PAP: A Privacy-Preserving Authentication Scheme with Anonymous Payment for V2G Networks

Xiaohan Yue, Xue Bi, Haibo Yang, Shi Bai, and Yuan He

Abstract-- Vehicle-to-grid (V2G) networks, as an emerging smart grid paradigm, can be integrated with renewable energy resources to provide power services and manage electricity demands. When accessing electricity services, an electric vehicle($\mathcal{EV}$) typically provides authentication or/and payment information containing identifying data to a service provider, which raises privacy concerns as malicious entities might trace $\mathcal{EV}$ activity or exploit personal information. Although numerous anonymous authentication and payment schemes have been presented for V2G networks, no such privacy-preserving scheme supports authentication and payment simultaneously. Therefore, this paper is the first to present a privacy-preserving authentication scheme with anonymous payment for V2G networks (PAP, for short). In addition, this scheme also supports *accountability* and *revocability*, which are practical features to prevent malicious behavior; *minimal attribute disclosure*, which maximizes the privacy of $\mathcal{EV}$ when responding to the service provider's flexible access policies; *payment binding*, which guarantees the accountability in the payment phase; *user-controlled linkability*, which enables $\mathcal{EV}$ to decide whether different authentication sessions are linkable for continuous services. On the performance side, we implement PAP with the pairing cryptography library, then evaluate it on different hardware platforms, showing that it is essential for V2G applications.

Index Terms--**Authentication, payment, V2G, anonymous credential, revocability, accountability.**

I. Introduction

Vehicle-to-Grid (V2G), as the key technology of a smart grid[1]-[4], plays an essential role in balancing the energy load and tracking power demand, which can achieve bidirectional current flow and information exchange between electric vehicles ($\mathcal{EV}$s) and the grid. However, since the messages transmitted often contain $\mathcal{EV}$s' sensitive data, such as identity, location, charging/discharging, and payment information, etc, which can bring significant challenges to ensuring data security and privacy.

Currently, privacy-preserving/anonymous authentication techniques for V2G networks mainly rely on two approaches. The first is pseudonymous identity[5]-[9], which means an $\mathcal{EV}$ uses different pseudonym IDs instead of its real ID in each authentication process, and only the trusted authority (TA) can reveal the real identity of the $\mathcal{EV}$ in pseudonym. However, when using pseudonyms, the slight correlation of data may expose the user's behavior and habits. In addition, TA generating pseudonym IDs must participate in each authentication process, which imposes extra computation and communication burden on the TA. On the contrary, another approach is ZKP(zero-knowledge proof)-based anonymous authentication[10]-[12] that $\mathcal{EV}$ can generate a proof to prove the possession of required credentials in zero knowledge, which avoid the presence of TA in the authentication process. However, the existing anonymous authentication schemes for V2G networks are strawman construction as the below challenges in practice:

(1) How to guarantee the anonymity of the $\mathcal{EV}$ during the payment process to service providers while keeping accountability?

(2) How to address the issue of supporting flexible access policies made by service providers in V2G networks, such as charging stations ($\mathcal{CS}$), and linkability controlled by $\mathcal{EV}$ for uninterrupted services?

(3) How to ensure accountability and revocability for malicious $\mathcal{EV}$s?

Unfortunately, to the best of our knowledge, there is no such existing cryptographic primitive or scheme that can overcome all the challenges. As a result, our proposal is the first to complete authentication and payment anonymously in V2G while featuring minimal attribute disclosure, accountability, user-controlled linkability, and payment binding.

In the case of *challenge (1)*, due to the complexity of existing decentralized anonymous payment(DAP) systems, such as Zcash[13], Hawk[14], and Zexe[15], it will cause high overhead for $\mathcal{EV}$s with limited resources (as spenders). Another issue is that these systems focus on anonymity and ignore authentication between entities, which implies these DAP schemes without authentication cannot offer accountability. However, the anonymity of anonymous authentication creates another regulatory challenge for payment: how to establish a binding relationship between anonymous authentication information and payment information of the same $\mathcal{EV}$. To tackle the challenges, in PAP, beyond anonymous authentication, we employ the blockchain with practical byzantine fault tolerance (PBFT) consensus[16] and one-time signature (OTS)[17] to design a lightweight anonymous payment method with payment binding, while maintaining accountability.

For *challenge (2)*, in practice, the access policies of service providers can be complex and varied. In any case, the service provider always expects to be able to determine whether the $\mathcal{EV}$ owns the set of attributes that satisfies the access policy. For

Xiaohan Yue, Xue Bi, Haibo Yang, and Shi Bai are with the School of Information Science and Engineering, Shenyang University of Technology, Shenyang 110870, Liaoning, China (e-mail: xhyue@sut.edu.cn, xueb@smail.sut.edu.cn, yanghb@sut.edu.cn, baishi@sut.edu.cn).

Yuan He is with the Graduate School of Science and Technology, Keio University, Yokohama, Kanagawa 223-8522, Japan (e-mail: isaacyhe@acm.org).

this situation, the attribute-based anonymous credential[18] scheme is a desirable method in the V2G networks. However, the traditional attribute-based anonymous credential scheme requires $\mathcal{EV}$s to present all attributes in the authentication phase, which incurs additional overhead and privacy attacks (including differencing attacks, linkage attacks, and reconstruction attacks)[19]. Hence, utilizing redactable attribute-based credentials to show an attribute subset according to the access policy of the verifier (i.e., service provider) will effectively preserve the user's privacy. For example, an $\mathcal{EV}$ obtains an access credential from an issuer ($\mathcal{Iss}$) on the attribute set $Attr$ ={"services = charging/discharging," "year = 2023," "vehicle types = Tesla," "locations = Seattle," "license plate number=5QWE678"}. Then, the access policy $\Gamma_{Attr'}$ of the attribute set $Attr'$ = {"services = charging/discharging," "vehicle types=Tesla"} is set by the $\mathcal{CS}$. When accessing V2G services, the $\mathcal{EV}$ simply proves that its credentials contain the required $Attr'$ without showing unnecessary attributes. Based on meeting the access policy, the redactable method can achieve the minimum attribute disclosure (Min-AD) of $\mathcal{EV}$s.

Due to anonymity, it is impossible for the service provider (SP) to establish correlation between anonymous sessions initiated by the same $\mathcal{EV}$, thereby preventing continuous services in the event of communication interruption, such as charging or discharging service. To address the tension between privacy and utility, we adopt a user-controlled way, i.e., basename-based way[20], to make different sessions or transactions linkable, which means all authentication information with the same basename can easily be linked by SP, but not vice versa.

As for *challenge (3)*, to ensure accountability and revocability for malicious $\mathcal{EV}$s, the registration center ($\mathcal{RC}$) needs to adopt a proper membership management mechanism in V2G, especially revocation for illegal members. Currently, several methods can achieve this point, including revocation list[21], dynamic accumulator[22], [23], and NNL[24], [25]. Among these methods, although the dynamic accumulator offers a membership update algorithm with computation complexity $O(r)$(where r is the number of revoked $\mathcal{EV}$s), it is history-dependent: requiring $\mathcal{EV}$ to update its membership witness by itself every time the accumulator value changes. Thus, this method is unsuitable for $\mathcal{EV}$ with limited computing resources. To reduce the computation overhead of $\mathcal{EV}$, we employ the complete subtree (CS) method in NNL[24] to build a novel revocation mechanism for membership which is history-independent and more efficient than that based on the revocation list.

Contributions. Following the above ideas, our contributions are as below:

●**A new privacy framework for V2G networks.** We introduce PAP, the first lightweight privacy-preserving authentication framework with anonymous payment in the V2G networks. We further describe the functionality and security requirements the PAP should meet and present the responsibilities of each entity and the formal definitions of algorithms in PAP.

●**A construction of PAP**. With the linear homomorphic signature, PBFT consensus mechanism, membership management mechanism based on the CS method, and the ZKP system, we present a construction of PAP captures:

‣ *Anonymous authentication with Min-AD and UCL.* Beyond anonymity in the authentication phase, we employ linear homomorphic signatures to implement aggregation of credentials and redaction of attributes for matching access policies with Min-AD. For long-duration services in V2G, e.g., charging services, our proposal offers UCL to guarantee service continuity even if the session is interrupted.

‣ *Lightweight anonymous payment with payment binding.* By the PBFT consensus mechanism and one-time signature, we design a lightweight anonymous payment method after authentication, which supports payment binding to keep accountability, and resists the double-spending attack.

‣ *History-independent* membership management *mechanism.* We present a novel membership management mechanism including accountability, and revocation method based on NNL framework[24], [25].

●**A new trade-off between privacy and efficiency.** We provide an evaluation that shows PAP is practical and can handle transactions anonymously, which implies that PAP offers a new, practically relevant balance between privacy and efficiency compared to all previous approaches in V2G.

II. Building Blocks

A. Redactable Signatures

Our approach to *challenge (2)* is to utilize redactable signatures[26] with linear message homomorphism, which provides *redactability* and *rerandomizability* of signatures and can be redacted to remain authentic on a subset of n_s messages of the message set $\{m_i\}_{i=1}^{n}$. As the number $(n - n_s)$ of undisclosed messages does not affect the complexity of communication, one can efficiently perform partial verification of the signature, redact message-signature pairs, randomize the origin signatures and reveal only their relevant parts each time they are used. It's easy to see that *redactability* implies Min-AD, and *rerandomizability* offers unlinkability between the original signature and the randomized signature.

However, a redactable signature is not friendly to parallelly aggregate multiple credentials from different issuers. Thus, we adopt a serial approach to aggregate multiple credentials to reduce cryptographic operations for $\mathcal{EV}$s, detailed in Sec IV.D.

B. PBFT Consensus Mechanism

To optimize anonymous payment, we apply the PBFT consensus mechanism[16] in our construction ensuring that it can reach consensus correctly with one-third fault tolerance while satisfying the liveness and security of the distributed system.

Note that for convenience, in our setting we set the committee ($\mathcal{Cmt}$) to invoke PBFT and refer to the members of $\mathcal{Cmt}$ as commissioners. If the system approves a block through the PBFT, it is final and will not be revoked, and there is no need to wait for confirmation to ensure that the current block is in the longest chain because each node reaches a consensus simultaneously. Therefore, the computational cost in the payment verification process can be greatly reduced.

C. Membership Management Mechanism

As for *challenge (3)*, we use the complete subtree (CS) method to ensure revocability for the malicious based on accountability as shown in Figure 1, when an $\mathcal{EV}_u$ enrolls our system, $\mathcal{RC}$ assigns it a leaf node n_ℓ and a path ρ_u from that leaf node to the root node of a complete binary tree T_ℓ of ℓ-depth. Based on the path ρ_u, $\mathcal{RC}$ generates accountability tokens and a revocation token for $\mathcal{EV}_u$.

When a new member enters the system, the system center puts it into a leaf node and distributes the node path. $\mathcal{RC}$ also maintains a set $S_R^{(t)}$, including root nodes of unrevoked subtrees. Hence, judging whether a member is revoked is only to confirm the intersection node between its path and $S_R^{(t)}$. For example, Figure 1 (a) is a binary tree with $S_R^{(t)} = \{S_1 = n_0\}$ and the path of n_9 is $\rho_{n_9} = (n_0, n_1, n_4, n_9)$. Then, revoking n_9 as shown in (b) also subsequently revoked all the nodes contained in this path. Afterward, the updated tree contains three subtrees and the root node set becomes $S_R^{(t+1)} = \{S_1 = n_3, S_2 = n_{10}, S_3 = n_2\}$, and at this time $\rho_{n_9} \cap S_R^{(t+1)} = \emptyset$, which means that n_9 is an illegal leaf node, that is to say, this member has been revoked. Similarly, the path of n_{10} is $\rho_{n_{10}} = (n_0, n_1, n_4, n_{10})$, and then $\rho_{n_{10}} \cap S_R^{(t+1)} \neq \emptyset$ means n_{10} is still a legal member.

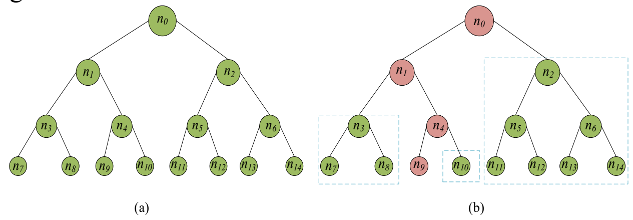


Figure 1. The process of revoking a member. (a) is the initial T_ℓ; (b) is the process of revoking n_9.

We leverage this method to revoke illegal $\mathcal{EV}$s within our scheme. In the initial phase, the $\mathcal{RC}$ issues the revocation token on an $\mathcal{EV}$'s path nodes in revocation epoch t. Then, the unrevoked $\mathcal{EV}$ can prove that its path and the set $S_R^{(t)}$ have an intersection node in epoch t. Further, the status of the complete subtree needs to be updated in the next epoch $t+1$. Since the number of $S_R^{(t)}$ is less than $r_{\mathcal{EV}} \cdot \log \frac{N_{\mathcal{EV}}}{r_{\mathcal{EV}}}$, the computation cost of the update algorithm is quasi-linear, i.e., $O(r_{\mathcal{EV}} \cdot \log \frac{N_{\mathcal{EV}}}{r_{\mathcal{EV}}})$ where $N_{\mathcal{EV}}$ is the maximum number of $\mathcal{EV}$s, $r_{\mathcal{EV}}$ is the number of revoked $\mathcal{EV}$s.

D. Zero-knowledge argument system

In a zero-knowledge argument, a prover needs to convince a verifier that the statement is true without the verifier learning anything except the validity of the statement.

For $(x, w) \in R$, x is statement, w is witness, R is the relationship between x and w, a non-interactive zero-knowledge argument system in the random oracle model[32] for R consists of three PPT algorithms$(\mathrm{Setup}, \mathcal{P}^{\mathrm{H}}, \mathcal{V}^{\mathrm{H}})$, if there exists a PPT simulator $\mathcal{S}$ that can operate two oracles $\mathcal{S}_1, \mathcal{S}_2$ such that for all PPT distinguisher $\mathcal{D}$, its advantage is as below.

We define $\mathcal{P}^{\mathrm{H}}, \mathcal{V}^{\mathrm{H}}$ to denote the prover and verifier only have oracle access to H. $\mathcal{S}_1$ can return the random oracle calls to H_i on input and $\mathcal{S}_2$ is a proof simulation oracle.

$$\begin{aligned}\mathrm{Adv}(\mathcal{D}, \mathcal{S}) = |\Pr\left[\mathcal{D}^{H, \mathcal{P}_{\mathrm{FS}}^{\mathrm{H}}}(pp) = 1 : (pp, \mathrm{H}) \leftarrow \mathrm{Setup}(1^\lambda)\right] \\ - \Pr\left[\mathcal{D}^{\mathcal{S}_1, \mathcal{S}_2}(pp) = 1 : (pp, \mathrm{H}) \\ \leftarrow \mathrm{Setup}(1^\lambda)\right]|\end{aligned}$$

E. One-time signature(OTS)

A one-time signature[17] (OTS) scheme is a digital signature scheme that can be used to sign one message per key pair. Specifically, the public key can be used to verify the validity of the signature, while the private key can only be used during the signature process and should be kept strictly confidential. And the significant feature of OTS is that each signature can be used only once. After verifying, the signature becomes invalid and cannot be used again. This mechanism provides greater security because even if an adversary intercepts one signature, she cannot use it to forge valid signatures for other messages.

F. Preliminaries in Cryptography

(1) Bilinear pairing:

Given cyclic groups, $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_\mathrm{T}$ of prime order p and g_1, g_2 are the generators of the group $\mathbb{G}_1$, $\mathbb{G}_2$, respectively. Type-3 pairing groups [55] are chosen in this paper due to its asymmetry so that DDH assumption can be satisfied.

Bilinear pair $e: \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_\mathrm{T}$ is a mapping that satisfies the following properties:

- *Bilinearity*: For all $a, b \in \mathbb{Z}_\mathrm{p}$, $g_1 \in \mathbb{G}_1$, $g_2 \in \mathbb{G}_2$, $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$.
- *Non-degeneracy*: For all $e(g_1, g_2) \neq 1_T$.
- *Computability*: e is efficiently computable.

(2) Complexity Assumption:

Definition 1 Discrete Logarithm (DL) Assumption[29]: Let $\mathbb{G}$ be a cyclic group of prime order p, given (g, g^x), so it is difficult to recover x in the calculation for any generator g.

Definition 2 Decisional Diffie-Hellman (DDH) Assumption [57]: Let $\mathbb{G}$ be a cyclic group of prime order p where the generator is g. Given (g, g^x, g^y, g^z) and it is hard to decide whether $z = x \cdot y$ or z is random.

Definition 3 Pointcheval-Sanders (PS) Assumption[30]: Let $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, e)$ be a bilinear group while g_1, g_2 are the generators of $\mathbb{G}_1$, $\mathbb{G}_2$, respectively. Given (g_2, g_2^x, g_2^y) with $x, y \in \mathbb{Z}_p$ and unlimited access to an oracle where input $m \in \mathbb{Z}_p$, randomly choosing $a \in \mathbb{G}_1$ and outputting a tuple (a, a^{x+my}) so that no PPT adversary can efficiently generate such a new tuple for m^* is not queried.

