330)$ ms = 1.772 ms. Subsequently, our scheme has an overall computation load of $(0.442 + 0.885)$ ms = 1.327 ms, which is better than other schemes as shown in Figure 4.

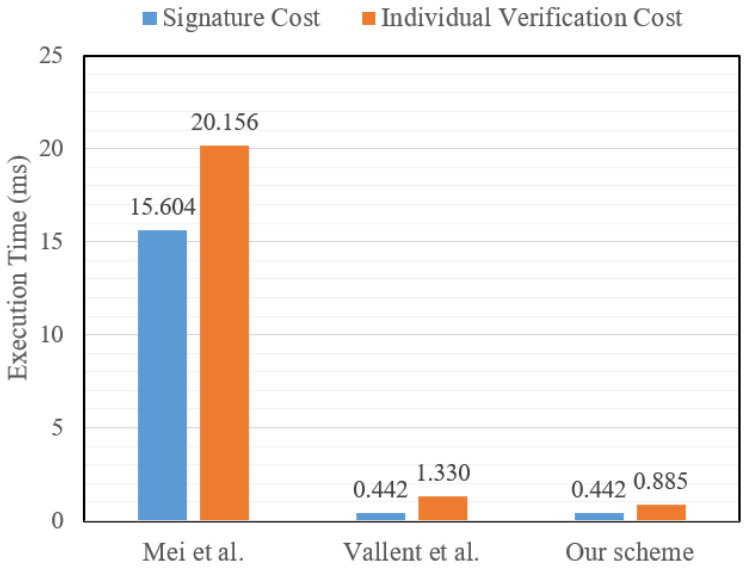

Figure 4. Comparison of execution time in signature and individual verification Mei et al. [35] and Vallent et al. [36].

Compared to Baseline 1 (Mei et al. [35]), we can find that their proposed scheme has a significantly higher execution time than ours, mainly due to its computation complexity in the bilinear pairing operation. However, our proposed scheme precludes complex cryptographic operations like bilinear pairings and map-to-point hash operations by implementing ECC-based algorithms instead, and reduces redundant computational overhead through aggregated signature verification. Compared to Baseline 2 (Vallent et al. [36]), although we both utilize ECC-based methods and have a similarly low computation overhead in the process of the signature, the individual verification cost of our scheme is lower than theirs. That is because we improve the authentication procedures in our carefully designed scheme, therefore reducing one scalar multiplication operation, $T_{e.m}$; two point addition operations, $T_{e.a}$; as well as one hash function operation, T_h, of ECC in the process of individual verification.

The relationship of computation costs for a particular number of signature messages in aggregate verification for the three schemes is shown in Figure 5. As illustrated in Table 3, the aggregate verification cost of our scheme is $(2.968n + 16.309)$ ms and $(0.445n + 0.006)$ ms lower compared to the other two methods, respectively. In Figure 5, we take the maximum value of the number of signature messages n to be 2000. It is clear that, with the increase in the number of signature messages, our proposed scheme will

exhibit a more significant advantage in terms of computational overhead compared to other schemes. Furthermore, being based on ECC as well, our scheme saves approximately half of the computational overhead compared to Vallent et al. [36].

