2022 IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom) | 978-1-6654-9425-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/TrustCom56396.2022.00066

CECAS: A cloud-edge collaboration authentication scheme based on V2G short randomizable signature

Zhuoqun Xia*, Hongrui Li†,and Ke Gu‡
*†‡Changsha University of Science and Technology, Changsha 410114, China
*xiazhuoqun@csust.edu.cn, †839390265@qq.com, ‡gk4572@163.com

***Abstract*—With the growing demand for energy-efficient and environmentally friendly transportation solutions, electric vehicles have become an emerging mode of transportation. However, a large number of security issues have affected its further development. Existing schemes use identity-based restricted partial blind signature technique to implement authentication schemes with high computational cost. Local aggregators suffer from tampering with transaction records. Malicious EVs continuously listen to power transactions to infer the lifestyle of EV users. This scheme designs a cloud-edge collaboration authentication scheme based on V2G short randomizable signature. Specifically, privacy during EV power transactions is guaranteed using lightweight cryptographic primitives. EVs evaluate the reputation value of local aggregators through additive secret sharing. Cloud computing center and edge servers are capable of collaborative authentication and collaborative tracking. This scheme can improve the authentication efficiency of V2G network.**

***Index Terms*—V2G networks, Cloud-edge collaboration, Lightweight, Identity tracking, Identity authentication.**

I. INTRODUCTION

Electric vehicles(EVs) are widely popular in reducing air pollution caused by fossil fuel-driven vehicles. Studies have shown that carbon dioxide emissions may be reduced by 70% if fossil fuel-driven vehicles are replaced by EVs [1]. Governments in various countries have developed strategies to encourage people to choose EVs and use them as the best mode of transportation [2]. Commercial charging stations are emerging around the globe such as in Asia, the United States and Europe. By 2027, global demand for EVs will reach $802.8 billion [3]. To accelerate the transition, many governments have decided to stop selling all new non-electric vehicles, including gasoline, diesel, and hybrid vehicles, starting in 2035 [4]. V2G can help the grid selection process for efficient charging of EVs [9]. However, there are privacy and security issues in V2G networks. On the one hand, charging stations deployed in various locations can have message tampering, delay or spoofing. On the other hand, there are problems with sensitive data and hardware corruption at EV charging stations. Before the grid interacts with EVs, they should verify messages with each other. Without this process, a malicious adversary can impersonate an illegal charging station and steal messages. In addition, a malicious adversary can also impersonate an unregistered EV to obtain information.

Liu et al. [11] proposed a battery state-aware authentication scheme based on the battery state, which considers the charging, fully charged and discharged states of the battery. In addition, Liu et al. [10] proposed an aggregated proof-based authentication scheme, which achieves EV authentication by verifying a set of EVs and establishing aggregated proofs. However, all these schemes have huge overhead and are not suitable for V2G networks that require fast and efficient authentication. Yang et al. [5] designed an exact reward scheme in V2G networks. The scheme uses identity-based restricted partial blind signatures [8] to generate licenses for EVs. The scheme guarantees in this way that the message verifier is not able to connect to the real identity of EV through the license. Later Yang et al. [5] scheme was shown to be problematic because the license generated in Yang's scheme is susceptible to forgery by malicious EVs [8]. Wang et al. [6] proposed a traceable and accurate reward scheme based on Yang [5]. Wang et al. [7] designed an anonymous authentication scheme with conditional privacy protection. The above schemes are based on symmetric bilinear group construction, they have large computational and communication costs.

Based on the above discussion, this scheme proposes a cloud-edge collaboration authentication scheme based on V2G short randomizable signature. The main contributions of this scheme are as follows.

- A cloud-edge collaboration authentication scheme based on V2G short randomizable signature is proposed, our scheme is based on short randomizable signatures and hash functions to safeguard the privacy of EVs and provide lightweight anonymous authentication between network entities.
- EV prevents local aggregators from tampering with transaction information by evaluating the reputation value of local aggregators.
- The experiment verifies that our scheme has low computational costs and communication costs.

The rest of this paper is organized as follows. We provide the preliminaries in Section II. The system definition is presented in Section III. Our scheme is presented in Section IV. The evaluation is presented in Section V. Finally, we conclude this paper in Section VI.

II. PRELIMINARIES

A. (Asymmetric) Bilinear Groups of Prime Order

The bilinear group is the set of three cyclic groups of prime order p, G_1 and G_2 and the bilinear map $e : G_1 \times G_2 \rightarrow G_T$, which has the following properties.

- Bilinear: For all $g \in G_1$, $\tilde{g} \in G_2$ and $a, b \in G_p$, there are $e(g^a, \tilde{g}^b) = e(g, \tilde{g})^{a \cdot b}$.

2324-9013/22/$31.00 ©2022 IEEE
DOI 10.1109/TrustCom56396.2022.00066

- Non-degeneracy: for $g \neq 1_{G_1}$ and $\tilde{g} \neq 1_{G_2}$, $e(g, \tilde{g}) \neq 1_{G_T}$; .
- Computability: the bilinear pairings e are efficiently computable.

B. Short Randomizable Signature

A new signature scheme proposed by David Pointcheval and Olivier Sanders in the scheme [12], this scheme has the same characteristics as the CL signature [13], but it does not have the disadvantage of linear size: the signature of the short random signature scheme contains only two elements and the algorithm performs more efficiently.