Definition 4 q-Strong Diffie-Hellman (q-SDH) Assumption[31]: Let $g_1, g_2, g_2^a, g_2^{a^2}, \cdots, g_2^{a^q}$ as input, where g_2 is the generator of $\mathbb{G}_2$, $g_1 = \varphi(g_2)$, the q-SDH problem is to compute a pair $(c, g^{\frac{1}{a+c}}) \in \mathbb{Z}_p \times \mathbb{G}$ for a freely chosen integer c in $\mathbb{Z}_p$.

Definition 5 External Diffie-Hellman (XDH) Assumption[20]: Given an asymmetric bilinear pairing: $e: \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_\mathrm{T}$, if DDH assumption cannot be solved in $\mathbb{G}_1$, then XDH assumption can hold.

Definition 6 q-Modified Strong Diffie-Hellman (q-MSDH-1) Assumption[56]: Let $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, e)$ be a bilinear group for type-3 while g_1, g_2 are the generators of $\mathbb{G}_1$, $\mathbb{G}_2$, respectively. Given $\left\{\left(g_1^{x^i}, g_2^{x^i}\right)\right\}_{i=0}^{q}$ along with $(g_1^{a}, g_2^{a}, g_2^{ax})$, for $a, x \leftarrow \mathbb{Z}_p$, no PPT adversary can output a tuple $(w, P, h^{\frac{1}{x+w}}, h^{\frac{a}{P(x)}})$ for some $h \in \mathbb{G}_1$, where P is a polynomial of degree at most q and w is a scalar such that $(X + w)$ and $P(X)$ are relatively prime.

III. Definitions of Framework and Security Requirements

In this section, we briefly define the framework and security requirements of PAP.

A. *The Framework of PAP*

To formally define the PAP framework for V2G, we first present the responsibility of each party in Figure 2 as follows.

● **Registration center ($\mathcal{RC}$)** is responsible for generating global public parameters and accepting registration requests in V2G networks and maintaining a complete binary tree used by membership management mechanism.

● **Credentials issuer ($\mathcal{Iss}$)**, which can be the service provider, is responsible for issuing attributed-based credentials for $\mathcal{EV}$s.

● **Charging station ($\mathcal{CS}$)** can check whether an $\mathcal{EV}$ has permission to enjoy charging, discharging or other services. When a paid service completes, $\mathcal{CS}$ generates the corresponding payment information and makes transactions with $\mathcal{EV}$.

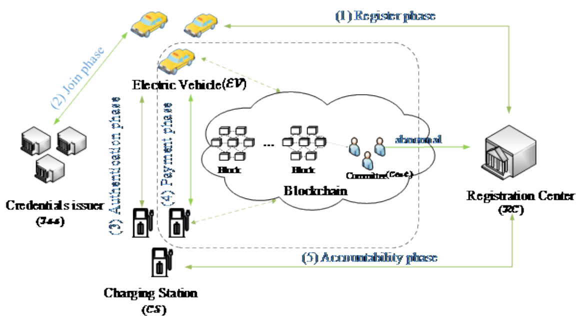

(1) Register phase
(2) Join phase
Electric Vehicle($\mathcal{EV}$)
(3) Authentication phase
(4) Payment phase
Block
Block
Committee($\mathcal{Cmt}$)
abnormal
Blockchain
Credentials issuer ($\mathcal{Iss}$)
Registration Center ($\mathcal{RC}$)
(5) Accountability phase
Charging Station ($\mathcal{CS}$)

Figure 2. The framework of PAP

● **Electric vehicle ($\mathcal{EV}$)**, the end entity, is equipped with an onboard computer (OBC) to performs complex computation tasks, such as authentication or payment.

● **Committee ($\mathcal{Cmt}$)**, the trusted party, is responsible for maintaining a blockchain, verifying the validity of transactions between the $\mathcal{EV}$ and the $\mathcal{CS}$ in the payment phase.

TABLE I

Notations Used in Our Scheme

$(pk_{\mathcal{RC}}, sk_{\mathcal{RC}})$	Public-private keys of $\mathcal{RC}$
$(apk_{\mathcal{RC}}, ask_{\mathcal{RC}})$	Public-private keys of $\mathcal{RC}$ for accountability
$(pk_{\mathcal{Iss}_k}, sk_{\mathcal{Iss}_k})$	Public-private keys of $\mathcal{Iss}_k$ for issuing credentials
$(pk_{\mathcal{EV}_u}, sk_{\mathcal{EV}_u})$	Public-private keys of $\mathcal{EV}_u$
$(pk_{\mathcal{EV}_u}^{\text{addr}}, sk_{\mathcal{EV}_u}^{\text{addr}})$	Wallet address key pair of $\mathcal{EV}_u$
$\left(\widetilde{pk_{\mathcal{EV}_u}^{\text{addr}}}, \widetilde{sk_{\mathcal{EV}_u}^{\text{addr}}}\right)$	Pseudo-address keys derived from $(pk_{\mathcal{EV}_u}^{\text{addr}}, sk_{\mathcal{EV}_u}^{\text{addr}})$
$(pk_{\mathcal{CS}}, sk_{\mathcal{CS}})$	Address key pair of the $\mathcal{CS}$ for transacting
$(pk_{\mathcal{Cmt}}, sk_{\mathcal{Cmt}})$	Address key pair of $\mathcal{Cmt}$
tk_u^{acc}	The accountability token issued to $\mathcal{EV}_u$ by $\mathcal{RC}$
tk_u^{rev}	The revocation token issued to $\mathcal{EV}_u$ by $\mathcal{RC}$
$\{\mathfrak{a}_i\}$	The attribute set of an $\mathcal{EV}$
$\widetilde{cred_u}$	The redacted credential presented to the $\mathcal{CS}$ by $\mathcal{EV}_u$
ρ_u	The $\mathcal{EV}_u$'s path of a complete binary tree
Σ	EdDSA[27] scheme, i.e., $\Sigma := (\text{Setup}, \text{Sign}, \text{Verify})$
L_t	The list of $\mathcal{EV}$s need to be revoked at epoch t
$\hat{\sigma}_u^{(k)}$	Aggregated signature signed by $\mathcal{Iss}_k$

Considering the practical functionality and security requirements in V2G, the following presents the formal definitions of our scheme. The symbols used in our scheme are shown in TABLE I.

(1) Initialization phase:

This phase features several algorithms generating global public parameters and public-private keys. The definitions are as follows:

▪$\text{Setup}(1^\lambda) \to pp$: on the input of the security parameter λ, outputs the global public parameters pp.

▪$\text{GenK}_{\mathcal{RC}}(pp) \to (pk_{\mathcal{RC}}, sk_{\mathcal{RC}}, apk_{\mathcal{RC}}, ask_{\mathcal{RC}})$: is run by $\mathcal{RC}$ to create key pairs $(pk_{\mathcal{RC}}, sk_{\mathcal{RC}})$ and $(apk_{\mathcal{RC}}, ask_{\mathcal{RC}})$ from pp.

▪$\text{GenK}_{\mathcal{CS}}(pp) \to (pk_{\mathcal{CS}}, sk_{\mathcal{CS}})$: is run by each $\mathcal{CS}$ to generate the key pair $(pk_{\mathcal{CS}}, sk_{\mathcal{CS}})$.

▪$\text{GenK}_{\mathcal{Iss}}(pp) \to$: is run by $\mathcal{Iss}_k$ to generate its key pair $(pk_{\mathcal{Iss}_k}, sk_{\mathcal{Iss}_k})$.

▪$\text{GenK}_{\mathcal{EV}}(pp)$: is run by each $\mathcal{EV}$ who wants to enjoy V2G services to generate its key pair $(pk_{\mathcal{EV}_u}, sk_{\mathcal{EV}_u})$ from pp.

▪$\text{GenK}_{\mathcal{Cmt}}(pp)$: is run by $\mathcal{Cmt}$ to generate its key pair $(pk_{\mathcal{Cmt}}, sk_{\mathcal{Cmt}})$.

(2) Registration phase:

In this phase, $\mathcal{EV}$ interacts with $\mathcal{RC}$ to obtain tokens including a revocation token and accountability tokens.

▪$\text{Reg}_{\mathcal{EV}}(pp, sk_{\mathcal{EV}_u}, pk_{\mathcal{EV}_u}, pk_{\mathcal{RC}}) \to \pi_u^{\text{tok}}$: is run by $\mathcal{EV}$ to create a proof π_u^{tok} of knowing the private key $sk_{\mathcal{EV}_u}$ w.r.t. $pk_{\mathcal{EV}_u}$.

▪$\text{Reg}_{\mathcal{RC}}(pp, sk_{\mathcal{RC}}, t, pk_{\mathcal{EV}_u}, \pi_u^{\text{tok}}, ID_u) \to (tk_u, \rho_u)$: is run by $\mathcal{RC}$ to verify the proof π_u^{tok}, assign a path ρ_u and compute the token $tk_u := (tk_u^{\text{acc}}, tk_u^{\text{rev}})$ from its private key $sk_{\mathcal{RC}}$, then restore $(tk_u^{\text{acc}}, \rho_u, ID_u, pk_{\mathcal{EV}_u})$ to its local database db_{Reg}, where ID_u is the real identity of $\mathcal{EV}_u$.

▪$\text{Verify}_{\mathcal{EV}}^{\text{tok}}(pp, pk_{\mathcal{RC}}, pk_{\mathcal{EV}_u}, tk_u^{\text{acc}}, tk_u^{\text{rev}}, \rho_u, t) \to 1/0$: is run by $\mathcal{EV}_u$ to verify the validity of the received tokens $(tk_u^{\text{acc}}, tk_u^{\text{rev}})$ using the public keys $pk_{\mathcal{RC}}$ and $pk_{\mathcal{EV}_u}$.

(3) Issue phase:

In this phase, each $\mathcal{EV}$ sequentially interacts with issuers to get aggregated attribute-based credentials.

▪$\text{Issue}_{\mathcal{EV}}(pp, sk_{\mathcal{EV}_u}, pk_{\mathcal{Iss}_k}) \to (\pi_u^{\text{cred}}, \hat{\sigma}_u^{(k-1)})$: is run by $\mathcal{EV}$ to generate a proof π_u^{cred} of knowing the private key $sk_{\mathcal{EV}_u}$ w.r.t. $pk_{\mathcal{EV}_u}$ and forward $(\pi_u^{\text{cred}}, \hat{\sigma}_u^{(k-1)})$ to $\mathcal{Iss}_k$.

▪ $\text{Issue}_{\mathcal{Iss}}(pp, pk_{\mathcal{EV}_u}, sk_{\mathcal{Iss}_k}, \{\mathfrak{a}_{k,i}\}, \hat{\sigma}_u^{(k-1)}, \pi_u^{\text{cred}}) \to \hat{\sigma}_u^{(k)}$: is run by $\mathcal{Iss}_k$ to check the validity of the proof π_u^{cred} and generate the aggregated signature $\hat{\sigma}_u^{(k)}$ embedding the attribute set $\{\mathfrak{a}_{k,i}\}_{i \in [N_{a,k}]}$ and containing $\hat{\sigma}_u^{(k-1)}$, where $N_{a,k}$ denotes the number of attributes that $\mathcal{Iss}_k$ can issue.

▪ $\text{Verify}_{\mathcal{EV}}^{\text{cred}}\left(pp, \left\{pk_{\mathcal{Iss}_k}\right\}_{k\in[N_\mathcal{J}]}, sk_{\mathcal{EV}_u}, \hat{\sigma}_u^{(N_\mathcal{J})}\right) \to cred_u/\perp$: is run by $\mathcal{EV}_u$ to verify the validity of the received signature $\hat{\sigma}_u^{(N_\mathcal{J})}$ $cred_u$using public keys $\{pk_{\mathcal{Iss}_k}\}_{k\in[N_\mathcal{J}]}$ and its private key $sk_{\mathcal{EV}_u}$, where $N_\mathcal{J}$ is the number of issuers. If valid, output the aggregated credential $cred_u$, else $\perp$.

(4) Authentication phase:

This phase aims to verify whether attributes held by a valid $\mathcal{EV}$ satisfy the access policy.

▪$\text{Auth}_{\mathcal{EV}}\begin{pmatrix} pp, sk_{\mathcal{EV}_u}, tk_u, cred_u, \{\mathfrak{a}_i\}_{i\in I_\Gamma}, \\ \Gamma, \{pk_{\mathcal{Iss}_k}\}_{k\in[N_\mathcal{J}]}, apk_{\mathcal{RC}}, bsn \end{pmatrix} \to (\pi_u^{\text{auth}}, \widehat{cred}_u, (sk_{ots}, vk_{ots}))$: taking as input public keys set $\{pk_{\mathcal{Iss}_k}\}_{k\in[N_\mathcal{J}]}$, $\mathcal{RC}$'s public key for accountability $apk_{\mathcal{RC}}$, a private key $sk_{\mathcal{EV}_u}$, the credential $cred_u$, the attribute set $\{\mathfrak{a}_i\}_{i\in I_\Gamma}$ of the access policy Γ, the basename bsn of $\mathcal{CS}$ and tk_u, $\mathcal{EV}_u$ outputs the redacted credential $\widehat{cred}_u$, a zero-knowledge proof π_u^{auth} and the temporary key pair of one-time signature (sk_{ots}, vk_{ots}).

▪$\text{Verify}_{\mathcal{CS}}^{\text{auth}}\begin{pmatrix} pp, \{pk_{\mathcal{Iss}_k}\}_{k\in[N_\mathcal{J}]}, \pi_u^{\text{auth}}, \\ pk_{\mathcal{RC}}, \widehat{cred}_u, \{\mathfrak{a}_i\}_{i\in I_\Gamma}, t \end{pmatrix} \to 1/0$: taking as input the public keys $\{pk_{\mathcal{Iss}_k}\}_{k\in[N_\mathcal{J}]}$, the public key of $\mathcal{RC}$, a redacted credential $\widehat{cred}_u$, a proof π_u^{auth}, the attribute set $\{\mathfrak{a}_i\}_{i\in I_\Gamma}$ and epoch t, it outputs 1 if $\widehat{cred}_u$ and π_u^{auth} are both valid and 0 otherwise.

(5) Payment phase:

If the service involves transactions, such as charging and discharging services, then the below algorithms are carried out by the $\mathcal{EV}$ and the service provider $\mathcal{CS}$, respectively.

▪$\text{Gen}_{\mathcal{EV}_u}^{\text{addr}}\left(pp, pk_{\mathcal{EV}_u}^{\text{addr}}, sk_{\mathcal{EV}_u}^{\text{addr}}, pk_{\mathcal{Cmt}}\right) \to \left(\widetilde{pk_{\mathcal{EV}_u}^{\text{addr}}}, \widetilde{sk_{\mathcal{EV}_u}^{\text{addr}}}\right)$: taking as input the key pair $(pk_{\mathcal{EV}_u}^{\text{addr}}, sk_{\mathcal{EV}_u}^{\text{addr}})$ associated with the wallet address, and $\mathcal{Cmt}$'s public key $pk_{\mathcal{Cmt}}$, $\mathcal{EV}_u$ outputs a key pair $\left(\widetilde{pk_{\mathcal{EV}_u}^{\text{addr}}}, \widetilde{sk_{\mathcal{EV}_u}^{\text{addr}}}\right)$ associated with pseudonymous address.

▪$\text{Trans}_{\mathcal{S}}\begin{pmatrix} pp, pk_{\mathcal{S}}, sk_{\mathcal{S}}, pk_{\mathcal{R}}, sk_{ots} \\ pk_{\mathcal{Cmt}}, \text{info}_{pay}, \pi_u^{\text{auth}} \end{pmatrix} \to (tx, \breve{\sigma})$: taking as input the key pair $(pk_{\mathcal{S}}, sk_{\mathcal{S}})$ of the spender, the recipient's public key $pk_{\mathcal{R}}$, a temporary signing key sk_{ots} for one-time signature, $\mathcal{Cmt}$'s public key $pk_{\mathcal{Cmt}}$, and the payment information info_{pay}, the spender outputs a transaction tx and signature $\breve{\sigma}$ on tx.

▪$\text{Verify}_{\mathcal{Cmt}}^{\text{tx}}(pp, sk_{\mathcal{Cmt}}, vk_{ots}, tx, \breve{\sigma}) \to 1/0$: taking as input a transaction tx, and $\mathcal{Cmt}$'s private key $sk_{\mathcal{Cmt}}$, the verification key vk_{ots}, $\mathcal{Cmt}$ outputs 1, if tx and $\breve{\sigma}$ are valid and 0 otherwise.

(6) Accountability phase:

In a malicious incident, $\mathcal{RC}$ can trace malicious $\mathcal{EV}$s and hold them accountable.

▪ $\text{Acc}_{\mathcal{RC}}\left(pp, \pi_u^{\text{auth}}, ask_{\mathcal{RC}}\right) \to tk_u^{\text{acc}*}$: taking as input an accountability secret key $ask_{\mathcal{RC}}$ and the zero-knowledge proof of possession of tokens and credentials π_u^{auth}, $\mathcal{RC}$ outputs $\mathcal{EV}_u$'s accountability token $tk_u^{\text{acc}*}$.