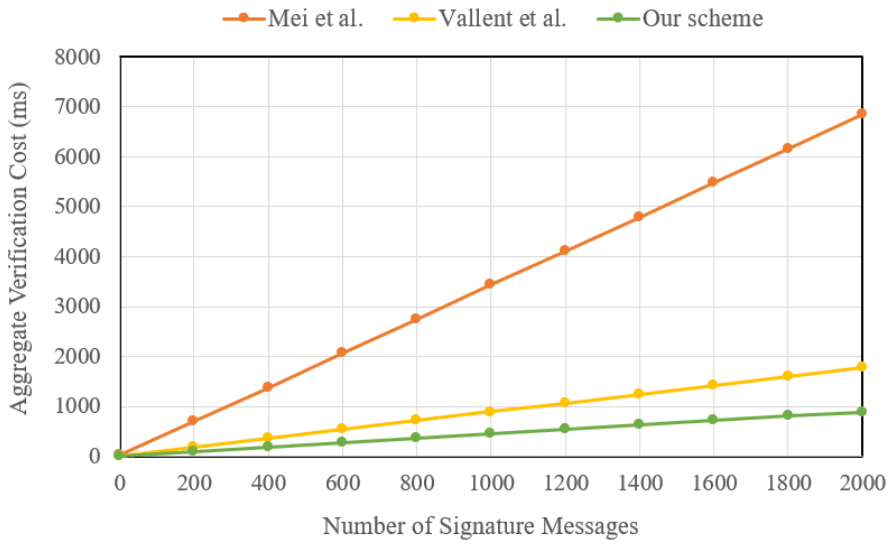

Figure 5. Relationship of computation costs and signature numbers in aggregate verification Mei et al. [35] and Vallent et al. [36].

Based on the generated summary results of computation cost comparison shown in Table 3 and the visual representation shown in Figures 4 and 5, we can come to a conclusion that our proposed scheme has all-over computation efficiency compared to the other two related schemes in terms of the signature, individual verification, and aggregate verification. However, it is essential to note that the security level of the authentication scheme relies on the difficulty of utilized mathematical problems, and any advancements in algorithmic or computational techniques could potentially weaken these assumptions to some extent. Therefore, this is actually a trade-off between security and efficiency.

7. Discussions

7.1. Real-World Scenario

7.1.1. Network Heterogeneity

In our system model, the V2G network adopts a distributed structure, where several distributed grid servers are interconnected via the consortium blockchain, and then each grid server uniformly coordinates and manages multiple charging stations within a certain regional scope. However, its real-world application will be faced with lots of challenges due to the network heterogeneity.

- The application of smart grid and blockchain technology is still in the developmental stage. Many regions still employ centralized PKI architecture in their grid infrastructure, which needs to be gradually adjusted to accommodate the distributed communication and energy transaction demands in a V2G network.
- In practical V2G networks, a significant portion of the purchased charging stations come from third-parties. These charging stations, which originate from different batches, possess varying hardware specifications, charging capacities, and communication interfaces. Therefore, they need to be individually registered and enrolled in the grid server within the respective regions. The grid server then uniformly allocates charging resources and manages communication protocols to ensure the interoperability between different devices and the compatibility with V2G operations.

7.1.2. Varying Computational Capabilities

The proposed scheme assumes that the grid server has strong computational capabilities and resources, while the charging stations and electric vehicles each have a certain level of computational capability, which is sufficient for conducting several cryptographic operations. In real-world scenarios, grid servers typically possess the required computational capability. However, a vast array of different types of charging stations and electric vehicles have varying computational capabilities. Although the proposed scheme has improved the authentication procedures to reduce the computational burden on charging stations and electric vehicles as much as possible, these entities may still encounter difficulties in handling the computational load. In addition, EV users come to the charging station at regular intervals or frequently or random in nature. If more numbers of EVs are coming to the charging station at the same time, a scheduling problem occurs, which is due to the dynamic participation of the EVs in the V2G network, therefore inevitably leading to phenomena such as communication delays and reduced computational efficiency. This paper has not adequately addressed the aforementioned challenges and plans to prioritize them as future work.

7.1.3. Scalability

It is crucial to ensure that the proposed scheme is capable of accommodating an expanding V2G network with the substantial number of verified vehicles and growing transaction volumes. As illustrated in Table 3 and shown in Figure 5, by leveraging the aggregate verification method, the computational cost of this scheme increases linearly with the growing number of vehicles to be verified and transaction volume n, which means that the time complexity is $O(n)$. In addition, when large numbers of vehicles are arriving at the charging station, the fine-grained access control technique [37] can be used, in which the electric vehicles are arranged in the queue and priority is given to the first-come one, thereby meeting the scalability requirements.

7.2. Potential Privacy Threats

As analyzed in Section 5, although our proposed scheme possesses certain security and privacy-preserving features, specifically including message integrity, anonymity, unlinkability, traceability, and resistance to common attacks, there still exists a range of potential privacy compromises.