$Setup(1^k)$: Given a security parameter k, the algorithm outputs the public parameter $pp \leftarrow (p, G_1, G_2, G_T, e)$.

$Keygen(pp)$: Given the public parameters pp, the $\hat{g}$ and signature keys $sk = (x, y)$ are selected and the corresponding verification keys $pk = (\hat{g}, \hat{X}, \hat{Y})$ are computed.

$Sign(sk, m)$: Given a signature key sk and a message m, a random number r is selected and a signature σ is output.

$Verify(pk, m, \sigma)$: Given a verification key pk, a message m, and a signature σ. If the verification passes, output 1; otherwise, output 0.

$GSetup(1^k)$: A group administrator executes the $Setup$ and $Keygen$ algorithm and obtains $sk = (x, y)$ and $pk = (\hat{g}, \hat{X}, \hat{Y})$, sets gpk the group public key to pk and a generation element g, sets the group private key gsk to sk, and selects a hash function $\mathcal{H}$.

$PKIJoin(i, 1^k)$: A user i generates $(sk_i, pk_i) \leftarrow Keygen(pp)$ and sends pk_i to a certification authority.

$GJoin$: A user i generates a secret s_i and sends and a signature η, and the group administrator verifies η and $(\tau, \hat{\tau})$. If the user passes the s_i Schnorr interactive knowledge proof [14], the group administrator chooses a random number h and computes $\hat{\sigma} \leftarrow (\hat{\sigma}_1, \hat{\sigma}_2)$. Finally the group administrator records $(i, \tau, \eta, \hat{\tau})$ and sends $\hat{\sigma}$ to the user $i(s_i, \hat{\sigma}, e(\hat{\sigma}_1, \hat{Y}))$ as gsk_i.

$GSign(gsk_i, m)$: The user i selects a random number u and computes $e(\hat{\sigma}_1, \hat{Y})^u$ to randomized σ, selects a random number t and computes $e(\hat{\sigma}_1, \hat{Y})^u \leftarrow e(\sigma_1, \hat{Y})^{tu}$, then computes $ss \leftarrow u + c \cdot s_i$. Then user i outputs a group signature μ_i on the message m.

$GVerify(gpk_i, m, \mu_i)$: A verifier verifies the signature μ_i on the message m and outputs 1 if it passes otherwise it outputs 0.

$GOpen(gsk, m, \mu_i)$: The group manager queries all items $(i, \tau_i, \eta_i, \hat{\tau}_i)$. Then it returns a match (i, τ_i, η_i) and a valid proof $\hat{\tau}_i$ of knowledge.

C. Additive Secret Sharing

The additive secret sharing [15] proceeds as follows: given an initial domain F, the domain Z_p of prime numbers p, and a secret $s \in F$, the share of the secret s consists of shares $s_1, ..., s_n$, where $s = s_1 + s_2 + \cdots + s_n$. $s_1, s_2, ..., s_{n-1}$ is chosen randomly from F and $s_n = s - (s_1 + s_2 + \cdots + s_{n-1})$. Let $S_n : F \rightarrow F^n$ be the additive secret sharing function which outputs the share of the input n. Let $S_n(s)[i]$ be the i share of s. Given the shares n, it can simply add these shares to reconstruct s.

III. SYSTEM DEFINITION

A. The System Model

The entities in the system model mainly include a cloud computing center, edge servers, electric vehicles, local aggregators, and charging stations.

Cloud Computing Center: The cloud computing center is a third-party organization that is responsible for the initialization of security parameters in this system.

Edge Servers: Edge servers are distributed at the edge of the network and are responsible for processing data transmitted by local aggregators.

Electric Vehicles: Each electric vehicle is fitted with an On-Board Unit (OBU) tamper-proof device.

Local Aggregator: The local aggregator is a communication intermediary. EVs communicate with edge servers, cloud computing centers, and charging stations in the network through it.

Charging stations: Charging stations are responsible for energy transactions from EVs to the grid in the network.

IV. PROPOSED SCHEME

This section describes the proposed scheme. The system parameters are given in Table 1.

TABLE I
SUMMARY OF NOTATIONS

Notations	Descriptions
$p; G_1, G_2, G_T$	Prime numbers; Cyclic groups
$g, \hat{g}; e$	Group generating element; Bilinear pairing
$\oplus$	XOR operation
$H(.), H_0^X(.)$	Hash functions
$(x, y); (g, X, Y)$	Group private key; Group public key
SK_x	Session key for entity x
β_{ev}	Fingerprints of electric vehicle users
T	Timestamp
pk_x, sk_x	Public and private key of entity x
$Token$	Trading tokens

A. Initialization

Each EV in the V2G network needs to be registered, and EV chooses a random number $sk_{ev} \in Z_p$ as its private key and calculates $pk_{ev} = g^{sk_{ev}}$ as its public key. Then EV sends its real identity ID_{ev} to cloud computing center, cloud computing center calculates EV's pseudonym $PID_{ev} = H_0^X(ID_{ev} \parallel VT_i)$. Then cloud computing center select a hash key $key_i \in X$ where VT_i is the valid time for the pseudonym of EV. Cloud computing center calculates $O_{ev} = H(PID_{ev})$ and sends it to EV through a secure channel, EV stores $< H[O_{ev}, ID_{ev}, sk_{ev}, pk_{ev}, PID_{ev}] \oplus \beta_{ev}, pk_{ev} >$ in a tamper-proof device. The tamper-proof device ensures that the stored data is not damaged by any computational feasibility effects [17].