With the accountability token, $\mathcal{RC}$ can retrieve the corresponding $\mathcal{EV}_u$'s real identity ID_u and other information from db_{Reg}, such as the path ρ_u. After that, $\mathcal{RC}$ can optionally performs the revocation operation as below if necessary:

▪$\text{Revoke}_{\mathcal{RC}}\left(\{\rho_u\}_{u\in[L_t]}, \text{T}_\ell, t\right) \to S_R^{(t)}$: on the input of the revocation path set $\{\rho_u\}_{u\in[L_t]}$ that L_t is the list of revoked $\mathcal{EV}_u$, the complete binary tree T_ℓ, and current epoch t, $\mathcal{RC}$ outputs an updated node set $S_R^{(t)}$.

▪$\text{Update}_{\mathcal{RC}}\left(pp, sk_{\mathcal{RC}}, S_R^{(t)}, t\right) \to \{tk_n^{\text{rev}}\}_{n\in S_R^{(t)}}$: on input the secret key $sk_{\mathcal{RC}}$ and node set $S_R^{(t)}$ at current epoch t, $\mathcal{RC}$ outputs the updated revocation token set $\{tk_n^{\text{rev}}\}_{n\in S_R^{(t)}}$.

B. Security and Privacy Requirements

A PAP scheme for V2G networks should fulfill the following required security and privacy properties:

Anonymous Authentication: In the authentication phase, the identifying information of an $\mathcal{EV}$ should be kept private so that an adversary cannot determine its real identity according to the authentication information presented. If necessary, only $\mathcal{RC}$ can reveal the real identity.

UCL: This property supports the linkability between the multi-authentication information from the same $\mathcal{EV}$.

Accountability: This property endows $\mathcal{RC}$ the ability to reveal the real identity of an illegal or malicious $\mathcal{EV}_u$. Moreover, an adversary cannot forge the valid token tk_u^{acc} evading accountability without knowing $\mathcal{RC}$'s private key $ask_{\mathcal{RC}}$.

Anonymous Payment: To preserve the privacy of $\mathcal{EV}$s, the payment process should support anonymity, in the sense that any other entity (apart from the $\mathcal{Cmt}$) is cannot determine $\mathcal{EV}$s' real identities by analyzing the transactions.

Non-frameability: This property requires no entities including $\mathcal{RC}$ and $\mathcal{Cmt}$ could frame an innocent $\mathcal{EV}_u$ of having misbehaved. In other words, even if an adversary can collude with arbitrary entities, it cannot forge a valid anonymous authentication $(\pi_u^{\text{auth}}, \widehat{cred}_u)$ without knowing the $\mathcal{EV}_u$'s private key $sk_{\mathcal{EV}_u}$.

Unforgeability of credentials: This property implies if the $\mathcal{Iss}$'s private key $sk_{\mathcal{Iss}}$ is not disclosed, the adversary cannot forge a valid credential.

Payment binding: It implies a binding relationship between anonymous authentication information and payment information, which ensures that $\mathcal{CS}$ can confirm which anonymized $\mathcal{EV}$ the payment came from and prevents the malicious from generating unaccountable payment information.

Mutual authentications: To prevent malicious $\mathcal{EV}$s or illegal $\mathcal{CS}$s from appearing in V2G networks, $\mathcal{EV}$s and $\mathcal{CS}$ must perform mutual authentications before the service begins, which can ensure that the legal $\mathcal{CS}$ can be confident that $\mathcal{EV}$ is eligible for accessing the service and vice versa.

Revocability: The $\mathcal{RC}$ can revoke malicious $\mathcal{EV}$s and prohibit them from querying any service in V2G networks.

Min-AD: This property means $\mathcal{EV}$ only presents the attributes and corresponding redacted credentials that can match attributes the access policy requires. In PAP, it guarantees that $\mathcal{EV}$ can disclose a set of required attributes while keeping their other attributes hidden in the authentication phase.

No double-spending: In DAPs, this property means that no adversary can spend the same coin twice anonymously.

IV. Our Construction

Based on the above definitions, this section presents a construction of PAP implemented by the pairing cryptography[28].

A. Initialization Phase

In the initialization phase, we assume that parties in PAP should register to the $\mathcal{RC}$ before performing their initialization.

▪$\mathrm{Setup}(1^\lambda)$: Given a secure parameter λ and an asymmetric bilinear map $e: \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_\mathrm{T}$, where $(g_1, \dot{g}_1, \ddot{g}_1)$ and g_2 are the generators of groups $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. The public parameter can be denoted by $pp := (\mathbb{G}_1, \mathbb{G}_2, g_1, \dot{g}_1, \ddot{g}_1, g_2, e, p, \mathrm{H}_1, \mathrm{H}_2, \mathrm{H}_3)$, where $\mathrm{H}_1: \{0,1\}^* \to \mathbb{G}_1$, $\mathrm{H}_2: \{0,1\}^* \to \mathbb{G}_2$ and $\mathrm{H}_3: \{0,1\}^* \to \mathbb{Z}_p$ are hash functions, p is the prime order of $\mathbb{G}_1$ and $\mathbb{G}_2$.

▪$\mathrm{GenK}_{\mathcal{RC}}(pp)$: The $\mathcal{RC}$ chooses $\xi_1, \xi_2, \xi_3 \leftarrow \mathbb{Z}_\mathrm{p}$, and sets $sk_{\mathcal{RC}} := (\xi_1, \xi_2, \xi_3)$ as its private keys, and then computes $pk_{\mathcal{RC}} := (pk_{\mathcal{RC},1}, pk_{\mathcal{RC},2}, pk_{\mathcal{RC},3}) := (g_2^{\xi_1}, g_2^{\xi_2}, g_2^{\xi_3})$. $\mathcal{RC}$ picks $ask_{\mathcal{RC}} \leftarrow \mathbb{Z}_p$ and sets $apk_{\mathcal{RC}} := g_1^{ask_{\mathcal{RC}}}$.

▪$\mathrm{GenK}_{\mathcal{Iss}}(pp)$: $\mathcal{Iss}_k$ chooses $x_k, y_{k,0}, y_{k,1}, \ldots, y_{k,N_{a,k}} \leftarrow \mathbb{Z}_p$ as its private key $sk_{\mathcal{Iss}_k}$, and computes $X_k := g_2^{x_k}$, $\{Y_{k,i} := g_2^{y_{k,i}}, Y'_{k,i} := g_1^{y_{k,i}}\}_{i \in [N_{a,k}]}$, $\{Z_{i,j,k} := g_2^{y_{k,i} \cdot y_{k,j}}\}_{i,j \in [N_{a,k}] \wedge i \neq j}$, and sets $pk_{\mathcal{Iss}_k} := \left(X_k, \{(Y_{k,i}, Y'_{k,i})\}_{0 \le i \le N_{a,k}}, \{Z_{i,j,k}\}_{0 \le i \neq j \le N_{a,k}}\right)$ as its public key.

▪$\mathrm{GenK}_{\mathcal{EV}}(pp)$: Each $\mathcal{EV}$ picks $sk_{\mathcal{EV}} \leftarrow \mathbb{Z}_p$, sets $pk_{\mathcal{EV}} := g_1^{sk_{\mathcal{EV}}}$.

▪ $\mathrm{GenK}_{\mathcal{Cmt}}(pp)$: $\mathcal{Cmt}$ picks $sk_{\mathcal{Cmt}} \leftarrow \mathbb{Z}_p$, and sets $pk_{\mathcal{Cmt}} := g_1^{sk_{\mathcal{Cmt}}}$. Note that $\mathrm{GenK}_{\mathcal{Cmt}}(\cdot)$ is executed only once while the key pair is shared among the commissioners, and we propose that commissioners use group key agreement protocols[33] to share this key pair.

B. Registration Phase

To dynamically manage $\mathcal{EV}$s, the $\mathcal{RC}$ maintains a ℓ-depth complete binary tree as Sec.II.C. The token issued by the $\mathcal{RC}$ is required for each $\mathcal{EV}$ to enter the V2G networks.

▪$\mathrm{Reg}_{\mathcal{EV}}(pp, sk_{\mathcal{EV}_u}, pk_{\mathcal{EV}_u}, pk_{\mathcal{RC}})$: $\mathcal{EV}$ first selects $r_a \leftarrow \mathbb{Z}_p$, computes $A := g_1^{r_a}$, $c := \mathrm{H}_3(pk_{\mathcal{RC}} || pk_{\mathcal{EV}_u} || A)$, $s_a := r_a + c \cdot sk_{\mathcal{EV}_u}$, and sends $\pi_u^{\mathrm{tok}} := (A, c, s_a)$ to the $\mathcal{RC}$.

▪ $\mathrm{Reg}_{\mathcal{RC}}(pp, sk_{\mathcal{RC}}, t, pk_{\mathcal{EV}_u}, \pi_u^{\mathrm{tok}}, ID_u)$: Upon receiving π_u^{tok}, $\mathcal{RC}$ computes $A' := g_1^{s_a} pk_{\mathcal{EV}_u}^{-c}$, and $c' := \mathrm{H}_3(pk_{\mathcal{RC}} || pk_{\mathcal{EV}_u} || A')$. If $c' = c$, the $\mathcal{RC}$ issues the revocation token for the $\mathcal{EV}$. And then, it places the $\mathcal{EV}$ into a leaf node $n_{\ell,u}$, and assign a path $\rho_u := (n_{1,u}, \ldots, n_{\ell,u})$ from the root node to $n_{\ell,u}$.

Then, it generates $tk_u := (tk_u^{\mathrm{acc}}, tk_u^{\mathrm{rev}})$, where $n_i \in \rho_u$, $tk_u^{\mathrm{acc}} := \left\{\left(\eta_i \leftarrow \mathbb{Z}_p, tk_{u,n_i}^{\mathrm{acc}*} = \left(\dot{g}_1 \ddot{g}_1^{n_i} pk_{\mathcal{EV}_u}\right)^{\frac{1}{\xi_1 + \eta_i}}\right)\right\}_{n_i \in \rho_u}$ and $tk_u^{\mathrm{rev}} := h_{t,n^*}^{\xi_1 + \xi_2 \cdot n^* + \xi_3 \cdot t}$, $h_{t,n^*} := \mathrm{H}_1(t \parallel n^*)$, n^* is an intersection node between ρ_u and $S_R^{(t)}$, i.e. $n^* \in \rho_u \cap S_R^{(t)}$.

Finally, $\mathcal{RC}$ stores $(\rho_u, ID_u, pk_{\mathcal{EV}_u}, tk_u)$ in database $\mathrm{db}_{\mathrm{Reg}}$ and forwards $(tk_u := (tk_u^{\mathrm{acc}}, tk_u^{\mathrm{rev}}), \rho_u)$ to $\mathcal{EV}_u$.

▪ $\mathrm{Verify}_{\mathcal{EV}}^{\mathrm{tok}}(pp, pk_{\mathcal{RC}}, pk_{\mathcal{EV}_u}, tk_u^{\mathrm{acc}}, tk_u^{\mathrm{rev}}, \rho_u, t) \to \frac{1}{0}$: $\mathcal{EV}$ verifies the validity of tk_u by checking if $\left\{e\left(tk_{u,n_i}^{\mathrm{acc}*}, pk_{\mathcal{RC},1} \cdot g_2^{\eta_i}\right) = e\left(\dot{g}_1 \cdot \ddot{g}_1^{n_i} \cdot pk_{\mathcal{EV}_u}, g_2\right)\right\}_{n_i \in \rho_u}$ and $e(tk_u^{\mathrm{rev}}, g_2) = e(h_{t,n^*}, pk_{\mathcal{RC},1} \cdot pk_{\mathcal{RC},2}^{n^*} \cdot pk_{\mathcal{RC},3}^{t})$ hold. If all hold, return 1; otherwise, 0.

Correctness. The correctness of the equation above can be proved as follows:

$$\begin{aligned} e\left(tk_{u,n_i}^{\mathrm{acc}*}, pk_{\mathcal{RC},1} \cdot g_2^{\eta_i}\right) &= e\left(\left(\dot{g}_1 \ddot{g}_1^{n_i} pk_{\mathcal{EV}_u}\right)^{\frac{1}{\xi_1 + \eta_i}}, g_2^{\xi_1 + \eta_i}\right) \\ &= e(\dot{g}_1 \ddot{g}_1^{n_i} pk_{\mathcal{EV}_u}, g_2) \end{aligned}$$

$$\begin{aligned} e(tk_u^{\mathrm{rev}}, g_2) &= e\left(h_{t,n^*}^{\xi_1 + \xi_2 \cdot n^* + \xi_3 \cdot t}, g_2\right) \\ &= e\left(h_{t,n^*}, g_2^{\xi_1 + \xi_2 \cdot n^* + \xi_3 \cdot t}\right) \\ &= e(h_{t,n^*}, pk_{\mathcal{RC},1} pk_{\mathcal{RC},2}^{n^*} pk_{\mathcal{RC},3}^{t}) \end{aligned}$$

C. Issue Phase

In this phase, $\mathcal{EV}$ requests multi-issuer to obtain aggregatable attribute-based credentials. The process of aggregation is that each issuer sequentially signs attributes and appends its credential to a given aggregated credential.

▪$\mathrm{Issue}_{\mathcal{EV}}(pp, sk_{\mathcal{EV}_u}, pk_{\mathcal{Iss}_k})$: As the $\mathrm{Reg}_{\mathcal{EV}}$ algorithm, $\mathcal{EV}_u$ generates a proof π_u^{cred}, and sends $(\pi_u^{\mathrm{cred}}, \hat{\sigma}_u^{(k-1)})$ to $\mathcal{Iss}_k$.[1]

▪ $\mathrm{Issue}_{\mathcal{Iss}}(pp, pk_{\mathcal{EV}_u}, sk_{\mathcal{Iss}_k}, \{\mathfrak{a}_{k,i}\}, \hat{\sigma}_u^{(k-1)}, \pi_u^{\mathrm{cred}})$: On receiving the request from $\mathcal{EV}_u$, $\mathcal{Iss}_k$ generate an aggregated signature $\hat{\sigma}_u^{(k)}$ based on $\hat{\sigma}_u^{(k-1)}$ from $\mathcal{Iss}_{k-1}$. Specifically, if π_u^{cred} is valid, $\mathcal{Iss}_k$ chooses $r_k \leftarrow \mathbb{Z}_p$ to compute a signature $\sigma_k := (\sigma_{k,1}, \sigma_{k,2}) := (\sigma_{k-1,1}^{r_k}, \sigma_{k-1,2}^{r_k} \cdot \sigma_{k-1,1}^{r_k \cdot (x_k + \sum_i y_{k,i} \cdot \mathfrak{a}_{k,i})} \cdot \phi_{k-1}^{r_k \cdot y_{k,0}})$ and returns $\hat{\sigma}_u^{(k)} := (\sigma_k := (\sigma_{k,1}, \sigma_{k,2}), \phi_k := \phi_{k-1}^{r_k})$ to $\mathcal{EV}$.

Finally, the aggregated signature received by $\mathcal{EV}$ can be parsed as:

$$\begin{aligned} \sigma_{N_\mathcal{J}} &:= \left(\sigma_{N_\mathcal{J},1}, \sigma_{N_\mathcal{J},2}\right) \\ &:= \left(\sigma_{N_\mathcal{J}-1,1}^{r_{N_\mathcal{J}}}, \sigma_{N_\mathcal{J}-1,2}^{r_{N_\mathcal{J}}} \cdot \sigma_{N_\mathcal{J}-1,1}^{r_{N_\mathcal{J}} \cdot \left(x_{N_\mathcal{J}} + \sum_i y_{N_\mathcal{J},i} \cdot \mathfrak{a}_{N_\mathcal{J},i}\right)} \cdot \phi_{N_\mathcal{J}-1}^{r_{N_\mathcal{J}} \cdot y_{N_\mathcal{J},0}}\right) \\ &:= \left(g_1^{\prod_{k \in [N_\mathcal{J}]} r_k}, pk_{\mathcal{EV}_u}^{\left(\prod_{k \in [N_\mathcal{J}]} r_k\right) \cdot \left(\sum_{k \in [N_\mathcal{J}]} y_{k,0}\right)} \cdot \sigma_1^{\sum_{k,i}(x_k + y_{k,i} \mathfrak{a}_{k,i})}\right). \end{aligned}$$

As a result, $\mathcal{EV}$ obtains an aggregated signature $\sigma_{N_\mathcal{J}^*}$. ($N_\mathcal{J}^*$ is the number of issuers with which $\mathcal{EV}_u$ requests credentials.)

▪$\mathrm{Verify}_{\mathcal{EV}}^{\mathrm{cred}}(pp, \{pk_{\mathcal{Iss}_k}\}_{k \in [N_\mathcal{J}]}, sk_{\mathcal{EV}_u}, \hat{\sigma}_u^{(N_\mathcal{J})})$: Upon receiving $\hat{\sigma}_u^{(N_\mathcal{J})}$, $\mathcal{EV}$ verifies the validity by checking whether $e\left(\sigma_{N_\mathcal{J},2}, g_2\right) = e(\prod_{k \in [N_\mathcal{J}], i \in [N_{a,k}]} Y_{k,0}^{sk_{\mathcal{EV}_u}} X_k Y_{k,i}^{\mathfrak{a}_{k,i}}, \sigma_{N_\mathcal{J},1})$ holds, if positive, it returns $cred_u := \sigma_{N_\mathcal{J}}$ and $\perp$ otherwise.