- This scheme assumes that the grid server and *KGC* are fully trusted entities, and their compromise would pose severe security and privacy threats. Therefore, the grid needs to strengthen security oversight of these entities, particularly guarding against cyber–physical attacks or social engineering attacks.
- This scheme assumes that the negotiated key between the charging station and grid server is transmitted through a secure channel. However, in actual scenarios, the technology used in the communication between these two entities is often based on wireless networks, making it susceptible to various eavesdropping techniques [38]. If the negotiated key is intercepted by attackers through an eavesdropping attack, it may pose severe security risks. One feasible countermeasure is to periodically update the negotiated key between these two entities.
- The EV connects to the charging station in the public area network, and the payment is usually carried out through the mobile phone application, where the user's private data are susceptible to be stolen. In addition, although blockchain-based transactions are anonymous, once associated with real identities, they may leak sensitive information of individuals or organizations. In some cases, through techniques like deep learning, it is possible for attackers to infer participants' identities by analyzing their transaction patterns or habits. The countermeasures against the above security and privacy challenges require further research.

8. Conclusions

In this paper, we proposed a privacy-preserving and efficient cross-domain authentication scheme for V2G networks in a smart grid based on consortium blockchain and certificateless signature technology. We adopted elliptic curve cryptography and the UTXO mechanism as the backbone, and systematically presented the detailed process of this scheme. In the aspect of security, the proposed work simultaneously achieves message integrity, anonymity, unlinkability, traceability, as well as resistance to common attacks through a theoretical analysis, thus satisfying the security requirements for V2G networks. As for efficiency, the scheme precludes complex cryptographic operations like bilinear pairing and map-to-point hash function operations. Furthermore, in the scenario of aggregated verification, the charging station aggregates signatures from multiple vehicles and submits them to the grid server for unified verification, thereby reducing redundant computational overhead and further improving the performance on computation efficiency. Therefore, the proposed cross-domain, certificateless, and consortium-blockchain-based authentication method proves to be a comparatively secure and efficient scheme suitable for V2G applications in the smart grid.

Author Contributions: Conceptualization, Q.M., J.D., Y.C. and W.X.; methodology, formal analysis, and investigation, Q.M.; software, Q.M. and T.R.; writing—original draft, Q.M.; writing—review and editing, Q.M., T.R., J.D., Y.C. and W.X.; validation and visualization, Q.M. and T.R.; supervision, J.D., Y.C. and W.X.; project administration, Y.C. and W.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the science and technology project of State Grid Corporation of China: "Research on Key Technologies of Multi-agent Trusted Interaction and Monitoring Response for New-type Power System User Side Business" (Grant No. 5108-202218280A-2-405-XG).

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: Authors Tianyu Ren and Jiahan Dong were employed by the company State Grid Beijing Electric Power Research Institute. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

V2G	Vehicle-to-Grid
IP	Internet Protocol
RSA	Rivest–Shamir–Adleman
SIoT	Social Internet of Things
PKI	Public Key Infrastructure
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
ECDL	Elliptic Curve Discrete Logarithm
ECCDH	Elliptic Curve Computational Diffie–Hellman
MAC	Message Authentication Code
UTXO	Unspent Transaction Output
EV	Electric Vehicle
CS	Charging Station
GS	Grid Server
KGC	Key Generation Center
OBU	On-Board Unit
RSU	Road-Side Unit