Edge server in the V2G network also needs to register. Edge server chooses a random number $sk_{es} \in Z_p$ as its private key

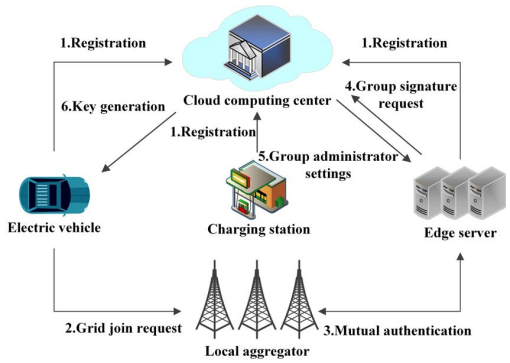

Fig. 1. Initialization.

and computes $pk_{es} = g^{sk_{es}}$ as its public key. Edge server acts as a cluster manager to collect electricity transaction requests $request_{ev_1}$ from EVs. EV sends a request to edge server to join the V2G network. The EV user first enters his fingerprint $\alpha_{ev} = h(\beta_{ev})$ and then generates a random number $n_1 \in Z_p$ then computes $sk_{ev} = sk'_{ev} \oplus h(\beta_{ev})$, $SK = (pk_{es})^{sk_{ev}} = g^{sk_{es} \cdot sk_{ev}}$, $request_{ev_1} = request'_{ev_1} \oplus h(\beta_{ev})$ and $vm_1 = h(PID_{ev} \parallel n_1 \parallel sk_{ev} \parallel T_1)$. EV sends a message $m_1 : \{PID_{ev}, n_1, vm_1\}$ to edge server. Edge server generates a random number $n_2 \in Z_p$ then computes $SK = (pk_{ev})^{sk_{es}} = g^{sk_{ev} \cdot sk_{es}}$ $vm_2 = h(ID_{es} \parallel n_2 \parallel T_2)$. Edge server sends a request $m_2 : \{m_1, ID_{es}, n_2, vm_2\}$ to cloud computing center to generate a group signature.

After receiving the group signature generation request from edge server, the cloud center selects a security parameter k, and generates the public parameter $pp \leftarrow (P, G_1, G_2, G_T, e, H)$, $sk = (x, y)$ and $pk = (\hat{g}, \hat{X}, \hat{Y})$ where $\hat{g} \leftarrow G_2$ and $(\hat{X}, \hat{Y}) \leftarrow (\hat{g}^x, \hat{g}^y)$. The cloud center generates the group public key gpk as pk and $g \in G_1$, and the group private key gsk as sk. G_1, G_2 and G_T are three cyclic groups of prime orders p. e is a bilinear mapping: $G_1 \times G_2 \rightarrow G_T$, and H is a hash function.

B. Mutual authentication

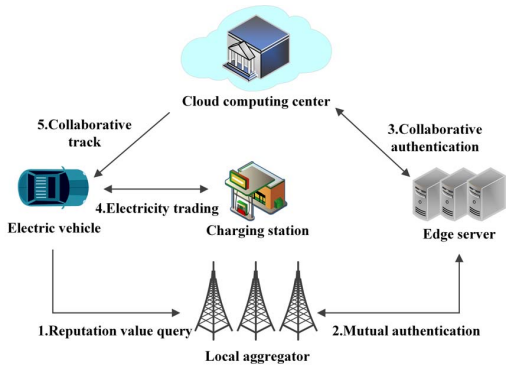

Fig. 2. Mutual authentication.

1) Mutual Authentication of Electric Vehicles and Edge Servers: EV chooses a secret $st_{ev} \in Z_p$ and sends both $(\tau_{ev}, \hat{\tau}_{ev}) \leftarrow (g^{st_{ev}}, \hat{Y}^{st_{ev}})$ and a signature $\eta_{ev} \leftarrow Sign(sk_{ev}, \tau_{ev})$ to cloud computing center. Cloud computing center verifies the equations $e(\tau_{ev}, \hat{Y}) = e(g, \hat{\tau}_{ev})$, $(\tau_{ev}, \hat{\tau}_{ev})$ and η_{ev}. If EV passes the Schnorr interactive knowledge proof of sk_{ev} [14], cloud computing center chooses the random number $u_{ev} \in Z_p$, then computes $\hat{\sigma}_{ev} = (\hat{\sigma}_{ev_1}, \hat{\sigma}_{ev_2}) = (g^{u_{ev}}, (g^x \cdot \tau_{ev}^y)^{u_{ev}})$. Finally, cloud computing center records $(ev, \tau_{ev}, \eta_{ev}, \hat{\tau}_{ev})$ and sends $\hat{\sigma}_{ev}$ to EV and then sets the group private key $gsk_{ev} = (u_{ev}, \hat{\sigma}_{ev}, e(\hat{\sigma}_{ev1}, Y))$ for EV.