[1] Note that in the case of $k := 0$, we assume the initialized signature as $\hat{\sigma}_u^{(0)} := (\sigma_0 := (\sigma_{0,1}, \sigma_{0,2}), \phi_0) := ((g_1, 1_{\mathbb{G}_1}), pk_{\mathcal{EV}_u})$.

Correctness. The correctness of the equation above can be proved as follows:

$$e\left(\sigma_{N_J,2}, g_2\right) = e\left(pk_{\mathcal{EV}_u}^{\left(\prod_{k\in[N_J]} r_k\right)\cdot\left(\sum_{k\in[N_J]} y_{k,0}\right)} \cdot \sigma_{N_J,1}^{\sum_{k,i}(x_k + y_{k,i}\mathfrak{a}_{k,i})}, g_2\right)$$

$$= e\left(g_1^{sk_{\mathcal{EV}_u}\cdot\left(\prod_{k\in[N_J]} r_k\right)\cdot\left(\sum_{k\in[N_J]} y_{k,0}\right)+\left(\prod_{k\in[N_J]} r_k\right)\cdot\sum_{k,i}(x_k + y_{k,i}\mathfrak{a}_{k,i})}, g_2\right)$$

$$= e(g_1, g_2)^{\left(\prod_{k\in[N_J]} r_k\right)\cdot\left(sk_{\mathcal{EV}_u}\cdot\left(\sum_{k\in[N_J]} y_{k,0}\right)+\sum_{k,i}(x_k + y_{k,i}\mathfrak{a}_{k,i})\right)}$$

$$= e\left(g_1^{\prod_{k\in[N_J]} r_k}, g_2^{sk_{\mathcal{EV}_u}\cdot\left(\sum_{k\in[N_J]} y_{k,0}\right)+\sum_{k,i}(x_k + y_{k,i}\mathfrak{a}_{k,i})}\right)$$

$$= e\left(\sigma_{N_J,1}, \prod_{k,i} Y_{k,0}^{sk_{\mathcal{EV}_u}} X_k Y_{k,i}^{\mathfrak{a}_{k,i}}\right)$$

D. Authentication Phase

In this phase, $\mathcal{EV}_u$ should convince the service provider that: 1) It has a revocation token tk_{u,n^*}^{rev} at the current epoch t; 2) It has a valid accountability token $tk_{u,n^*}^{\text{acc}*}$; 3) It has a set of credentials matching the access policy Γ.

We employ the ZKP system to prove these statements in zero knowledge and the linear message homomorphism to achieve the redaction of attributes for satisfying Min-AD. The details are shown in Figure 3.

User-controlled linkability (UCL). To ensure continuous service is not interrupted, our scheme offers UCL. Specifically, since each authentication information from the same $\mathcal{EV}$ accompanies the same value of (B, φ_4), which is generated by the ID of $\mathcal{CS}$ and $\mathcal{EV}$'s private key, $\mathcal{CS}$ can compare whether the (B, φ_4) of the two information is consistent to determine the linkability of the service.

Protocol 1: When applying for services, $\mathcal{EV}$ shows a redacted credential containing the required attribute index set I_Γ based on the access policy Γ. Let $\{\bar{I}_{k,\Gamma} \coloneqq N_{a,k} \backslash I_\Gamma\}_{k\in[N_J]}$ denote the index of attribute which is not in I_Γ. To satisfy Min-AD, redactable signatures are required, which can refer to the Step 5 and 6 of $\text{Auth}_{\mathcal{EV}}$.

$\text{Auth}_{\mathcal{EV}}$ $\left(pp, sk_{\mathcal{EV}_u}, tk_u, cred_u, \{\mathfrak{a}_i\}_{i\in I_\Gamma}, \Gamma, \{pk_{\mathcal{JSS}_k}\}_{k\in[N_J]}, apk_{\mathcal{RC}}, \boxed{bsn}\right) \to (\pi_u^{\text{auth}}, \widehat{cred}_u, (sk_{ots}, vk_{ots}))$**:**

1. Generate $sk_{ots} \coloneqq (sk_{ots}^{(1)}, sk_{ots}^{(2)}) \leftarrow \mathbb{Z}_p$, compute $vk_{ots} \coloneqq (vk_{ots}^{(1)} \coloneqq g_1^{sk_{ots}^{(1)}}, vk_{ots}^{(2)} \coloneqq g_1^{sk_{ots}^{(2)}})$ and select $r_v, r_e, s_e, s_z, s_\eta, s_n, s_o, s_\beta \leftarrow \mathbb{Z}_p$.
2. Set: $\varphi_1 \coloneqq tk_{u,n^*}^{\text{acc}*} \cdot apk_{\mathcal{RC}}^{sk_{ots}^{(1)}}, \varphi_2 \coloneqq tk_{u,n^*}^{\text{rev}} \cdot h_{t,n^*}^{r_e}, \varphi_3 \coloneqq vk_{ots}^{(1)}, \beta \coloneqq sk_{ots}^{(1)} \cdot \eta_{n^*}$, where $n^* \in \rho_u \cap S_R^{(t)}$.
3. Let bsn denote a basename, such as the ID of $\mathcal{CS}$, compute $\boxed{(\mathrm{B} \coloneqq \mathrm{H}_1(bsn), \varphi_4 \coloneqq \mathrm{B}^{sk_{\mathcal{EV}_u}}, \varphi_5 \coloneqq \mathrm{B}^{s_z})}$ for the linkability of services.
4. Compute: $D_1 \coloneqq e(\varphi_1, g_1,)^{-s_m} e(g_1, \ddot{g}_2)^{s_n} e(g_1, g_2)^{s_z} e\left(apk_{\mathcal{RC}}, pk_{\mathcal{RC},1}\right)^{s_o} e(apk_{\mathcal{RC}}, g_2)^{s_\beta}$
   $D_2 \coloneqq e\left(h_{t,n^*}, pk_{\mathcal{RC},2}\right)^{s_n} e\left(h_{t,n^*}, g_2\right)^{s_e}$
   $D_3 \coloneqq {g_1}^{s_o}$
5. For each credential, set $I_0 \coloneqq \{0\} \cup I_\Gamma$ and compute $\{(\tilde{\sigma}_{k,1}, \tilde{\sigma}_{k,2})\}_{k\in[N_J]}$, where $\tilde{\sigma}_{k,1} \coloneqq g_2^{r_v} \cdot \prod_{j\in\bar{I}_{k,\Gamma}} Y_{k,j}^{\mathfrak{a}_{k,j}}, \tilde{\sigma}_{k,2} \coloneqq \left(\prod_{i\in I_0} Y_{k,i}\right)^{r_v} \cdot \prod_{i\in I_0, j\in\bar{I}_{k,\Gamma}} Z_{i,j,k}^{\mathfrak{a}_{k,j}}$.
6. Compute $\tilde{\sigma}_1 \coloneqq g_2^{r_v} \cdot \prod_{k\in[N_J]} \prod_{j\in\bar{I}_{k,\Gamma}} Y_{k,j}^{\mathfrak{a}_{k,j}}, \sigma_1' \coloneqq \sigma_{N_J,1}^{sk_{ots}^{(1)}}, \sigma_2' \coloneqq \sigma_{N_J,2}^{sk_{ots}^{(1)}} \cdot (\sigma_1')^{r_v}, E \coloneqq e\left(\sigma_1', \prod_{k=1}^{N_J} Y_{k,0}^{s_z}\right)$ and set $\widehat{cred}_u \coloneqq \left(\{(\tilde{\sigma}_{k,1}, \tilde{\sigma}_{k,2})\}_{k\in[N_J]}, \sigma_1', \sigma_2', \tilde{\sigma}_1\right)$.
7. Let $\mathbb{C}_{auth} \coloneqq \mathrm{H}_3(\varphi_1 \parallel \varphi_2 \parallel \varphi_3 \parallel \boxed{\varphi_4 \parallel \varphi_5} \parallel D_1 \parallel D_2 \parallel D_3 \parallel E \parallel \widehat{cred}_u)$, and compute $w_z \coloneqq s_z + \mathbb{C}_{auth} \cdot sk_{\mathcal{EV}_u}, w_\eta \coloneqq s_\eta + \mathbb{C}_{auth} \cdot \eta_{n^*}, w_n \coloneqq s_n + \mathbb{C}_{auth} \cdot n^*, w_o \coloneqq s_o + \mathbb{C}_{auth} \cdot sk_{ots}^{(1)}, w_\beta \coloneqq s_\beta + \mathbb{C}_{auth} \cdot \beta, w_e \coloneqq s_e + \mathbb{C}_{auth} \cdot r_e$.
8. Send $(\pi_u^{\text{auth}}, \widehat{cred}_u)$ to $\mathcal{CS}$, where $\pi_u^{\text{auth}} \coloneqq (\varphi_1, \varphi_2, \varphi_3, \boxed{\mathrm{B}, \varphi_4, \varphi_5,} w_z, w_e, w_\eta, w_n, w_o, w_\beta, \mathbb{C}, E)$.

$\text{Verify}_{\mathcal{CS}}^{\text{auth}}$ $\left(pp, \{pk_{\mathcal{JSS}_k}\}_{k\in[N_J]}, \pi_u^{\text{auth}}, pk_{\mathcal{RC}}, \widehat{cred}_u, \{\mathfrak{a}_i\}_{i\in I_\Gamma}, t\right) \to 0/1$**:**

1. Compute: $D_1' \coloneqq e(\varphi_1, g_1)^{-w_\eta} e(g_1, \ddot{g}_2)^{w_n} e(g_1, g_2)^{w_z} e(apk_{\mathcal{RC}}, pk_{\mathcal{RC},1})^{w_o} e(apk_{\mathcal{RC}}, g_2)^{w_\beta} \left(\frac{e(\varphi_1, pk_{\mathcal{RC},1})}{e(g_1, \dot{g}_2)}\right)^{-\mathbb{C}_{auth}}$
   $D_2' \coloneqq e\left(h_{t,n^*}, pk_{\mathcal{RC},2}\right)^{w_n} e\left(h_{t,n^*}, g_2\right)^{w_e} \left(\frac{e(\varphi_2, g_1)}{e(h_{t,n^*}, pk_{\mathcal{RC},1}) e(h_{t,n^*}, pk_{\mathcal{RC},3})^t}\right)^{-\mathbb{C}_{auth}}$
   $D_3' \coloneqq g_1^{w_o} \varphi_3^{-\mathbb{C}_{auth}}$
   $\boxed{\varphi_5' \coloneqq \mathrm{B}^{w_z} \cdot \varphi_4^{-\mathbb{C}_{auth}}}$
   $\mathbb{C}_{auth}' \coloneqq \mathrm{H}_3(\varphi_1 \parallel \varphi_2 \parallel \varphi_3 \parallel \boxed{\varphi_4 \parallel \varphi_5'} \parallel D_1' \parallel D_2' \parallel D_3' \parallel E \parallel \widehat{cred}_u)$
2. If $\mathbb{C}_{auth} = \mathbb{C}_{auth}'$ holds, it next computes $F \coloneqq e(\tilde{\sigma}_1 \cdot \prod_{i\in I_\Gamma, k\in[N_J]} X_k \cdot Y_{k,i}^{\mathfrak{a}_{k,i}}, (\sigma_1')^{-1})$, and verifies whether $e\left(\prod_{k=1}^{N_J} Y_{k,0}^{w_z}, \sigma_1'\right) \cdot E^{-1} = \left(F \cdot e(\sigma_2', g_2)\right)^{\mathbb{C}_{auth}'}$ holds, if not, abort; otherwise, it verifies $e\left(\tilde{\sigma}_{k,1}, \prod_{i\in I_0} Y_{k,i}'\right) = e(g_1, \tilde{\sigma}_{k,2})$ for each element in the set $\{(\tilde{\sigma}_{k,1}, \tilde{\sigma}_{k,2})\}_{k\in[N_J]}$. If all equations are hold, the algorithm returns 1 and 0 otherwise.
3. If the tuple $\boxed{(\mathrm{B}, \varphi_4)}$ is consistent with the one in the last authentication information, then $\mathcal{CS}$ continues the last service for $\mathcal{EV}$.

Figure 3.Illustrates the process of authentication between the $\mathcal{EV}$ and the $\mathcal{CS}$. If linkability is not required, removing the box $\boxed{}$ parts.

Correctness. The correctness of the equation in $\text{Verify}_{\mathcal{CS}}^{\text{auth}}$ can be proved as follows:

$$D_1 = e(\varphi_1, g_2)^{-s_\eta} e(\ddot{g}_1, g_2)^{s_n} e(g_1, g_2)^{s_z} e\left(apk_{\mathcal{RC}}, pk_{\mathcal{RC},1}\right)^{s_o} e(apk_{\mathcal{RC}}, g_2)^{s_\beta}$$

$$= e\left(\left(\dot{g}_1 \ddot{g}_1^{n^*} pk_{\mathcal{EV}_u}\right)^{\frac{1}{\xi_1+\eta_{n^*}}}, g_2\right)^{-s_\eta} e(\ddot{g}_1, g_2)^{s_n} e(g_1, g_2)^{s_z} e(g_1, g_2)^{s_o\cdot\xi_1\cdot ask_{\mathcal{RC}}} e(g_1, g_2)^{s_\beta\cdot\xi_1\cdot ask_{\mathcal{RC}}}$$

$$= e(\dot{g}_1, g_2)^{\frac{-s_\eta}{\xi_1+\eta_{n^*}}} e\left(\ddot{g}_1^{n^*}, g_2\right)^{\frac{-s_\eta}{\xi_1+\eta_{n^*}}} e\left(pk_{\mathcal{EV}_u}, g_2\right)^{\frac{-s_\eta}{\xi_1+\eta_{n^*}}} e(\ddot{g}_1, g_2)^{s_n} e(g_1, g_2)^{s_z} e(g_1, g_2)^{s_o\cdot\xi_1\cdot ask_{\mathcal{RC}}} e(g_1, g_2)^{s_\beta\cdot\xi_1\cdot ask_{\mathcal{RC}}}$$

$$= e(\dot{g}_1, g_2)^{\frac{-s_\eta}{\xi_1+\eta_{n^*}}} \cdot e(\ddot{g}_1, g_2)^{s_n+\frac{-s_\eta\cdot n^*}{\xi_1+\eta_{n^*}}} \cdot e(g_1, g_2)^{\frac{-s_\eta\cdot sk_{\mathcal{EV}_u}}{\xi_1+\eta_{n^*}}+s_z+s_o\cdot\xi_1\cdot ask_{\mathcal{RC}}+s_\beta\cdot\xi_1\cdot ask_{\mathcal{RC}}}$$

$$D_1' = e(\varphi_1, g_2)^{-w_\eta} e(\ddot{g}_1, g_2)^{w_n} e(g_1, g_2)^{w_z} e\left(apk_{\mathcal{RC}}, pk_{\mathcal{RC},1}\right)^{w_o} e(apk_{\mathcal{RC}}, g_2)^{w_\beta} \left(\frac{e\left(\varphi_1, pk_{\mathcal{RC},1}\right)}{e(\dot{g}_1, g_2)}\right)^{-\mathbb{C}_{auth}}$$

$$= e(\dot{g}_1, g_2)^{\frac{-w_\eta}{\xi_1+\eta_{n^*}}} e(\ddot{g}_1, g_2)^{\frac{-w_\eta\cdot n^*}{\xi_1+\eta_{n^*}}} e(g_1, g_2)^{\frac{-w_\eta\cdot sk_{\mathcal{EV}_u}}{\xi_1+\eta_{n^*}}} e(\ddot{g}_1, g_2)^{w_n} \cdot e(g_1, g_2)^{w_z} e(g_1, g_2)^{s_o\cdot\xi_1\cdot ask_{\mathcal{RC}}} e(g_1, g_2)^{s_\beta\cdot\xi_1\cdot ask_{\mathcal{RC}}} .$$

$$e(\dot{g}_1, g_2)^{\frac{-\mathbb{c}_{auth}\cdot\xi_1}{\xi_1+\eta_{n^*}}} e(\ddot{g}_1, g_2)^{\frac{-\mathbb{c}_{auth}\cdot\xi_1\cdot n^*}{\xi_1+\eta_{n^*}}} e(g_1, g_2)^{\frac{-\mathbb{c}_{auth}\cdot\xi_1\cdot sk_{\mathcal{EV}_u}}{\xi_1+\eta_{n^*}}} \cdot$$
$$e(\dot{g}_1, g_2)^{\mathbb{c}_{auth}}$$
$$= e(\dot{g}_1, g_2)^{-\left(\frac{w_\eta}{\xi_1+\eta_{n^*}}+\frac{\mathbb{c}_{auth}\cdot\xi_1}{\xi_1+\eta_{n^*}}\right)+\mathbb{c}_{auth}} \cdot$$
$$e(\ddot{g}_1, g_2)^{-\left(\frac{w_\eta\cdot n^*}{\xi_1+\eta_{n^*}}+\frac{\mathbb{c}_{auth}\cdot\xi_1\cdot n^*}{\xi_1+\eta_{n^*}}\right)+w_n}$$
$$e(g_1, g_2)^{-\left(\frac{w_\eta\cdot sk_{\mathcal{EV}_u}}{\xi_1+\eta_{n^*}}+\frac{\mathbb{c}_{auth}\cdot\xi_1\cdot sk_{\mathcal{EV}_u}}{\xi_1+\eta_{n^*}}\right)+w_z+s_o\cdot\xi_1\cdot ask_{\mathcal{RC}}+s_\beta\cdot\xi_1\cdot ask_{\mathcal{RC}}}$$
$$= e(\dot{g}_1, g_2)^{-\frac{s_\eta+\mathbb{c}_{auth}\cdot(\eta_{n^*}+\xi_1)}{\xi_1+\eta_{n^*}}+\mathbb{c}_{auth}} \cdot$$
$$e(\ddot{g}_1, g_2)^{-\frac{n^*\cdot\left(s_\eta+\mathbb{c}_{auth}\cdot(\eta_{n^*}+\xi_1)\right)}{\xi_1+\eta_{n^*}}+w_n} \cdot$$
$$e(g_1, g_2)^{-\frac{sk_{\mathcal{EV}_u}\cdot\left(s_\eta+\mathbb{c}_{auth}\cdot(\eta_{n^*}+\xi_1)\right)}{\xi_1+\eta_{n^*}}+w_z+s_o\cdot\xi_1\cdot ask_{\mathcal{RC}}+s_\beta\cdot\xi_1\cdot ask_{\mathcal{RC}}}$$
$$= e(\dot{g}_1, g_2)^{\frac{-s_\eta}{\xi_1+\eta_{n^*}}} e(\ddot{g}_1, g_2)^{s_n+\frac{-s_\eta\cdot n^*}{\xi_1+\eta_{n^*}}} \cdot$$
$$e(g_1, g_2)^{\frac{-s_\eta\cdot sk_{\mathcal{EV}_u}}{\xi_1+\eta_{n^*}}+s_z+s_o\cdot\xi_1\cdot ask_{\mathcal{RC}}+s_\beta\cdot\xi_1\cdot ask_{\mathcal{RC}}}$$
$$= D_1$$