References

1. Kempton, W.; Tomić, J. Vehicle-to-grid power fundamentals: Calculating capacity and net revenue. *J. Power Sources* **2005**, *144*, 268–279. [CrossRef]
2. Park, J.; Kim, H.; Choi, J.Y. Improving TCP performance in vehicle-to-grid (V2G) communication. *Electronics* **2019**, *8*, 1206. [CrossRef]
3. Pazos-Revilla, M.; Alsharif, A.; Gunukula, S.; Guo, T.N.; Mahmoud, M.; Shen, X. Secure and privacy-preserving physical-layer-assisted scheme for EV dynamic charging system. *IEEE Trans. Veh. Technol.* **2017**, *67*, 3304–3318. [CrossRef]
4. Sovacool, B.K.; Hirsh, R.F. Beyond batteries: An examination of the benefits and barriers to plug-in hybrid electric vehicles (PHEVs) and a vehicle-to-grid (V2G) transition. *Energy Policy* **2009**, *37*, 1095–1103. [CrossRef]
5. Guille, C.; Gross, G. A conceptual framework for the vehicle-to-grid (V2G) implementation. *Energy Policy* **2009**, *37*, 4379–4390. [CrossRef]
6. Fernandez, L.P.; San Román, T.G.; Cossent, R.; Domingo, C.M.; Frias, P. Assessment of the impact of plug-in electric vehicles on distribution networks. *IEEE Trans. Power Syst.* **2010**, *26*, 206–213. [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. Raya, M.; Hubaux, J.P. Securing vehicular ad hoc networks. *J. Comput. Secur.* **2007**, *15*, 39–68. [CrossRef]
9. Sun, Y.; Lu, R.; Lin, X.; Shen, X.; Su, J. An efficient pseudonymous authentication scheme with strong privacy preservation for vehicular communications. *IEEE Trans. Veh. Technol.* **2010**, *59*, 3589–3603. [CrossRef]
10. Abdallah, A.; Shen, X.S. Lightweight authentication and privacy-preserving scheme for V2G connections. *IEEE Trans. Veh. Technol.* **2016**, *66*, 2615–2629. [CrossRef]
11. Shen, J.; Zhou, T.; Wei, F.; Sun, X.; Xiang, Y. Privacy-preserving and lightweight key agreement protocol for V2G in the social Internet of Things. *IEEE Internet Things J.* **2017**, *5*, 2526–2536. [CrossRef]
12. Eiza, M.H.; Shi, Q.; Marnerides, A.K.; Owens, T.; Ni, Q. Efficient, secure, and privacy-preserving PMIPv6 protocol for V2G networks. *IEEE Trans. Veh. Technol.* **2018**, *68*, 19–33. [CrossRef]
13. Roman, L.F.; Gondim, P.R.; Lloret, J. Pairing-based authentication protocol for V2G networks in smart grid. *Ad Hoc Netw.* **2019**, *90*, 101745. [CrossRef]
14. Park, K.; Park, Y.; Das, A.K.; Yu, S.; Lee, J.; Park, Y. A dynamic privacy-preserving key management protocol for V2G in social internet of things. *IEEE Access* **2019**, *7*, 76812–76832. [CrossRef]
15. Su, Y.; Shen, G.; Zhang, M. A novel privacy-preserving authentication scheme for V2G networks. *IEEE Syst. J.* **2019**, *14*, 1963–1971. [CrossRef]
16. Secchi, M.; Barchi, G.; Macii, D.; Petri, D. Smart electric vehicles charging with centralised vehicle-to-grid capability for net-load variance minimisation under increasing EV and PV penetration levels. *Sustain. Energy Grids Netw.* **2023**, *35*, 101120. [CrossRef]
17. Reddy, A.G.; Babu, P.R.; Odelu, V.; Wang, L.; Kumar, S.A. V2G-Auth: Lightweight Authentication and Key Agreement Protocol for V2G Environment leveraging Physically Unclonable Functions. *IEEE Trans. Ind. Cyber Phys. Syst.* **2023**, *1*, 66–78. [CrossRef]
18. Aitzhan, N.Z.; Svetinovic, D. Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams. *IEEE Trans. Dependable Secur. Comput.* **2016**, *15*, 840–852. [CrossRef]
19. Guan, Z.; Si, G.; Zhang, X.; Wu, L.; Guizani, N.; Du, X.; Ma, Y. Privacy-preserving and efficient aggregation based on blockchain for power grid communications in smart communities. *IEEE Commun. Mag.* **2018**, *56*, 82–88. [CrossRef]
20. Garg, S.; Kaur, K.; Kaddoum, G.; Gagnon, F.; Rodrigues, J.J. An efficient blockchain-based hierarchical authentication mechanism for energy trading in V2G environment. In Proceedings of the 2019 IEEE International Conference on Communications Workshops (ICC Workshops), Shanghai, China, 20–24 May 2019; pp. 1–6.
21. Wang, H.; Wang, Q.; He, D.; Li, Q.; Liu, Z. BBARS: Blockchain-based anonymous rewarding scheme for V2G networks. *IEEE Internet Things J.* **2019**, *6*, 3676–3687. [CrossRef]
22. Ali, I.; Gervais, M.; Ahene, E.; Li, F. A blockchain-based certificateless public key signature scheme for vehicle-to-infrastructure communication in VANETs. *J. Syst. Archit.* **2019**, *99*, 101636. [CrossRef]
23. Patil, A.S.; Hamza, R.; Hassan, A.; Jiang, N.; Yan, H.; Li, J. Efficient privacy-preserving authentication protocol using PUFs with blockchain smart contracts. *Comput. Secur.* **2020**, *97*, 101958. [CrossRef]
24. Miller, V.S. Use of elliptic curves in cryptography. In Proceedings of the Conference on the Theory and Application of Cryptographic Techniques, Santa Barbara, CA, USA, 18–22 August 1985; pp. 417–426.
25. Johnson, D.; Menezes, A.; Vanstone, S. The elliptic curve digital signature algorithm (ECDSA). *Int. J. Inf. Secur.* **2001**, *1*, 36–63. [CrossRef]
26. Menezes, A. *Evaluation of Security Level of Cryptography: The Elliptic Curve Discrete Logarithm Problem (ECDLP)*; University of Waterloo: Waterloo, ON, Canada, 2001.
27. Boneh, D. The decision diffie-hellman problem. In Proceedings of the International Algorithmic Number Theory Symposium, Portland, OR, USA, 21–25 June 1998; pp. 48–63.
28. Dib, O.; Brousmiche, K.L.; Durand, A.; Thea, E.; Hamida, E.B. Consortium blockchains: Overview, applications and challenges. *Int. J. Adv. Telecommun.* **2018**, *11*, 51–64.
29. Li, Z.; Kang, J.; Yu, R.; Ye, D.; Deng, Q.; Zhang, Y. Consortium blockchain for secure energy trading in industrial internet of things. *IEEE Trans. Ind. Inform.* **2017**, *14*, 3690–3700. [CrossRef]