For EVs with charging requests, they transmit their transaction messages to edge server through local aggregator. EV needs to evaluate the reputation value of local aggregator to prevent local aggregator from tampering with the transaction records. EV acts as the querying party and it wants to calculate the reputation value v_{lag} of local aggregator. EV first sends a query request $request_{ev_2}$ and then EV generates a random number $n_3 \in Z_p$, then computes $request_{ev_2} = request'_{ev_2} \oplus h(\beta_{ev})$, $SK = (pk_{lag})^{sk_{ev}} = g^{sk_{lag} \cdot sk_{ev}}$ and $vm_3 = h(PID_{ev} \parallel n_3 \parallel sk_{ev} \parallel s \parallel T_3)$. EV sends a message $m_3 : \{PID_{ev}, n_3, vm_3\}$ to local aggregator. Local aggregator generates a random number $n_4 \in Z_p$ then computes $SK = (pk_{ev})^{sk_{lag}} = g^{sk_{ev} \cdot sk_{lag}}$ and $vm_4 = h(ID_{lag} \parallel n_4 \parallel T_4)$.

Local aggregator returns the reputation values v_{ik} and $m_4 : \{m_3, ID_{lag}, n_4, vm_4, v_{lag}\}$, where $s = s_1 + s_2 + \cdots + s_{|U|}$. $s_1, s_2, ..., s_{|U|}$ are the shared secrets in the witness set U. U is the set of witnesses for the computational input. The query scheme P_{add} is as follows: 1. ev_i send the description and identity i_k to each party in the set U; 2. for each $ev_x \in U$ calculate $(s_1, ..., s_{|U|}) = S_{|U|}(v_{xk})$ and send a share in the set U and keep a share for itself; 3. each ev_x get a part of the share from the other party in the set U, $(r_1, ..., r_{|U|})$ is the share it collects; 4. each ev_x calculate $t_x = r_1 + r_2 + \cdots + r_{|U|}$ and send the share t_x to ev_i; 5. ev_i receive the share $|U|$ from ev_x and calculate $v_{ik} = \frac{1}{|U|} \sum_{ev_x \in U} t_x$ the reputation value of local aggregator.

EV selects a random number $r_0 \in Z_p$ to encrypt the transaction message m_{ev}, and computes $em_{ev} = (em^1_{ev}, em^2_{ev})$, where $em^1_{ev} = g^{r_0}$ and $em^2_{ev} = m_{ev} \cdot PK^{r_0}_{es}$. EV select a random number u_0, then computes $(\hat{\sigma}'_{ev1}, \hat{\sigma}'_{ev2}) \leftarrow (\hat{\sigma}^{u_0}_{ev1}, \hat{\sigma}^{u_0}_{ev2})$. EV select a random number t_0, then computes $e(\hat{\sigma}_{ev1}, \hat{Y})^{u_0} \leftarrow e(\sigma_{ev1}, \hat{Y})^{t_0 u_0}$, $c_{ev} \leftarrow H(\hat{\sigma}'_{ev1}, \hat{\sigma}'_{ev2}, e(\hat{\sigma}_{ev1}, \hat{Y})^{t_0 u_0}, em_{ev})$ and $s_0 \leftarrow u_0 + c_{ev} \cdot st_{ev}$. Then EV output a group signature $\mu_{ev} = (\hat{\sigma}'_{ev1}, \hat{\sigma}'_{ev2}, c_{ev}, s_0)$ on em_{ev}. This message is $ms_{ev} = (em_{ev}, \hat{\sigma}'_{ev1}, \hat{\sigma}'_{ev2}, c_{ev}, s_0)$, then EV sends ms_{ev} to edge server. When a message ms_{ev} is received from EV, edge server first verifies ms_{ev}. The verification process is as follows:

$$D_{ev} \leftarrow (e(\sigma'^{-1}_{ev1}, \hat{X}) \cdot e(\sigma'_{ev2}, \hat{g}))^{-c_{ev}} \cdot e(\sigma'^{s_0}_{ev1}, \hat{Y})$$

The message $c_{ev} = H(\hat{\sigma}'_{ev1}, \hat{\sigma}'_{ev2}, D_{ev}, em_{ev})$ is then checked. Edge server generates the appropriate signature ρ^{ev}_{es} for the message ms_{ev} and returns μ_{es} to EV, μ^{ev}_{es} proves the validity of EV's message.

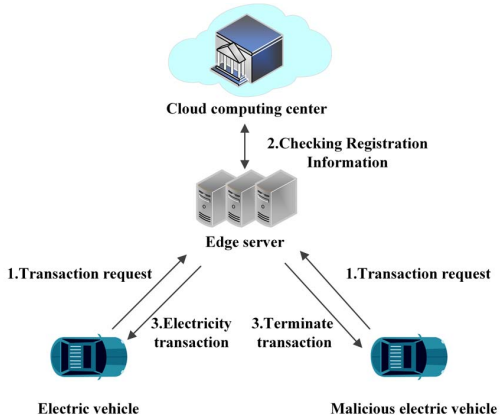

Fig. 3. Collaborative authentication.