Therefore, the equation $D_1 = D_1'$ can be proved, let $H \coloneqq h_{t,n^*}$. Similarly, the verification of D_2, D_3, φ_5 are as follows:

$$D_2 = e\left(H, pk_{\mathcal{RC},2}\right)^{s_n} e(H, g_2)^{s_e} = e(H, g_2)^{\xi_2\cdot s_n+s_e}$$
$$D_2' = e\left(H, pk_{\mathcal{RC},2}\right)^{w_n} e(H, g_2)^{w_e} \left(\frac{e(\varphi_2, g_2)}{e\left(H, pk_{\mathcal{RC},1}\right)e\left(H, pk_{\mathcal{RC},3}\right)^t}\right)^{-\mathbb{c}_{auth}}$$
$$= e(H, g_2)^{\xi_2\cdot w_n+w_e} e(H, g_2)^{-\mathbb{c}_{auth}\cdot(\xi_1+\xi_2\cdot n^*+\xi_3\cdot t+r_e)}$$
$$e(H, g_2)^{\mathbb{c}_{auth}\cdot\xi_1} e(H, g_2)^{\mathbb{c}_{auth}\cdot\xi_3\cdot t}$$
$$= e(H, g_2)^{\substack{\xi_2\cdot(s_n+\mathbb{c}_{auth}\cdot n^*)+s_e+\mathbb{c}_{auth}\cdot r_e-\mathbb{c}_{auth}\cdot(\xi_1+\xi_2\cdot n^*+\xi_3\cdot t+r_e)+ \\ \mathbb{c}_{auth}\cdot\xi_1+\mathbb{c}_{auth}\cdot\xi_3\cdot t}}$$
$$= e(H, g_2)^{\xi_2\cdot s_n+s_e}$$
$$= D_2$$
$$D_3' = g_1^{w_o}\varphi_3^{-\mathbb{c}_{auth}}$$
$$= g_1^{s_o+\mathbb{c}_{auth}\cdot sk_{ots}^{(1)}} \cdot g_1^{-\mathbb{c}_{auth}\cdot sk_{ots}^{(1)}}$$
$$= g_1{}^{s_o}$$
$$= D_3$$
$$\varphi_5' = B^{w_z}\cdot\varphi_4^{-\mathbb{c}_{auth}} = B^{s_z+\mathbb{c}_{auth}\cdot sk_{\mathcal{EV}_u}} \cdot B^{-\mathbb{c}_{auth}\cdot sk_{\mathcal{EV}_u}}$$
$$= B^{s_z} = \varphi_5$$

Next, we present the correctness verification of the randomized signature:

$$e\left(\sigma_1', \prod_{k=1}^{N_J} Y_{k,0}^{w_z}\right)\cdot E^{-1} = \left(F\cdot e(\sigma_2', g_2)\right)^{\mathbb{c}_{auth}'}$$

Proof. According to the above, E, F can be parsed as: $E \coloneqq e\left(\prod_{k=1}^{N_J} Y_{k,0}^{s_z}, \sigma_1'\right)$, $F \coloneqq e(\tilde{\sigma}_1\cdot\prod_{i\in I_\Gamma,k\in[N_J]} X_k\cdot Y_{k,i}^{\alpha_{k,i}}$, $(\sigma_1')^{-1})$, so the above equation can be transformed as:

$$L = e\left(g_1^{sk_{ots}^{(1)}\cdot\prod_k r_k}, g_2^{w_z\cdot\sum_k y_{k,0}}\right)\cdot e\left(g_1^{sk_{ots}^{(1)}\cdot\prod_k r_k}, g_2^{s_z\cdot\sum_k y_{k,0}}\right)^{-1}$$
$$= e(g_1, g_2)^{sk_{ots}^{(1)}\cdot\prod_k r_k\cdot(\sum_k y_{k,0})\cdot\mathbb{c}_{auth}\cdot sk_{\mathcal{EV}_u}}$$
$$R = e\left((\sigma_1')^{-1}, \tilde{\sigma}_1\cdot\prod_{i\in I_\Gamma,k\in[N_J]} X_k\cdot Y_{k,i}^{\alpha_{k,i}}\right)^{\mathbb{c}_{auth}'}\cdot e(\sigma_2', g_2)^{\mathbb{c}_{auth}'}$$
$$= e\left(g_1^{-sk_{ots}^{(1)}\cdot\prod_k r_k}, g_2^{r_v+\sum_{k\in[N_J^*],i\in[N_{a,k}]} y_{k,i}\alpha_{k,i}+x_k}\right)^{\mathbb{c}_{auth}'}\cdot$$
$$e\left(g_1^{\substack{sk_{ots}^{(1)}\cdot sk_{\mathcal{EV}_u}\cdot(\prod_k r_k)\cdot(\sum_k y_{k,0})+sk_{ots}^{(1)}\cdot(\prod_k r_k)\cdot \\ \sum_{k,i}(y_{k,i}\alpha_{k,i}+x_k)+r_v\cdot sk_{ots}^{(1)}\cdot\prod_k r_k}}, g_2\right)^{\mathbb{c}_{auth}'}$$
$$= e(g_1, g_2)^{sk_{ots}^{(1)}\cdot\prod_k r_k\cdot(\sum_k y_{k,0})\cdot\mathbb{c}_{auth}\cdot sk_{\mathcal{EV}_u}} = L$$

Hence, the correctness of randomized signature can be verified. And finally, we show the correctness verification of redactable signature as below.

$$e\left(\tilde{\sigma}_{k,1}, \prod_{i\in I_0} Y_{k,i}'\right) = e(g_1, \tilde{\sigma}_{k,2})$$

Proof. According to the process of authentication, $\tilde{\sigma}_{k,1}$, $\tilde{\sigma}_{k,2}$ can be parsed as $\tilde{\sigma}_{k,1} \coloneqq g_2^{r_v}\cdot\prod_{j\in\bar{I}_{k,\Gamma}} Y_{k,j}^{\alpha_{k,j}}$, $\tilde{\sigma}_{k,2} \coloneqq \left(\prod_{i\in I_0} Y_{k,i}\right)^{r_v}\cdot\prod_{i\in I_0,j\in\bar{I}_{k,\Gamma}} Z_{i,j,k}^{\alpha_{k,j}}$, so the above equation can be transformed as:

$$L = e\left(g_2^{r_v}\cdot\prod_{j\in\bar{I}_{k,\Gamma}} Y_{k,j}^{\alpha_{k,j}}, \prod_{i\in I_0} Y_{k,i}'\right)$$
$$= e(g_1, g_2)^{\left(r_v+\sum_{j\in\bar{I}_{k,\Gamma}} y_{k,j}\alpha_{k,j}\cdot\sum_{i\in I_0} y_{k,i}\right)}$$
$$R = e\left(g_1, \left(\prod_{i\in I_0} Y_{k,i}\right)^{r_v}\cdot\prod_{i\in I_0,j\in\bar{I}_{k,\Gamma}} Z_{i,j,k}^{\alpha_{k,j}}\right)$$
$$= e(g_1, g_2)^{r_v\cdot\sum_{i\in I_0} y_{k,i}+\sum_{i\in I_0,j\in\bar{I}_{k,\Gamma}} y_{k,i}y_{k,j}\alpha_{k,j}}$$
$$= L$$

So far, all the correctness verification in the authentication phase has been proved.

E. Payment Phase

The paid service in V2G mainly includes charging and discharging services, thus, we take these two as examples shown in Figure 4 to illustrate the transaction process when the $\mathcal{EV}$ acts as spender and recipient, respectively.

In the case of $\mathcal{EV}$ as a spender, to guarantee payment anonymity, we design $\mathrm{Gen}_{\mathcal{EV}_u}^{\mathrm{addr}}$ to generate $\mathcal{EV}$'s pseudonymous address based on its wallet address before each transaction. Hence for each transaction process, $\mathcal{EV}$ needs to prove: 1). It has a valid pseudonyms address generated from its wallet address; 2). there exists a binding relationship between the anonymized authentication π_u^{auth} already provided and the transaction tx_{ch} generated in this phase. Generated in this phase. We employ ZKP, as in the authentication phase, to prove the first statement, and arrive at payment binding by OTS.

Payment binding. In the authentication phase, $\mathrm{Auth}_{\mathcal{EV}}$ generates verification-signing key pair of OTS and takes the verification key φ_3 as the element in the proof π_u^{auth}. In the payment phase, $\mathrm{Trans}_{\mathcal{EV}}$ outputs a signature σ_{ots} about the payment information with the signing key of OTS. Once the corresponding verification in both phases is passed, an authenticated $\mathcal{EV}$ has made a payment associated with the authentication information already submitted. Still, the recipient cannot determine the identity of $\mathcal{EV}$ except knowing the binding relationship. Further, if $\mathcal{CS}$ is spender, since it has no anonymity requirement, the transaction process is much simpler. As for the accountability of $\mathcal{CS}$, it can be guaranteed by the signature scheme Σ.

Correctness. The correctness of the equation in $\mathrm{Verify}_{\mathcal{Cmt}}^{\mathrm{tx}}$ can be proved as follows:

$$
\begin{aligned}
R_1' &= g_1^{w_1} \cdot Q_u^{\mathbb{c}_{pay}} \\
&= g_1^{sk_{ots}^{(1)} \cdot sk_{ots}^{(2)} - \mathbb{c}_{pay} \cdot sk_{ots}^{(1)} \cdot sk_{\mathcal{EV}_u}^{\text{addr}}} \cdot {pk_{\mathcal{EV}_u}^{\text{addr}}}^{sk_{ots}^{(1)} \cdot \mathbb{c}_{pay}} \\
&= g_1^{sk_{ots}^{(1)} \cdot sk_{ots}^{(2)} - \mathbb{c}_{pay} \cdot sk_{ots}^{(1)} \cdot sk_{\mathcal{EV}_u}^{\text{addr}} + \mathbb{c}_{pay} \cdot sk_{ots}^{(1)} \cdot sk_{\mathcal{EV}_u}^{\text{addr}}} \\
&= g_1^{sk_{ots}^{(1)} \cdot sk_{ots}^{(2)}} \\
&= R_1 \\
R_3' &= g_1^{w_2} \cdot Q_u'^{\mathbb{c}_{pay}} \cdot pk_{\mathcal{C}mt}^{w_1} / R_2 \\
&= g_1^{sk_{ots}^{(2)} - \mathbb{c}_{pay} \cdot sk_{\mathcal{EV}_u}^{\text{addr}}} \cdot pk_{\mathcal{C}mt}^{sk_{ots}^{(1)} \cdot sk_{\mathcal{EV}_u}^{\text{addr}} \cdot \mathbb{c}_{pay}} \cdot {pk_{\mathcal{EV}_u}^{\text{addr}}}^{\mathbb{c}_{pay}} \cdot \\
&\quad \frac{pk_{\mathcal{C}mt}^{sk_{ots}^{(1)} \cdot sk_{ots}^{(2)} - \mathbb{c}_{pay} \cdot sk_{ots}^{(1)} \cdot sk_{\mathcal{EV}_u}^{\text{addr}}}}{pk_{\mathcal{C}mt}^{sk_{ots}^{(1)} \cdot sk_{ots}^{(2)}}} \\
&= g_1^{sk_{ots}^{(2)}} \\
&= R_3
\end{aligned}
$$

F. Accountability Phase

In this phase, when the $\mathcal{EV}$ performs some malicious behavior after authentication, such as damaging the charging station or refusing to pay a fee in the payment phase, the $\text{Acc}_{\mathcal{RC}}$ algorithm will be triggered:

▪$\text{Acc}_{\mathcal{RC}}(pp, \pi_u^{\text{auth}}, ask_{\mathcal{RC}})$: Upon receiving a request for accountability with evidence from $\mathcal{CS}$ or $\mathcal{C}mt$, $\mathcal{RC}$ parses π_u^{auth} and recovers the accountability token $tk_{u,n^*}^{\text{acc}*} := \varphi_1 / \varphi_3^{ask_{\mathcal{RC}}}$ to trace the real identity of the malicious $\mathcal{EV}$ by looking up the database db_{Reg} with $tk_{u,n^*}^{\text{acc}*}$.

Note that, if necessary, the power of accountability can be distributed to a set of auditors and adopt threshold decryption schemes[31] to reveal the identity of the malicious $\mathcal{EV}$.

$\mathcal{RC}$ continues to perform revocation operations as below if the revocation for malicious $\mathcal{EV}$s is required:

▪$\text{Revoke}_{\mathcal{RC}}(\{\rho_u\}_{u \in [L_t]}, \text{T}_\ell, t)$: To cope with the request to revoke $\mathcal{EV}_u$, $\mathcal{RC}$ removes paths $\{\rho_u\}_{u \in [L_t]}$ from the complete binary tree T_ℓ, and regenerates a new set of root nodes $S_R^{(t)}$ for the next epoch t.

▪$\text{Update}_{\mathcal{RC}}(pp, sk_{\mathcal{RC}}, S_R^{(t)}, t)$: Taking the set $S_R^{(t)}$ as input, $\mathcal{RC}$ updates the revocation token set $\left\{tk_n^{\text{rev}} := h_{n,t}^{\xi_1 + \xi_2 \cdot n + \xi_3 \cdot t}, h_{n,t} := \text{H}_1(t \parallel n)\right\}_{n \in S_R^{(t)}}$.

Protocol 2: Let $E_{req}, E_{bat}, \mathfrak{t}, \mathfrak{n}$ denote the amount of required electricity, the battery capacity, a timestamp, and a nonce, respectively. Note that, in our setting, there is no need to generate a pseudonymous address for $\mathcal{CS}$ because they are public infrastructures. Thus, let $(pk_{\mathcal{CS}}, sk_{\mathcal{CS}}) \leftarrow \Sigma.\text{Setup}(1^\lambda)$ be the wallet address of $\mathcal{CS}$, and $\left(pk_{\mathcal{EV}_u}^{\text{addr}} := g_1^{sk_{\mathcal{EV}_u}^{\text{addr}}}, sk_{\mathcal{EV}_u}^{\text{addr}}\right)$ is a key pair corresponding to the $\mathcal{EV}$'s wallet address.

$\mathcal{EV}$ as spender:

$\text{Gen}_{\mathcal{EV}_u}^{\text{addr}}$ $(pp, pk_{\mathcal{EV}_u}^{\text{addr}}, sk_{\mathcal{EV}_u}^{\text{addr}}, pk_{\mathcal{C}mt}, sk_{ots}) \rightarrow (\widetilde{pk_{\mathcal{EV}_u}^{\text{addr}}}, \widetilde{sk_{\mathcal{EV}_u}^{\text{addr}}}, \text{info}_{pay})$**:**

1. Compute $Q_u := {pk_{\mathcal{EV}_u}^{\text{addr}}}^{sk_{ots}^{(1)}}, Q_u' := pk_{\mathcal{C}mt}^{sk_{ots}^{(1)} \cdot sk_{\mathcal{EV}_u}^{\text{addr}}} \cdot pk_{\mathcal{EV}_u}^{\text{addr}}$ and set $\widetilde{pk_{\mathcal{EV}_u}^{\text{addr}}} := (Q_u, Q_u')$ and $\widetilde{sk_{\mathcal{EV}_u}^{\text{addr}}} := (sk_{\mathcal{EV}_u}^{\text{addr}}, sk_{ots}^{(1)})$.
2. Send $\text{info}_{req} := (E_{req}, E_{bat}, \mathfrak{t}, \mathfrak{n})$ to $\mathcal{CS}$, and the $\mathcal{CS}$ should respond to the info_{pay}, see **Remark1** for details.