30. Che, Z.; Wang, Y.; Zhao, J.; Qiang, Y.; Ma, Y.; Liu, J. A distributed energy trading authentication mechanism based on a consortium blockchain. *Energies* **2019**, *12*, 2878. [CrossRef]
31. McGinn, D.; Birch, D.; Akroyd, D.; Molina-Solana, M.; Guo, Y.; Knottenbelt, W.J. Visualizing dynamic bitcoin transaction patterns. *Big Data* **2016**, *4*, 109–119. [CrossRef] [PubMed]
32. Vallois, V.; Guenane, F.A. Bitcoin transaction: From the creation to validation, a protocol overview. In Proceedings of the 2017 1st Cyber Security in Networking Conference (CSNet), Rio de Janeiro, Brazil, 18–20 October 2017; pp. 1–7.
33. Delgado-Segura, S.; Pérez-Sola, C.; Navarro-Arribas, G.; Herrera-Joancomartí, J. Analysis of the bitcoin utxo set. In Proceedings of the Financial Cryptography and Data Security: FC 2018 International Workshops, BITCOIN, VOTING, and WTSC, Nieuwpoort, Curaçao, 2 March 2018; pp. 78–91.
34. Rajasekaran, A.S.; Azees, M.; Al-Turjman, F. A comprehensive survey on security issues in vehicle-to-grid networks. *J. Control Decis.* **2023**, *10*, 150–159. [CrossRef]
35. Mei, Q.; Xiong, H.; Chen, J.; Yang, M.; Kumari, S.; Khan, M.K. Efficient certificateless aggregate signature with conditional privacy preservation in IoV. *IEEE Syst. J.* **2020**, *15*, 245–256. [CrossRef]
36. Vallent, T.F.; Hanyurwimfura, D.; Mikeka, C. Efficient certificate-less aggregate signature scheme with conditional privacy-preservation for vehicular ad hoc networks enhanced smart grid system. *Sensors* **2021**, *21*, 2900. [CrossRef]
37. Xu, S.; Yang, G.; Mu, Y.; Deng, R.H. Secure fine-grained access control and data sharing for dynamic groups in the cloud. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 2101–2113. [CrossRef]
38. Novak, A.; Ivanov, A. Network Security Vulnerabilities in Smart Vehicle-to-Grid Systems Identifying Threats and Proposing Robust Countermeasures. *J. Artif. Intell. Mach. Learn. Manag.* **2023**, *7*, 48–80.