2) Cloud Computing Center and Edge Server Collaborative Authentication: Edge server requests cloud computing center to collaboratively authenticate the message sent by EV, edge server sends a collaborative authentication request $request_{es}$. Edge server first enters its private key $\alpha_{es} = h(sk_{es})$, and generates a random number $n_5 \in Z_p$. Edge server computes $SK = (pk_{cc})^{sk_{es}} = g^{sk_{cc} \cdot sk_{es}}$, $request_{es} = request'_{es} \oplus h(sk_{es})$ and $vm_5 = h(ID_{es} \parallel n_5 \parallel sk_{es} \parallel T_5)$. Then edge server sends a message to cloud computing center, it computes $SK = (pk_{es})^{sk_{cc}} = g^{sk_{es} \cdot sk_{cc}}$ and $vm_6 = h(ID_{cc} \parallel n_6 \parallel T_6)$. Cloud computing center checks the registration message of EV, then cloud computing center sends a token and message $m_6 : \{m_5, n_6, vm_6\}$ to EV for EV that has the registration message. Cloud computing center signs the message vector $(m_1, ..., m_n) \in Z_p^n$ at once, cloud computing center selects $\hat{g} \in G_2$ and $(x, y_1, ..., y_n) \in Z_p^{n+1}$. Then cloud computing center computes $(\hat{X}, \hat{Y}_1, ..., \hat{Y}_n) \leftarrow (\hat{g}^x, \hat{g}^{y_1}, ..., \hat{g}^{y_n})$ where the private key sk is $(x, y_1, ..., y_n)$ and the public key pk is $(\hat{g}, \hat{X}, \hat{Y}_1, ..., \hat{Y}_n)$.

Cloud computing center signs the message. Cloud computing center selects $h \leftarrow G_1^*$ and computes $\sigma \leftarrow (h, h^{(x+\sum y_j \cdot m_j)})$, where $G_1^* = G_1 \backslash \{1_{G_1}\}$. To verify the message from cloud computing center, EV first resolves σ to (σ_1, σ_2) and then verifies $e(\sigma_1, \hat{X} \cdot \prod \hat{Y}_j^{m_j}) = e(\sigma_2, \hat{g})$ and $\sigma_1 \neq 1_{G_1}$. EV resolve σ to obtain the signature sent by cloud computing center. The cloud center first selects $g \leftarrow G_1^*$, $\hat{g} \leftarrow G_2^*$ and $(x, y_1, ..., y_n) \in Z_p^{n+1}$, where $G_1^* = G_1 \backslash \{1_{G_1}\}$, $G_2^* = G_2 \backslash \{1_{G_2}\}$.

Cloud computing center computes $(X, Y_1, ..., Y_n) \leftarrow (g^x, g^{y_1}, ..., g^{y_n})$ and $(\hat{X}, \hat{Y}_1, ..., \hat{Y}_n) \leftarrow (\hat{g}^x, \hat{g}^{y_1}, ..., \hat{g}^{y_n})$. Then cloud computing center set the private key $sk \leftarrow X$, public key $pk \leftarrow (g, Y_1, ..., Y_n, \hat{g}, \hat{X}, \hat{Y}_1, ..., \hat{Y}_n)$. To get the signature on the message vector $(m_1, ..., m_n)$, EV first chooses a random number $t_1 \in Z_p$ and computes $C \leftarrow g^{t_1} \prod_{i=1}^n Y_i^{m_i}$.Then EV sends C to cloud computing center. Cloud computing center and EV both run an open commitment to proof of knowledge. If cloud computing center is convinced, it chooses a random number $u_1 \in Z_p$ and returns $\sigma' \leftarrow (g^{u_1}, (XC)^{u_1})$. Cloud computing center computes $\sigma \leftarrow (\sigma'_1, {\sigma'_2}/{\sigma'_1}^{t_1})$. The elements σ meet $\sigma_1 = g^{u_1}$ and $\sigma_2 = (XC)^{u_1}/g^{u_1 t_1}$.

After EV resolves the signature σ, it needs to be determined that it is authenticated by the cloud center and that EV needs to remain anonymous. $pk \leftarrow (\hat{g}, \hat{X}, \hat{Y}_1, ..., \hat{Y}_n)$ is the public key and $\sigma = (\sigma_1, \sigma_2)$ is a valid signature for the message vector. To prove the content of the signature $(m_1, ..., m_n)$, the prover does the following: it first selects the random number $r_1, t_1 \in Z_p$ and then computes $\sigma' \leftarrow (\sigma_1^{r_1}, (\sigma_2 \cdot \sigma_1^{t_1})^{r_1})$. It sends $\sigma' = (\sigma'_1, \sigma'_2)$ to the verifier and performs a zero-knowledge interaction proof on the knowledge π and t_1 of the message vector $(m_1, ..., m_n)$. EV computes $e(\sigma'_1, \hat{X}) \cdot \prod e(\sigma'_1, \hat{Y}_j)^{m_j} \cdot e(\sigma'_1, \hat{g})^{t_1} = e(\sigma'_2, \hat{g})$. If π is valid, the verifier accepts it.