$\text{Trans}_{\mathcal{EV}}$ $\begin{pmatrix} pp, pk_{\mathcal{S}} := \widetilde{pk_{\mathcal{EV}_u}^{\text{addr}}}, sk_{\mathcal{S}} := \widetilde{sk_{\mathcal{EV}_u}^{\text{addr}}}, sk_{ots}, \\ pk_{\mathcal{R}} := pk_{\mathcal{CS}}, pk_{\mathcal{C}mt}, \text{info}_{pay}, \pi_u^{\text{auth}} \end{pmatrix} \rightarrow (tx_{ch}, \check{\sigma}_{ch})$**:**

1. Compute $R_1 := g_1^{sk_{ots}^{(1)} \cdot sk_{ots}^{(2)}}$, $R_2 := pk_{\mathcal{C}mt}^{sk_{ots}^{(1)} \cdot sk_{ots}^{(2)}}$, $R_3 := g_1^{sk_{ots}^{(2)}}$, $w_1 := sk_{ots}^{(1)} \cdot sk_{ots}^{(2)} - \mathbb{c}_{pay} \cdot sk_{ots}^{(1)} \cdot sk_{\mathcal{EV}_u}^{\text{addr}}$, $w_2 := sk_{ots}^{(2)} - \mathbb{c}_{pay} \cdot sk_{\mathcal{EV}_u}^{\text{addr}}$, and set $\mathbb{c}_{pay} := \text{H}_3(\text{info}_{pay} \parallel \widetilde{pk_{\mathcal{EV}_u}^{\text{addr}}} \parallel pk_{\mathcal{CS}} \parallel pk_{\mathcal{C}mt} \parallel R_1 \parallel R_2 \parallel R_3)$.
2. Compute $\sigma_{ots} := h_{pay,t}^{sk_{ots}^{(1)} + sk_{ots}^{(2)} \cdot t} \in \mathbb{G}_2$, $h_{pay,t} := \text{H}_2(\text{info}_{pay} \parallel t)$.
3. Generate $tx_{ch} := (\text{info}_{pay}, \pi_u^{\text{ch}} := (\mathbb{c}_{pay}, w_1, w_2, R_2), \pi_u^{\text{auth}}, \sigma_{ots}, \widetilde{pk_{\mathcal{EV}_u}^{\text{addr}}})$.
4. Sign tx_{ch} by invoking $\check{\sigma}_{ch} := \Sigma.\text{Sign}(sk_{\mathcal{EV}_u}^{\text{addr}}, tx_{ch})$ and then forward $(tx_{ch}, \check{\sigma}_{ch})$ to $\mathcal{C}mt$.

$\mathcal{CS}$ as spender:

Before executing the transaction, $\mathcal{EV}$ performs $\text{Gen}_{\mathcal{EV}_u}^{\text{addr}}(\cdot)$ to obtain $(\widetilde{pk_{\mathcal{EV}_u}^{\text{addr}}}, \widetilde{sk_{\mathcal{EV}_u}^{\text{addr}}}, \text{info}_{pay})$.

$\text{Trans}_{\mathcal{CS}}$ $\begin{pmatrix} pp, pk_{\mathcal{S}} := pk_{\mathcal{CS}}, sk_{\mathcal{S}} := sk_{\mathcal{CS}}, sk_{ots} := \emptyset, \\ pk_{\mathcal{R}} := \widetilde{pk_{\mathcal{EV}_u}^{\text{addr}}}, pk_{\mathcal{C}mt}, \text{info}_{pay}, \pi_u^{\text{auth}} \end{pmatrix} \rightarrow (tx_{dis}, \check{\sigma}_{dis})$**:**

1. Set $tx_{dis} := (\text{info}_{pay}, \widetilde{pk_{\mathcal{EV}_u}^{\text{addr}}}, pk_{\mathcal{CS}}, \pi_u^{\text{auth}})$.
2. Sign tx_{dis} by invoking $\check{\sigma}_{dis} := \Sigma.\text{Sign}(sk_{\mathcal{CS}}, tx_{dis})$, and send $(tx_{dis}, \check{\sigma}_{dis})$ to $\mathcal{C}mt$.

$\mathcal{C}mt$:

$\text{Verify}_{\mathcal{C}mt}^{\text{tx}}$($\mathcal{EV}$ as spender)$(pp, sk_{\mathcal{C}mt}, tx_{ch}, \check{\sigma}_{ch}) \rightarrow 0/1$**:**

1. Compute $Q_u' / Q_u^{sk_{\mathcal{C}mt}}$ to obtain $pk_{\mathcal{EV}_u}^{\text{addr}}$.
2. Verify the validity of $\check{\sigma}_{ch}$ by invoking $\Sigma.\text{Verify}(pk_{\mathcal{EV}_u}^{\text{addr}}, \check{\sigma}_{ch})$. If $\check{\sigma}_{ch}$ is invalid, abort.
3. Compute $R_1' := g_1^{w_1} \cdot Q_u^{\mathbb{c}_{pay}}, R_3' := g_1^{w_2} \cdot Q_u'^{\mathbb{c}_{pay}} \cdot pk_{\mathcal{C}mt}^{w_1} / R_2$, $\mathbb{c}_{pay}' := \text{H}_3(\text{info}_{pay} || \widetilde{pk_{\mathcal{EV}_u}^{\text{addr}}} || pk_{\mathcal{CS}} || pk_{\mathcal{C}mt} || R_1' || R_2 || R_3')$.
4. To chain the valid pending transactions, the $\mathcal{C}mt$ uses the PBFT consensus mechanism. If and only if at least two-thirds of the total commissioners approve the pending block, the consensus is reached and the transaction can be published. If the equation $\mathbb{c}_{pay} = \mathbb{c}_{pay}'$ and $e\left(vk_{ots}^{(1)} \cdot {vk_{ots}^{(2)}}^t, h_{pay,t}\right) = e(g_1, \sigma_{ots})$ hold, the algorithm returns 1. Otherwise, the algorithm returns 0, and then the $\mathcal{C}mt$ will send the abnormal message and tx_{ch} to the $\mathcal{RC}$.

$\text{Verify}_{\mathcal{C}mt}^{\text{tx}}$($\mathcal{CS}$ as spender)$(pp, sk_{\mathcal{C}mt}, tx_{dis}, \check{\sigma}_{dis}) \rightarrow 0/1$**:**

1. Invoke $\Sigma.\text{Verify}(pk_{\mathcal{CS}}, \check{\sigma}_{dis})$ to verify the validity of the signature $\check{\sigma}_{dis}$. If $\check{\sigma}_{dis}$ is invalid, abort.
2. Verify the validity of tx_{dis} by using PBFT. If and only if at least two-thirds of the total commissioners approve the pending block, the consensus is reached and the transaction can be published; otherwise, $\mathcal{C}mt$ will send the abnormal message and tx_{dis} to the $\mathcal{RC}$.

Remark 1: On receiving info_{req}, $\mathcal{CS}$ responds with the payment message $\text{info}_{pay} := (m_{bill} := (ID_{Order}, V_{pay}, \mathfrak{t}, \mathfrak{n}), \sigma_{\mathcal{CS}}, pk_{\mathcal{CS}})$, where ID_{Order} is the unique index of the order, $\sigma_{\mathcal{CS}} := \Sigma.\text{Sign}(sk_{\mathcal{CS}}, m_{bill})$, V_{pay} denotes the value of electricity calculated by (E_{req}, E_{bat}).

Figure 4. The process of transaction in the paid service.

V. Security and Performance Analysis

A. Security Analysis

This subsection briefly analysis that our construction meets the security requirements of Sec III.B.

Anonymous Authentication: In the authentication phase, $\mathcal{EV}$ presents blinding and verifiable credentials and tokens to a service provider based on XDH assumption and ZKP argument system to avoid leaking identifying information of $\mathcal{EV}$.

UCL: Relying on the DL problem, any PPT adversary cannot forge others' value of (B, φ_4) to pass $\text{Verify}_{\mathcal{CS}}^{\text{auth}}$ and link to the others' anonymous authentication info.

Accountability: Since an $\mathcal{EV}$'s accountability token is encrypted and embedded into the anonymous authentication

π_u^{auth}, $\mathcal{RC}$ can recover the $\mathcal{EV}$'s accountability token by its secret key $ask_{\mathcal{RC}}$, and then retrieve the identity of the $\mathcal{EV}$ from db_{Reg} according to the token. Moreover, based on the q-SDH assumption, no PPT adversary can forge an unaccountability token with a non-negligible probability.

Anonymous Payment: Similar to anonymous authentication, this property is guaranteed by the XDH assumption, one-time signature, and ZKP argument system. During the payment, no one can reveal the identifying information $pk_{\mathcal{EV}_u}^{\text{addr}}$ from the transaction tx_{ch}, except $\mathcal{Cmt}$.

Non-frameability: Based on the DL assumption, no PPT adversary can replace an innocent $\mathcal{EV}$ to generate a valid anonymous authentication or payment information with non-negligible probability.

Unforgeability of credentials: Based on the PS assumption, any PPT adversary who does not know the $\mathcal{Iss}$'s private key cannot forge valid attribute-based credentials.

Payment binding: In the payment phase, any PPT adversary cannot break the binding of payment and authentication by forging an OTS signature σ_{ots} based on PS assumption.

Mutual authentications: The security of mutual authentications depends on the unforgeability of authentication information from the interacting parties. In PAP, accountability, non-frameability, and unforgeability of credentials imply the unforgeability of authentication information generated by $\mathcal{EV}$. As for service providers, the unforgeability relies on the unforgeability of ordinary signature schemes, such as EdDSA[54].

Revocability: In PAP, the membership management mechanism is constructed by the CS method, and the security is based on the unforgeability of revocation tokens, specifically, based on PS assumption.

Min-AD: With the redactability of credentials, $\mathcal{EV}$ can present the necessary attributes to the service provider and keep unnecessary attributes private to arrive at Min-AD.

No double-spending attack: In the payment phase, the PBFT consensus can tolerate a third of all commissioners to be faulty. Hence, it must corrupt over a third of commissioners if an adversary intends to double-spend. Also, in our setting, since $\mathcal{Cmt}$ is the trusted party, there exists no one who can control such a number of commissioners.

B. Performance Analysis

In this subsection, we first evaluate the execution time of multiplication and the pairing operation over bilinear group in TABLE II on different configuration platforms for different entities and simulation experiments are carried out[2]. In this table, let $T_{\mathbb{G}_1}$, $T_{\mathbb{G}_2}$, $T_{\mathbb{G}_T}$ denote the time to complete a multiplication operation in $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_\text{T}$, respectively, and T_e represent the time to complete a pairing operation.

Next, we give out the theoretical performance analysis of each phase in PAP to present computation cost of different entities, shown in TABLE III, where $N_{a,k}$ is the number of attributes that $\mathcal{Iss}_k$ issues, $N_a \coloneqq \sum_k^{N_\mathcal{J}} N_{a,k}$ denotes the number of attributes that the aggregated credential contains, N_Γ is the number of attributes presented by $\mathcal{EV}$ in the authentication phase, $\overline{N_\Gamma} \coloneqq N_a - N_\Gamma$ is the number of remaining attributes after presenting, $N_\mathcal{J}$ stands for the number of issuers who issue credentials for the $\mathcal{EV}$, N_l denotes the depth of complete binary tree, N_{S_R} denotes the number of root nodes. In this table, we focus on the expensive operations in each algorithm, i.e., multiplication and pairing operation.

TABLE II
EACH ENTITY PLATFORM CONFIGURATION INFORMATION

Entity	Configuration	Overhead (ms)		Pairing Lib.
$\mathcal{EV}$	CPU: SA8155P	$T_{\mathbb{G}_1}$	0.55	-JPBC library[28]
		$T_{\mathbb{G}_2}$	1.52	
	RAM: 8.0 GB	$T_{\mathbb{G}_T}$	0.03	-Curve Type: BN[35]
	OS: Android 11	T_e	5.55	-Security Lv.: 128bits
$\mathcal{CS}$	CPU: Intel i7-10875	$T_{\mathbb{G}_1}$	0.08	-Base Field: 254bits
$\mathcal{Iss}$		$T_{\mathbb{G}_2}$	0.19	-Embedding Deg.: 12
$\mathcal{RC}$	RAM: 16.0 GB	$T_{\mathbb{G}_T}$	0.01	
$\mathcal{Cmt}$	OS: Ubuntu 18.04	T_e	0.37	

Note that, in the authentication phase, $\mathcal{EV}$ generates a zero-knowledge proof π_u^{auth} to prove its possession of valid tokens to $\mathcal{CS}$. However, the generation and verification of the proof involve a large number of pairing operations, which is not friendly to both sides of the authentication. Fortunately, our pairing operations, such as $e(\dot{g}_1, g_2)$, $e(\ddot{g}_1, g_2)$, $e(g_1, g_2)$, $e(h_{t,n^*}, g_2)$, $e(apk_{\mathcal{RC}}, pk_{\mathcal{RC},1})$, $e(apk_{\mathcal{RC}}, g_2)$, $e(h_{t,n^*}, pk_{\mathcal{RC},2})$, $e(h_{t,n^*}, pk_{\mathcal{RC},1})$, $e(h_{t,n^*}, pk_{\mathcal{RC},3})$, can be pre-computed to reduce the computational cost to $\star$. Note that in TABLE III, the cost of authentication phase is in the case of the linkability required, and if without UCL, the cost of $\text{Auth}_{\mathcal{EV}}$ is $7T_{\mathbb{G}_1} + (N_\mathcal{J} + 2\overline{N_\Gamma} + 2) \cdot T_{\mathbb{G}_2} + 7T_{\mathbb{G}_T} + 2T_e$ and $\text{Verify}_{\mathcal{CS}}^{\text{auth}}$ is $5T_{\mathbb{G}_1} + (N_\mathcal{J} + N_\mathcal{J} \cdot N_\Gamma) \cdot T_{\mathbb{G}_2} + 15T_{\mathbb{G}_T} + (6 + 2N_\mathcal{J}) \cdot T_e$.

TABLE III
THE COMPUTATIONAL COST OF OUR SCHEME

Phase	Algo.	Comp. Cost (Theo.)	Benchmark (ms)
Init.	$\text{GenK}_{\mathcal{RC}}$	$T_{\mathbb{G}_1} + 3T_{\mathbb{G}_2}$	0.65
	$\text{GenK}_{\mathcal{CS}}$	$T_{\mathbb{G}_1}$	0.08
	$\text{GenK}_{\mathcal{Iss}}$	$N_{a,k} \cdot T_{\mathbb{G}_1} + (N_{a,k}^2 + 1) \cdot T_{\mathbb{G}_2}$	1.11
	$\text{GenK}_{\mathcal{EV}}$	$T_{\mathbb{G}_1}$	0.55
	$\text{GenK}_{\mathcal{Cmt}}$	$T_{\mathbb{G}_1}$	0.08
Reg.	$\text{Reg}_{\mathcal{EV}}$	$T_{\mathbb{G}_1}$	0.55
	$\text{Reg}_{\mathcal{RC}}$	$(3 + 2N_l) \cdot T_{\mathbb{G}_1}$	3.44
	$\text{Verify}_{\mathcal{EV}}^{\text{tok}}$	$T_{\mathbb{G}_1} + 3T_{\mathbb{G}_2} + (2N_l + 2)T_e$	127.21
Iss.	$\text{Issue}_{\mathcal{EV}}$	$T_{\mathbb{G}_1}$	0.55
	$\text{Issue}_{\mathcal{Iss}}$	$(N_{a,k} + 5) \cdot T_{\mathbb{G}_1}$	0.56
	$\text{Verify}_{\mathcal{EV}}^{\text{cred}}$	$(N_a + 1) \cdot T_{\mathbb{G}_1} + 2T_e$	17.15
Auth.	$\text{Auth}_{\mathcal{EV}}$	$9T_{\mathbb{G}_1} + (N_\mathcal{J} + 2\overline{N_\Gamma} + 2) \cdot T_{\mathbb{G}_2} + 7T_{\mathbb{G}_T} + 2T_e{}^\star$	48.18
	$\text{Verify}_{\mathcal{CS}}^{\text{auth}}$	$5T_{\mathbb{G}_1} + (N_\mathcal{J} + N_\mathcal{J} \cdot N_\Gamma) \cdot T_{\mathbb{G}_2} + 15T_{\mathbb{G}_T} + (6 + 2N_\mathcal{J}) \cdot T_e{}^\star$	10.27
Pay.	$\text{Gen}_{\mathcal{EV}_\text{u}}^{\text{addr}}$	$2T_{\mathbb{G}_1}$	1.10
	$\text{Trans}_{\mathcal{EV}}$	$3T_{\mathbb{G}_1} + T_{\mathbb{G}_2}$	3.17
	$\text{Verify}_{\mathcal{Cmt}}^{\text{tx}}$	$7T_{\mathbb{G}_1} + 2T_e$	1.30
Acc.	$\text{Acc}_{\mathcal{RC}}$	$T_{\mathbb{G}_1}$	0.08
	$\text{Update}_{\mathcal{RC}}$	$N_{S_R} \cdot T_{\mathbb{G}_1}$	0.08

[2] https://github.com/PropersonCyber/PAP

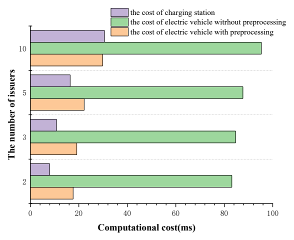
According to the analysis, we also conduct simulation experiments to test the actual performance. When the credential issued by each issuer contains different attributes, the computational cost of each phase is shown in TABLE III, where $\{N_{a,k} := 2\}_{k\in[N_{\mathcal{J}}]}$, $N_{\mathcal{J}} := 5$, $N_a := 10$, $N_\Gamma := 3$, $\overline{N_\Gamma} := 7$, $N_d := 20$, $N_{S_R} := 1$.