3) Mutual Authentication of Electric Vehicles and Charging Stations: After EV gets the transaction token from cloud computing center, it needs to authenticate with the charging station and then complete the transaction. EV first enters the transaction token and calculates $\alpha_{ev} = h(Token)$, then generates a random number $n_7 \in Z_p$. Then EV computes $sk_{ev} = sk'_{ev} \oplus h(Token)$ and $vm_7 = h(PID_{ev} \parallel n_7 \parallel sk_{ev} \parallel T_7)$. EV sends the message $m_7 : \{PID_{ev}, n_7, vm_7\}$ to the charging station. After receiving the message, the charging station generates a random number $n_8 \in Z_p$. EV computes $vm_8 = h(ID_{cs} \parallel n_8 \parallel k_{ce} \parallel T_8)$.

The charging station then sends the message $m_8 : \{m_7, ID_{cs}, n_8, vm_8\}$ to cloud computing center. When the message m_8 arrives, cloud computing center first locates PID_{ev} in its database, then cloud computing center verifies vm_7 and vm_8. The cloud center generates a session key SK and a new pseudonym PID_{ev}^n. Then the charging station computes $PID_{ev}^{n'} = h(PID_{ev} \parallel sk_{ev}) \oplus PID_{ev}^n$ $SK_{ev} = h(ID_{ev} \parallel sk_{ev} \parallel n_7) \oplus SK$, $SK_{cs} = h(ID_{cs} \parallel k_{ce} \parallel n_8) \oplus SK$, $vm_7 = h(SK_{cs} \parallel k_{ce} \parallel n_9)$ and $vm_8 = h(ID_{cc} \parallel sk_{ev} \parallel PID_{ev}^n)$.

Next, cloud computing center composes a message $m_9 : \{(PID_{ev}^{n'}, SK_{ev}, vm_8) \parallel (SK_{cs}, vm_7)\}$ and sends m_9 to the charging station. After receiving the message vm_7 from cloud computing center, the charging station first verifies and calculates vm_7. If the verification passes, the charging station decrypts the session key $SK = h(ID_{cs} \parallel k_{ce} \parallel n_8) \oplus SK_{cs}$ and sends a message $m_{10} : \{PID_{ev}^{n'}, SK_{ev}, vm_8\}$ to EV. When EV receives the message, it verifies vm_8 and calculates the session key $SK = h(ID_{ev} \parallel sk_{ev} \parallel n_7) \oplus SK_{ev}$. Mutual authentication process is completed.

C. Identity Update and Tracking

1) Electric Vehicle Identity Update: EV generate new pseudonyms after electricity trading is completed, EV compute $PID_{ev}^n = h(PID_{ev} \parallel sk_{ev}) \oplus PID_{ev}^{n'}$ where PID_{ev}^n is the updated pseudonym and sk_{ev} is the private key of EV. The private key is stored in the tamper-proof device of EV. A malicious EV cannot compute the updated pseudonym through the hash function without knowing EV's private key, and malicious EV cannot continue to listen to EV. Besides, cloud computing

TABLE II
RUNTIME OF CRYPTOGRAPHIC OPERATION

Notations	Descripyions	Runtime(ms)
T_{a-bp}	Symmetric bilinear pairing time	14.151
T_{a-bm}	Multiplication time in symmetric bilinear pairing	0.099
T_{a-ba}	Addition time in symmetric bilinear pairing	0.126
$T_{a-\exp}$	Exponential operation time in symmetric bilinear pairing	11.924
T_{d-bp}	Asymmetric bilinear pairing time	17.117
$T_{d-\exp}$	Exponential operation time in asymmetric bilinear pairing	4.647

TABLE III
COMPARISON OF COMPUTATIONAL COSTS

Literature Schemes	EV	CS/LAG	Total
Yang et al. [5]	$7T_{a-bp}+11T_{a-bm}+4T_{a-ba}$	$11T_{a-bp}+6T_{a-bm}+3T_{a-ba}$	$18T_{a-bp}+17T_{a-bm}+7T_{a-ba}$
Wang et al. [6]	$5T_{a-bp}+5T_{a-bm}+2T_{a-ba}$	$7T_{a-bp}+2T_{a-bm}+1T_{a-ba}$	$12T_{a-bp}+7T_{a-bm}+3T_{a-ba}$
Wang et al. [7]	$9T_{a-\exp}$	$4T_{a-bp}+6T_{a-\exp}$	$4T_{a-bp}+15T_{a-\exp}$
Our Scheme	$5T_{d-bp}+9T_{d-\exp}$	$2T_{d-\exp}$	$5T_{d-bp}+11T_{d-\exp}$

center needs to keep a copy $PID_{ev}^{n} = H_0^X(ID_{ev} \parallel VT_i')$ where VT_i' is the effective time of the new pseudonym. EV identity update is complete.

2) Electric Vehicle Identity Tracking: EV sends a message m during the authentication process that is tied to EV's group signature $\mu_{ev} = (\hat{\sigma}_{ev1}', \hat{\sigma}_{ev2}', c_{ev}, s)$. When EV communicates with edge server, edge server generates a signature μ_{es}^{ev} and cloud computing center verifies the validity of edge server signature before checking the following equation $e(\hat{\sigma}_{ev1}', \hat{g}) \cdot e(\hat{\sigma}_{ev1}', \hat{X}) = e(\hat{\sigma}_{ev2}', \hat{\tau})$ through the database entries $(i, \tau_i, \eta_i, \hat{\tau}_i)$ to track the identity of EV.