From TABLE III, we observe that the cost of $\text{Auth}_{\mathcal{EV}}(\cdot)$ is related to the number of issuers and presented attributes that access policy requires, and the execution of the algorithm is more frequently than other algorithm in PAP. From the view of the number of issuers $N_{\mathcal{J}}$, Figure 5(a) presents the $N_{\mathcal{J}}$-changing trend of computation cost of $\text{Auth}_{\mathcal{EV}}(\cdot)$ in two cases:

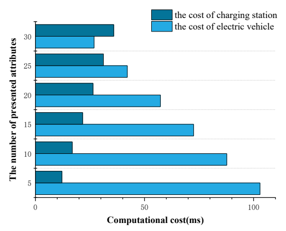
(1) *Case 1:* Without preprocessing of redactable signatures, the computational cost straightforwardly grows linear, and the computation complexity is $\mathcal{O}(\overline{N_\Gamma} + N_{\mathcal{J}})$.

(2) *Case 2:* In $\text{Auth}_{\mathcal{EV}}(\cdot)$, $\mathcal{EV}$ can compute $\{\tilde{\sigma}_{k,1}, \tilde{\sigma}_{k,2}\}_{k\in N_{\mathcal{J}}}$, $\sigma_1', \sigma_2', \tilde{\sigma}_1$ in advance if the access policy Γ is known. Consequently, the computational cost is reduced from * to $6T_{\mathbb{G}_1} + N_{\mathcal{J}} \cdot T_{\mathbb{G}_2} + 7T_{\mathbb{G}_T} + 2T_e$, the computation complexity changes to $\mathcal{O}(N_{\mathcal{J}})$.

Furthermore, Figure 5(a) also presents the corresponding cost of $\text{Verify}^{\text{auth}}_{\mathcal{CS}}(\cdot)$ that raises as the number of issuers increases.



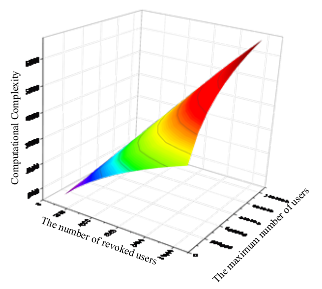
(a) The effect of different numbers of issuers on the computational cost when $N_\Gamma = 10$ attributes need to be presented according to a fixed access policy.



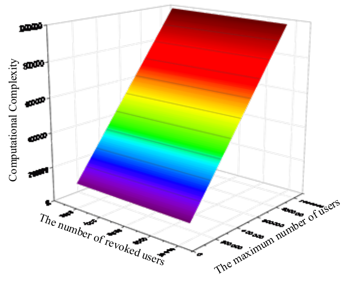
(b) The effect of different numbers of presented attributes on the computational cost when the aggregated credential contains credentials from $N_{\mathcal{J}} = 5$ issuers.

Figure 5. The analysis of the computational cost in different authentication case with the assumption that credential contains thirty attributes.

From the view of the attributes presented, we have an interesting result, as shown in Figure 5(b). Assuming that $\mathcal{EV}$'s credential is from a fixed amount of issuers, it is not hard to see the fact holds due to the number of redaction operations decreasing as the number of presented attributes increases in the authentication phase. On the contrary, the computation cost of $\text{Verify}^{\text{auth}}_{\mathcal{CS}}(\cdot)$ arises normally.



(a) CS method



(b) Traditional method

Figure 6. Comparison between the CS method and the traditional method

We next analyze the performance of another essential algorithm, i.e., $\text{Update}_{\mathcal{RC}}(\cdot)$. The computation complexity of the algorithm is $\mathcal{O}(r_{\mathcal{EV}} \cdot \log\frac{N_{\mathcal{EV}}}{r_{\mathcal{EV}}})$ as explained in Sec II.C, where $N_{\mathcal{EV}}$ is the maximum number of $\mathcal{EV}$s, $r_{\mathcal{EV}}$ is the number of revoked $\mathcal{EV}$s. Figure 6 illustrates an emulation of comparing the CS method[25] with the traditional method based on revocation list[21] (whose computation complexity is $\mathcal{O}(N_{\mathcal{EV}} - r_{\mathcal{EV}})$) for the updating operation, where we set $N_{\mathcal{EV}} \in [10^4, 10^6]$ and $r_{\mathcal{EV}} \in [0, 1000]$. The number of revocation tokens that need to be updated for the CS method and the traditional method is shown in Figure 6(a) and Figure 6(b), respectively. As seen in Figure 6, ours is more efficient.

VI. Related work

As for anonymous authentication in V2G networks, there has existed a huge amount of work, such as [5]-[12], [43]-[54]. Unfortunately, none of these schemes satisfy the security requirements of accountability, revocability, anonymous payment, and Min-AD at the same time. Additionally, in the case of anonymous payment, to our best knowledge, although the general decentralized payment schemes[13]-[15] can support a certain degree of anonymity, they still have the limitation of high computational cost and the lack of accountability. Besides, existing decentralized payment schemes for V2G[36]-[41] either fail to provide anonymous authentication or are computationally expensive.

Finally, we compare the proposed scheme to other existing schemes[5], [7], [8], [43] as TABLE IV, where the security and privacy requirements are defined in Section III.B. In TABLE IV, the schemes all satisfy the requirements of anonymous authentication, non-frameability, and unforgeability of credentials. However, none of these counterparts consider the property of accountability, anonymous payment, revocability, and Min-AD, which are also essential security requirements in practical V2G networks. Compared with these schemes, our scheme ensures all of the proposed security requirements, which can better protect $\mathcal{EV}s'$ privacy.

TABLE IV
Comparison of different schemes

Security & Privacy Requirements	Hou W [7]	Jiang Z [8]	Gope [43]	Zhang [5]	Our scheme
Ano.Auth.	✓	✓	✓	✓	✓
Non-frameability	✓	✓	✓	✓	✓
Accountability	✓	×	×	×	✓
Anon. Pay.	×	×	×	×	✓
Unfor.	✓	✓	✓	✓	✓
Revocability	✓	×	×	×	✓
Min-AD	×	×	×	×	✓
UCL	×	×	×	×	✓

VII. Conclusion

To protect physical privacy, the capability to achieve authentication and payment in privacy-preserving manner is crucial in V2G networks. In this work, we introduced a new privacy framework PAP for V2G networks that supports anonymous authentication and payment, and features security properties of accountability, revocability, Min-AD, UCL, non-frameability, and payment binding, etc. This enabled us to come up with a radically different design which offers a new, practically relevant balance between privacy and efficiency compared to all previous approaches.

VIII. Acknowledgment

This study was supported by Natural Science Foundation of Liaoning Province, China under Grant 2022-MS-438, Scientific Study Project for Institutes of Higher Learning, Ministry of Education, Liaoning Province under Grant LJKFZ20220184, Science and Technology Projects in Liaoning Province under Grant 2022JH1/10800085.

References

[1] M. Faheem et al., "Smart grid communication and information technologies in the perspective of Industry 4.0: Opportunities and challenges," Comput. Sci. Rev., vol. 30, pp. 1–30, Nov. 2018.

[2] A. Kumari, S. Tanwar, S. Tyagi, N. Kumar, M. S. Obaidat, and J. J. P. C. Rodrigues, "Fog Computing for Smart Grid Systems in the 5G Environment: Challenges and Solutions," IEEE Wirel. Commun., vol. 26, no. 3, pp. 47–53, Jun. 2019.

[3] G. Bansal, N. Naren, V. Chamola, B. Sikdar, N. Kumar, and M. Guizani, "Lightweight Mutual Authentication Protocol for V2G Using Physical Unclonable Function," IEEE Trans. Veh. Technol., vol. 69, no. 7, pp. 7234–7246, Jul. 2020.

[4] S. M. S. Hussain, T. S. Ustun, P. Nsonga, and I. Ali, "IEEE 1609 WAVE and IEC 61850 Standard Communication Based Integrated EV Charging Management in Smart Grids," IEEE Trans. Veh. Technol., vol. 67, no. 8, pp. 7690–7697, Aug. 2018.

[5] Zhang Y, Zou J, Guo R. Efficient privacy-preserving authentication for V2G networks. Peer-to-Peer Networking and Applications, 2021.

[6] Y. Zhang, J. Zou, and R. Guo, "Efficient privacy-preserving authentication for V2G networks," Peer--Peer Netw. Appl., vol. 14, no. 3, Art. no. 3, May 2021.

[7] Hou W, Sun Y, Li D, et al. Lightweight and Privacy-Preserving Charging Reservation Authentication Protocol for 5G-V2G. IEEE Transactions on Vehicular Technology, 2023.

[8] Jiang Z, Zhou Z, Xiong L, et al. An Efficient Lightweight Anonymous Authentication Scheme for V2G Using Physical Unclonable Function. 2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall). IEEE, 2021.

[9] Kilari V T, Yu R, Misra S, et al. Robust revocable anonymous authentication for vehicle to grid communications. IEEE Transactions on Intelligent Transportation Systems, 2020.

[10] Lin C, He D, Zhang H, et al. Privacy-enhancing decentralized anonymous credential in smart grids. Computer Standards & Interfaces, 2021.

[11] Kaur K, Garg S, Kaddoum G, et al. A secure, lightweight, and privacy-preserving authentication scheme for V2G connections in smart grid. IEEE INFOCOM 2019-IEEE conference on computer communications workshops (INFOCOM WKSHPS). IEEE, 2019.

[12] Parameswarath R P, Gope P, Sikdar B. A Privacy-Preserving Authenticated Key Exchange Protocol for V2G Communications Using SSI. IEEE Transactions on Vehicular Technology, 2023.

[13] E. B. Sasson et al., "Zerocash: Decentralized anonymous payments from bitcoin," in Proc. IEEE Symp. Secur. Privacy, Berkeley, CA, USA, May 2014, pp. 459–474.

[14] A. Kosba, A. Miller, E. Shi, Z. Wen and C. Papamanthou, "Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts," 2016 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 2016, pp. 839-858.

[15] S. Bowe, A. Chiesa, M. Green, I. Miers, P. Mishra and H. Wu, "ZEXE: Enabling Decentralized Private Computation," 2020 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 2020, pp. 947-964.

[16] M. Castro and B. Liskov, "Practical byzantine fault tolerance," in Proc. 3rd USENIX Symp. Operating Syst. Design Implement. (OSDI), M. I. Seltzer and P. J. Leach, Eds. New Orleans, LA, USA: USENIX Association, Feb. 1999, pp. 173–186.

[17] Zaverucha G M, Stinson D R. Short one-time signatures[J]. Advances in Mathematics of Communications, 2011, 5(3): 473-488.

[18] J. Camenisch and T. Groß, "Efficient Attributes for Anonymous Credentials," ACM Trans. Inf. Syst. Secur., vol. 15, no. 1, pp. 1–30, Mar. 2012.

[19] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. Foundations and Trends in Theoretical Computer Science, 9(3 4):211–407, 2014.

[20] Brickell E, Li J. A pairing-based DAA scheme further reducing TPM resources[C]//International Conference on Trust and Trustworthy Computing. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010: 181-195.

[21] D. Boneh and H. Shacham, "Group signatures with verifier-local revocation," in Proceedings of the 11th ACM conference on Computer and communications security - CCS '04, Washington DC, USA, 2004, p.168.

[22] Rathee D, Policharla G V, Xie T, et al. Zebra: Anonymous credentials with practical on-chain verification and applications to kyc in defi[J]. Cryptology ePrint Archive, 2022.

[23] Boneh, D., Bünz, B., Fisch, B. (2019). Batching Techniques for Accumulators with Applications to IOPs and Stateless Blockchains. In: Boldyreva, A., Micciancio, D. (eds) Advances in Cryptology – CRYPTO 2019. CRYPTO 2019. Lecture Notes in Computer Science (), vol 11692. Springer, Cham.

[24] Naor D, Naor M, Lotspiech J. Revocation and tracing schemes for stateless receivers//Crypto. Annual International Cryptology Conference. LNCS. Berlin: Springer, 2001: 41-62.

[25] Yue X, Zeng S, Wang X, et al. A practical privacy-preserving communication scheme for CAMs in C-ITS. Journal of Information Security and Applications, 2022, 65: 103103.

[26] Sanders, O. (2020). Efficient Redactable Signature and Application to Anonymous Credentials. In: Kiayias, A., Kohlweiss, M., Wallden, P., Zikas, V. (eds) Public-Key Cryptography – PKC 2020. Lecture Notes in Computer Science(), vol 12111. Springer, Cham.

[27] Josefsson S, Liusvaara I. Edwards-curve digital signature algorithm (EdDSA)[R]. 2017.

[28] Angelo D, C, Vincenzo jPBC. Java pairing based cryptography. In: Proceedings of the 16th IEEE Symposium on Computers and Communications, ISCC 2011. IEEE; 2011. p. 850–5.

[29] McCurley K S.The discrete logarithm problem//Proc. of Symp. inApplied Math. 1990, 42: 49-74.

[30] Pointcheval D, Sanders O. Short randomizable signatures[C]//Topics in Cryptology-CT-RSA 2016: The Cryptographers' Track at the RSA Conference 2016, San Francisco, CA, USA, February 29-March 4, 2016, Proceedings. Springer International Publishing, 2016: 111-126.

[31] Boneh D, Boyen X.Short signatures without random oracles and the SDH assumption in bilinear groups. Journal of cryptology,2008,21(2): 149-177.

[32] Fiat A and Shamir A. How to prove yourself: practical solutions to identification and signature problems. In: Proceedings of the conference on the theory and application of cryptographic techniques, Santa Barbara, CA, 11–15 August 1986, pp.186–194. Berlin: Springer.

[33] J. Shen, T. Zhou, D. He, Y. Zhang, X. Sun, and Y. Xiang, "Block design based key agreement for group data sharing in cloud computing," IEEE Trans. Depend. Sec. Comput., vol. 16, no. 6, pp. 996–1010, Nov. 2019.

[34] Rathee D, Policharla G V, Xie T, et al. Zebra: Anonymous credentials with practical on-chain verification and applications to kyc in defi[J]. Cryptology ePrint Archive, 2022.

[35] Barreto P, Naehrig M. Pairing-friendly elliptic curves of prime order. In: Proceedings of the 12th Int Workshop on Selected Areas in Cryptography. LNCSSpringerVerlag, Berlin Heidelberg; 2006. p. 319–31.

[36] Liu H, Zhang Y, Zheng S, et al. Electric vehicle power trading mechanism based on blockchain and smart contract in V2G network[J]. IEEE Access, 2019, 7: 160546-160558.

[37] C. Lin, D. He, X. Huang, M. K. Khan and K. -K. R. Choo, "DCAP: A Secure and Efficient Decentralized Conditional Anonymous Payment System Based on Blockchain," in IEEE Transactions on Information Forensics and Security, vol. 15, pp. 2440-2452, 2020.

[38] Z. Wan, T. Zhang, W. Liu, M. Wang and L. Zhu, "Decentralized Privacy-Preserving Fair Exchange Scheme for V2G Based on Blockchain," in IEEE Transactions on Dependable and Secure Computing, vol. 19, no. 4, pp. 2442-2456, 1 July-Aug. 2022.

[39] Gao F, Zhu L, Shen M, et al. A blockchain-based privacy-preserving payment mechanism for vehicle-to-grid networks[J]. IEEE network, 2018, 32(6): 184-192.

[40] Iqbal A, Rajasekaran A S, Nikhil G S, et al. A secure and decentralized blockchain based EV energy trading model using smart contract in V2G network[J]. IEEE Access, 2021, 9: 75761-75777.

[41] Radi E M, Lasla N, Bakiras S, et al. Privacy-preserving electric vehicle charging for peer-to-peer energy trading ecosystems[C]//ICC 2019-2019 IEEE International Conference on Communications (ICC). IEEE, 2019: 1-6.

[42] Garman, C., Green, M., Miers, I. (2017). Accountable Privacy for Decentralized Anonymous Payments. In: Grossklags, J., Preneel, B. (eds) Financial Cryptography and Data Security. FC 2016. Lecture Notes in Computer Science(), vol 9603. Springer, Berlin, Heidelberg.

[43] P. Gope and B. Sikdar, “An Efficient Privacy-Preserving Authentication Scheme for Energy Internet-Based Vehicle-to-Grid Communication,” IEEE Trans. Smart Grid, vol. 10, no. 6, Art. no. 6, Nov. 2019.

[44] Mona Jamjoom, Hussein Abulkasim, Safia Abbas, "Lightweight Authenticated Privacy-Preserving Secure Framework for the Internet of Vehicles", Security and Communication Networks, vol. 2022, Article ID 6573060, 11 pages, 2022.