V. EVALUATION

This section evaluates the performance of the proposed scheme and analyzes the computational costs and communication costs. The experiments simulate the electricity trading process using Java 14.0.1 and evaluate the running time of the operation, and analyze the computational costs and communication costs incurred during the process. The test environment is Win10, 3.0Ghz AMD Ryzen 5 4600H, 16GRAM. The scheme uses JPBC [16] to simulate the cryptographic operation, and the symmetric bilinear pairing uses an A-type elliptic curve $y^2 = x^3 + x \bmod \bar{p}$, where $\bar{q}$ is a 160-bit prime number and $\bar{p}$ is a 512-bit prime number; the asymmetric bilinear pairing uses a D-type elliptic curve. To demonstrate the validity of the proposed scheme, this section unifies the symbols and definitions in the scheme, as shown in Table 2.

A. Computational Costs

This section evaluates the computational costs incurred by EV, charging station and local aggregator. In addition, this section compares the proposed scheme with the other three schemes Yang et al. [5], Wang et al. [6], and Wang et al. [7]. The computational costs of these schemes are first analyzed mathematically, assuming that some lightweight cryptographic operations generate negligible overheads, such as hash operations, XOR operations, and message signatures. A mathematical comparison of the computational costs generated by these two schemes is given in Table 3. The computational costs are generated by EV, charging station and local aggregator.

As shown in Table 3, for Yang et al. [5], all the computational costs are $18T_{a-bp}+17T_{a-bm}+7T_{a-ba}$. For Wang et al. [6], all computational costs are $12T_{a-bp}+7T_{a-bm}+3T_{a-ba}$. For Wang et al. [7], all computational costs are $4T_{a-bp}+15T_{a-\exp}$. The computational costs incurred by this scheme in the entity is $5T_{d-bp}+11T_{d-\exp}$, and this scheme has good performance in theory. The computational costs of EVs is shown in figures 4.

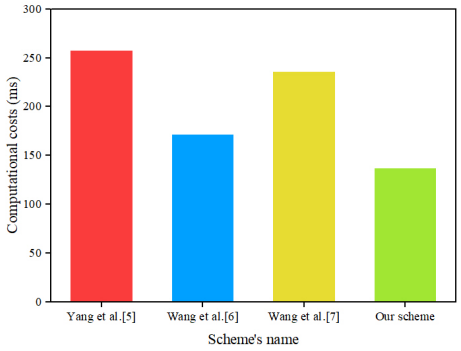

Fig. 4. Computational costs.

B. Communication Costs

The symmetric bilinear pair type uses an elliptic curve of type A $y^2 = x^3 + x$ with order 160-bit , so the size of the elements in G_1 and G_2 is 40 bit. The asymmetric bilinear pair type uses a D-type elliptic curve of order 170-bit, so the element size in G_1 and G_2 is 42.5 bit. This scheme assumes that the timestamp, hash function, and other strings are 5 bit. This scheme considers the costs incurred in communicating with EVs. The communication costs of EVs is shown in figures 5.

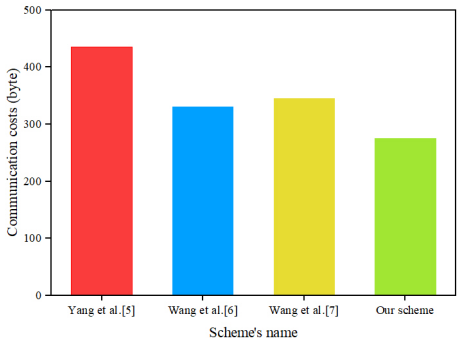

Fig. 5. Communication costs.

For the scheme Yang et al. [5], in the license generation phase, the EV sends messages $\{ID_{BV}, M, t_1, sig_{\Gamma_{BV}}(\cdot)\}$ and $\{h, t_3\}$ to the central aggregator. The central aggregator sends messages $\{z, a, b, U, Y, t_2\}$ and $\{S_1, S_2, t_4\}$ to the EV. The communication costs of Yang et al. [5] are $40 \times 10 + 5 \times 7 = 435$ bytes.

For the scheme Wang et al. [6], in the license generation phase, the EV sends messages $\{ID_{BV}, M, t_1\}$ and $\{p, t_3\}$ to the central aggregator. The central aggregator sends messages $\{R, R_1, R_1', t_2\}$ and $\{V, t_4\}$ to the EV. The communication costs of Wang et al. [6] are $40 \times 7 + 5 \times 10 = 330$ bytes.

For the scheme Wang et al. [7], in the authentication phase, the EV sends a message $\{M, T_\theta, \sigma, cert_i\}$ to the local aggregator. The local aggregator sends a message $\{c_1, c_2, c_3, c_4, time, \beta\}$ to the EV. The communication costs of Wang et al. [7] are $40 \times 8 + 5 \times 5 = 345$ bytes.

In our scheme, in the initialization phase, the EV sends a message $\{PID_{ev}, n_1, vm_1\}$ to the edge server. In the mutual authentication phase, the EV sends a message $\{PID_{ev}, n_3, vm_3\}$ to the local aggregator, and then the EV sends a message $\{em_{ev}, \hat{\sigma}'_{ev1}, \hat{\sigma}'_{ev2}, c_{ev}, s_0\}$ to the edge server. In the transaction phase, the EV sends a message $\{PID_{ev}, n_7, vm_7\}$ to the charging station, and the charging station sends a message $\{m_7, ID_{cs}, n_8, vm_8\}$ to the cloud computing center. Then the cloud computing center sends a message $\{(PID_{ev}^{n'}, SK_{ev}, vm_8) \parallel (SK_{cs}, vm_7)\}$ to the charging station, and the charging station sends a message $\{PID_{ev}^{n'}, SK_{ev}, vm_8\}$ to the EV. The communication costs of our scheme are $42.5 \times 4 + 5 \times 21 = 275$ bytes.

VI. CONCLUSION

This scheme proposes a cloud-edge collaboration authentication scheme based on V2G short randomizable signature to avoid the possibility of attacks in V2G networks. The scheme uses short randomizable signatures and hash functions to construct an anonymous authentication scheme for EV charging and discharging process. The simulation results show that the scheme has higher efficiency and practicality compared to other schemes.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China(52177067). The findings achieved herein are solely the responsibility of the authors.

REFERENCES

[1] K. Shuaib, E. Barka, J. A. Abdella, F. Sallabi, M. Abdel-Hafez, Ala Al-Fuqaha, Secure Plug-in Electric Vehicle (PEV) Charging in a Smart Grid Network, Journal Energies, v.10, 2017.
[2] R. S. Levinson and T. H. West, Impact of public electric vehicle charging infrastructure, Transp. Res. Part D Transp. Environ., vol. 64, pp. 158177, Oct. 2018, doi: 10.1016/j.trd.2017.10.006.
[3] Future electric vehicles market analysis, Accessed: Jan 27, 2021. [Online]. Available: https://www.alliedmarketresearch.com/electric-vehicle-market
[4] Plans to ban fuel powered vehicles in UK, Accessed: Jan 30,2021. [Online]. Available: https://www.downtoearth.org.in/news/climatechange/what-does-boris-johnson-s-plan-to-ban-fuel-powered-vehicles-mean-for-uk-74358
[5] Z. Yang, S. Yu, W. Lou, and C. Liu, P2: Privacy-preserving communication and precise reward architecture for V2G networks in smart grid, IEEE Trans. Smart Grid, vol. 2, no. 4, pp. 697706, Dec. 2011.
[6] H. Wang, B. Qin, Q. Wu, L. Xu, and J. Domingo-Ferrer, TPP: Traceable privacy-preserving communication and precise reward for vehicle-to-grid networks in smart grids, IEEE Trans. Inf. Forensics Secur., vol. 10, no. 11, pp. 23402351, Nov. 2015.
[7] Q. Wang, M. Ou, Y. Yang and Z. Duan, "Conditional Privacy-Preserving Anonymous Authentication Scheme With Forward Security in Vehicle-to-Grid Networks," IEEE Access,vol. 8, pp. 217592-217602, 2020, doi: 10.1109/ACCESS.2020.3040112.
[8] X. Hu and S. Huang, Analysis of ID-based restrictive partially blind signatures and applications, J. Syst. Softw., vol. 81, no. 11, pp. 19511954, Nov. 2008.
[9] F. Kupzog, H. J. Bacher, M. Glatz, W. Prüggler, A. Adegbite, and G. Kienesberger, Architectural options for vehicle to grid communication, e i Elektrotechnik und Informationstechnik, vol. 128, no. 1-2, pp. 4752,2011.
[10] H. Liu, H. Ning, Y. Zhang, and L. T. Yang, Aggregated-proof based privacy-preserving authentication for V2G networks in the smart grid, IEEE Trans. Smart Grid, vol. 3, no. 4, pp. 17221733, Dec. 2012.
[11] H. Liu, H. Ning, Y. Zhang, and M. Guizani, Battery status-aware authentication scheme for V2G networks in smart grid, IEEE Trans. Smart Grid, vol. 4, no. 1, pp. 99110, Mar. 2013.
[12] D. Pointcheval, O. Sanders, Short randomizable signatures, in: Proc. Cryptographers Track at the RSA Conference, CT-RSA, San Francisco, USA, 2016, pp. 111126.
[13] Camenisch J, Lysyanskaya A Signature schemes and anonymous credentials from bilinear maps[C]//Annual international cryptology conference. Springer, Berlin, Heidelberg, 2004: 56-72.
[14] C.P. Schnorr, Efficient identification and signatures for smart cards, in: Proc. 9th Annual International Cryptology Conference, CRYPTO, Santa Barbara, USA, 1989, pp. 239252.
[15] Damgård I, Pastro V, Smart N, et al. Multiparty computation from somewhat homomorphic encryption[C]//Annual Cryptology Conference. Springer, Berlin, Heidelberg, 2012: 643-662.
[16] The java pairing based cryptography library (JPBC). [Online]. Available: http:// gas.dia.unisa.it/projects/jpbc/, Accessed on: Dec. 10, 2018.
[17] He D, Zeadally S, Xu B, et al. An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks[J]. IEEE Transactions on Information Forensics and Security, 2015, 10(12): 2681-2691.