[45] Chendong H, Feng W and Zhongming H. (2022). A Lightweight Anonymous Key Agreement Scheme in V2G Networks 2022 2nd International Conference on Frontiers of Electronics, Information and Computation Technologies (ICFEICT).

[46] D. Gabay, K. Akkaya and M. Cebe, "Privacy-Preserving Authentication Scheme for Connected Electric Vehicles Using Blockchain and Zero Knowledge Proofs," in IEEE Transactions on Vehicular Technology, vol. 69, no. 6, pp. 5760-5772, June 2020.

[47] N. Xi, W. Li, L. Jing and J. Ma, "ZAMA: A ZKP-Based Anonymous Mutual Authentication Scheme for the IoV," in IEEE Internet of Things Journal, vol. 9, no. 22, pp. 22903-22913, 15 Nov.15, 2022.

[48] Belkaaloul A, Bensaber B A. Anonymous authentication protocol for efficient communications in vehicle to grid networks[C]//2021 IEEE Symposium on Computers and Communications (ISCC). IEEE, 2021: 1-5.

[49] Xia Z, Fang Z, Gu K, et al. Effective charging identity authentication scheme based on fog computing in V2G networks[J]. Journal of Information Security and Applications, 2021, 58: 102649.

[50] Zhang Y, Zou J, Guo R. Efficient privacy-preserving authentication for V2G networks[J]. Peer-to-Peer Networking and Applications, 2021, 14(3): 1366-1378.

[51] Sharma G, Joshi A M, Mohanty S P. An efficient physically unclonable function based authentication scheme for V2G network[C]//2021 IEEE International Symposium on Smart Electronic Systems (iSES). IEEE, 2021: 421-425.

[52] Kilari V T, Yu R, Misra S, et al. Robust revocable anonymous authentication for vehicle to grid communications[J]. IEEE Transactions on Intelligent Transportation Systems, 2020, 21(11): 4845-4857.

[53] Wang Q, Ou M, Yang Y, et al. Conditional privacy-preserving anonymous authentication scheme with forward security in vehicle-to-grid networks[J]. IEEE Access, 2020, 8: 217592-217602.

[54] Rajasekaran A S, Maria A, Al-Turjman F, et al. ABRIS: Anonymous blockchain based revocable and integrity preservation scheme for vehicle to grid network[J]. Energy Reports, 2022, 8: 9331-9343.

[55] S. D. Galbraith, K. G. Paterson, and N. P. Smart, “Pairings for cryptographers,” Discrete Appl. Math., vol. 156, no. 16, pp. 3113–3121, Sep. 2008.

[56] Pointcheval D, Sanders O. Reassessing security of randomizable signatures//CT-RSA. Cryptographers’ Track at the RSA Conference. LNCS. Berlin: Springer, 2018: 319-338.

[57] Boneh D. The decision diffie-hellman problem//Algorithmic Number Theory: Third International Symposiun, ANTS-III Portland, Oregon, USA, June 21–25, 1998 Proceedings. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006: 48-63.

[58] Qiao Zi-Rui, Yang Qi-Liang, Zhou Yan-Wei, et al. An Efficient Authentication key agreement Protocol with Provable Security for VANET. Chinese Journal of computers, 2023, 46(05): 929-944.

IX. Security Analysis

This section presents the security model and security proofs of PAP.

A. Security Model

We define the security properties for our scheme on unforgeability of credential, anonymity, payment binding, non-frameability and accountability using the following Oracle with two lists: L_H and L_C, which contains the identities of honest $\mathcal{EV}$s and corrupted $\mathcal{EV}$s, respectively. In addition, we define a table T_{tk} that stores the public-private key of $\mathcal{EV}$ and the corresponding tokens, T_{Addr} stores the wallet address of each $\mathcal{EV}$, T_{cred} stores the credential of each $\mathcal{EV}$, T_{Auth} stores the authentication information of each $\mathcal{EV}$, T_{tx} stores the transaction information. All lists and tables are initialized to be empty.

-$\mathcal{O}_{L_H}(j)$: on the input of an identity j, if j already exists (i.e.,$j \in L_H \cup L_C$), this oracle outputs $\perp$. Otherwise, it returns $(pk_{\mathcal{EV}_j}, sk_{\mathcal{EV}_j})$ and adds j to L_H.

-$\mathcal{O}_{L_C}(j)$: on the input of j, if it does not exist for $j \notin L_H$, this oracle adds j to L_C. Otherwise, it removes j from L_H and add it to L_C, then returns $(pk_{\mathcal{EV}_j}, sk_{\mathcal{EV}_j})$ and all the associated credentials.

- $\mathcal{O}_{Addr}$: on the input of an identity j , it generates $(pk^{\mathrm{addr}}_{\mathcal{EV}_j}, sk^{\mathrm{addr}}_{\mathcal{EV}_j})$.

-$\mathcal{O}_{\mathrm{H}}$: on input of the hash function, it outputs corresponding hash value.

- $\mathcal{O}_{\mathrm{REG}}(j)$: on the input of the identity j , it runs $\mathrm{Reg}_{\mathcal{RC}}\begin{pmatrix} pp, sk_{\mathcal{RC}}, t, \\ pk^*_{\mathcal{EV}_j}, \pi^{\mathrm{tok}}_j \end{pmatrix}$ to output token tk_j.

-$\mathcal{O}_{\mathrm{ISS}}(pp, pk_{\mathcal{EV}_j})$: on the input of an identity j, and a set of attributes $\{\mathfrak{a}_i\}_{i=1}^{n}$, if $j \notin L_H$, this oracle outputs $\perp$. Otherwise, it generates $cred_j$ and $\{\mathfrak{a}_{k,i}\}$ and stores then in T_{cred}.

-$\mathcal{O}_{\mathrm{AUTH}}(j, \{\mathfrak{a}_{k,i}\}, \Gamma)$: on input an identity j, the attribute set $\{\mathfrak{a}_{k,i}\}$ and access policy Γ, if $j \notin L_H$, outputs $\perp$. Otherwise, it outputs $(\pi^{\mathrm{auth}}_j, \widehat{cred_j})$.

-$\mathcal{O}_{\mathrm{ACC}}(j, \pi^{\mathrm{auth}}_j)$: on input an identity j and its proof π^{auth}_j outputs $tk^{\mathrm{acc}*}_j$ if $j \in L_H$ and $\perp$ otherwise.

-$\mathcal{O}_{\mathrm{VER}}(tx, j)$: on input the identity j and the transaction tx, if $j \in L_C$, it returns $\perp$. Otherwise, it outputs the wallet address $pk^{\mathrm{addr}}_{\mathcal{EV}_j}$.

Definition 7 (Authentication anonymity): The authentication scheme is anonymous if $\mathrm{Adv}^{\mathrm{ano}} = |\Pr[\mathrm{EXP}^{\mathrm{ano-1}}_{\mathcal{A}}(1^\lambda) = 1] - \Pr[\mathrm{EXP}^{\mathrm{ano-0}}_{\mathcal{A}}(1^\lambda) = 1]|$ is negligible for any polynomial-time adversary $\mathcal{A}$.

The $\mathrm{EXP}^{\mathrm{ano-b}}_{\mathcal{A}}$ experiment of authentication anonymity is defined as follows:

```
$\mathrm{EXP}^{\mathrm{ano-b}}_{\mathcal{A}}(1^\lambda)$:
$pp \leftarrow \mathrm{Setup}(1^\lambda)$
$(pk_{\mathcal{RC}}, sk_{\mathcal{RC}}, apk_{\mathcal{RC}}, ask_{\mathcal{RC}}) \leftarrow \mathrm{GenK}_{\mathcal{RC}}(pp)$
$b \leftarrow \mathcal{A}^{\mathcal{O}_{EV}, \mathcal{O}_{L_C}, \mathcal{O}_{CH}, \mathcal{O}_{REG}, \mathcal{O}_{ACC}}(pp, pk_{\mathcal{RC}}, sk_{\mathcal{RC}}, apk_{\mathcal{RC}}, pk_{\mathcal{J}ss_k}, sk_{\mathcal{J}ss_k})$
return $b$
```

Definition 8 (Unforgeability of credential): The credential scheme is unforgeable if $\mathrm{Adv}^{\mathrm{uf}} = |\Pr[\mathrm{EXP}^{\mathrm{uf}}_{\mathcal{A}}(1^\lambda) = 1]|$ is negligible for any polynomial-time adversary $\mathcal{A}$.

The $\mathrm{EXP}^{\mathrm{uf}}_{\mathcal{A}}$ experiment of unforgeability is defined as follows:

```
$\mathrm{EXP}^{\mathrm{uf}}_{\mathcal{A}}(1^\lambda)$:
$pp \leftarrow \mathrm{Setup}(1^\lambda)$
$(pk_{\mathcal{JSS}_k}, sk_{\mathcal{JSS}_k}) \leftarrow \mathrm{GenK}_{\mathcal{JSS}}(pp)$
$\hat{\sigma}^{(N_\mathcal{J})}_u \leftarrow \mathcal{A}^{\mathcal{O}_{\mathrm{ISS}}}(pp, pk_{\mathcal{EV}_u}, sk_{\mathcal{EV}_u}, \{\mathfrak{a}_{k,i}\}, \hat{\sigma}^{(k)}_u, \pi^{\mathrm{cred}}_u)$
If $1 \leftarrow \mathrm{Verify}^{\mathrm{cred}}_{\mathcal{EV}}\left(pp, \{pk_{\mathcal{JSS}_k}\}_{k \in [N_\mathcal{J}]}, sk_{\mathcal{EV}_u}, \hat{\sigma}^{(N_\mathcal{J})}_u\right)$
return 1; Else return 0
```

Definition 9 (Accountability) Our privacy-preserving authentication scheme guarantees accountability if $\mathrm{Adv}^{\mathrm{acc}} = |\Pr[\mathrm{EXP}^{\mathrm{acc}}_{\mathcal{A}}(1^\lambda) = 1]|$ is negligible for any polynomial-time adversary $\mathcal{A}$.

The $\mathrm{EXP}^{\mathrm{acc}}_{\mathcal{A}}$ experiment of the accountability is defined as follows:

```
$\mathrm{EXP}_{\mathcal{A}}^{\mathrm{acc}}(1^{\lambda})$:
$pp \leftarrow \mathrm{Setup}(1^{\lambda})$
$(pk_{\mathcal{RC}}, sk_{\mathcal{RC}}, apk_{\mathcal{RC}}, ask_{\mathcal{RC}}) \leftarrow \mathrm{GenK}_{\mathcal{RC}}(pp)$
$(\pi_j^{\mathrm{auth}}, \widehat{cred}_j) \leftarrow \mathcal{A}^{\mathcal{O}_{L_H}, \mathcal{O}_{L_C}, \mathcal{O}_{ACC}, \mathcal{O}_{REG}, \mathcal{O}_{ISS}}(pp, pk_{\mathcal{RC}}, ask_{\mathcal{RC}}, apk_{\mathcal{RC}})$
If $1 \leftarrow \mathrm{Verify}_{\mathcal{CS}}^{\mathrm{auth}}\left(pp, pk_{\mathcal{EV}_j}, pk_{\mathcal{ISS}_k}, \pi_j^{\mathrm{auth}}, \widehat{cred}_j, \{\mathfrak{a}_i\}_{i \in I_\Gamma}\right) \wedge tk_j^{\mathrm{acc}} \leftarrow$
$\mathrm{Acc}_{\mathcal{RC}}(pp, \pi_j^{\mathrm{auth}}, ask_{\mathcal{RC}}) \wedge tk_j^{\mathrm{acc}} \in \mathrm{db}_{\mathrm{Reg}}$
return 1; Else return 0
```

Definition 10 (Payment anonymity): Our scheme satisfies payment anonymity if $\mathrm{Adv}^{\mathrm{pa}} = |\Pr[\mathrm{EXP}_{\mathcal{A}}^{\mathrm{pa}-1}(1^{\lambda}) = 1] - \Pr[\mathrm{EXP}_{\mathcal{A}}^{\mathrm{pa}-0}(1^{\lambda}) = 1]|$ is negligible for any polynomial-time adversary $\mathcal{A}$:

The $\mathrm{EXP}_{\mathcal{A}}^{\mathrm{pa-b}}$ experiment of payment anonymity is defined as follows:

```
$\mathrm{EXP}_{\mathrm{A}}^{\mathrm{pa-b}}(1^{\lambda})$:
$pp \leftarrow \mathrm{Setup}(1^{\lambda})$
$(pk_{\mathcal{CS}}, sk_{\mathcal{CS}}) \leftarrow \Sigma.\mathrm{Setup}(1^{\lambda})$
$b \leftarrow \mathcal{A}^{\mathcal{O}_{Addr}, \mathcal{O}_{\mathrm{VER}}, \mathcal{O}_{\mathrm{CH}}}(pp, pk_{\mathcal{EV}_j}, pk_{\mathcal{C}mt}, pk_{\mathcal{CS}}, \mathrm{info}_{pay})$
return $b$
```

Definition 11 (Non-frameability): Our scheme satisfies non-frameability if $\mathrm{Adv}^{\mathrm{nf}} = |\Pr[\mathrm{EXP}_{\mathcal{A}}^{\mathrm{nf}}(1^{\lambda}) = 1]$ is negligible for any polynomial-time adversary $\mathcal{A}$:

The $\mathrm{EXP}_{\mathcal{A}}^{\mathrm{nf}}$ experiment of non-frameability is defined as follows:

```
$\mathrm{EXP}_{\mathcal{A}}^{\mathrm{nf}}(1^{\lambda})$:
$pp \leftarrow \mathrm{Setup}(1^{\lambda})$
$(pk_{\mathcal{RC}}, sk_{\mathcal{RC}}, apk_{\mathcal{RC}}, ask_{\mathcal{RC}}) \leftarrow \mathrm{GenK}_{\mathcal{RC}}(pp)$
$(pk_{\mathcal{ISS}_k}, sk_{\mathcal{ISS}_k}) \leftarrow \mathrm{GenK}_{\mathcal{ISS}}(pp)$
$(\pi_j^{\mathrm{auth}}, \widehat{cred}_j) \leftarrow$
$\mathcal{A}^{\mathcal{O}_{\mathrm{EV}}, \mathcal{O}_{L_C}, \mathcal{O}_{\mathrm{REG}}, \mathcal{O}_{\mathrm{ISS}}, \mathcal{O}_{\mathrm{AUTH}}}(pp, pk_{\mathcal{RC}}, sk_{\mathcal{RC}}, apk_{\mathcal{RC}}, ask_{\mathcal{RC}})$
If $1 \leftarrow \mathrm{Verify}_{\mathcal{CS}}^{\mathrm{auth}}\begin{pmatrix} pp, pk_{\mathcal{EV}_j}, pk_{\mathcal{ISS}_k}, \\ \pi_j^{\mathrm{auth}}, \widehat{cred}_j, \{\mathfrak{a}_i\}_{i \in I_\Gamma} \end{pmatrix}$
return 1; Else return 0
```

Definition 12 (Payment binding): Our scheme offers payment binding if $\mathrm{Adv}^{\mathrm{bind}} = |\Pr[\mathrm{EXP}_{\mathcal{A}}^{\mathrm{bind}}(1^{\lambda}) = 1]$ is negligible for any polynomial-time adversary $\mathcal{A}$:

The $\mathrm{EXP}_{\mathcal{A}}^{\mathrm{bind}}$ experiment of payment binding is defined as follows:

```
$\mathrm{EXP}_{\mathcal{A}}^{\mathrm{bind}}(1^{\lambda})$:
$pp \leftarrow \mathrm{Setup}(1^{\lambda})$
$(pk_{\mathcal{C}mt}, sk_{\mathcal{C}mt}) \leftarrow \mathrm{GenK}_{\mathcal{C}mt}(pp)$
$tx \leftarrow \mathcal{A}^{\mathcal{O}_{L_H}, \mathcal{O}_{L_C}, \mathcal{O}_{\mathrm{GEN}}, \mathcal{O}_{\mathrm{TX}}, \mathcal{O}_{\mathrm{AUTH}}}(pp, pk_{\mathcal{C}mt}, pk_{\mathcal{CS}}, \mathrm{info}_{pay})$
If $1 \leftarrow \mathrm{Verify}_{\mathcal{C}mt}^{\mathrm{tx}}(pp, sk_{\mathcal{C}mt}, tx)$
return 1; Else return 0.
```

B. Security Proof

Theorem 1. *Under the XDH assumption, the PAP scheme is anonymous in authentication phase. More specifically, if there is a PPT adversary $\mathcal{A}$ that succeeds with a non-negligible probability to break the authentication anonymity(Definition 7), then there is a polynomial-time algorithm $\mathcal{S}$ that solves the XDH problem with a non-negligible probability.*

Lemma 1.1 *Under the XDH assumption, no PPT adversary $\mathcal{A}$ can break authentication anonymity without linkability with a non-negligible probability.*

Proof. Suppose $\mathcal{A}$ can break the authentication anonymity of our scheme with non-negligible probability. We can build a polynomial-time simulator $\mathcal{S}$ that break XDH assumption as below. $\mathcal{S}$ is given a tuple $(u, v := u^a, w := u^b, z)$, where $u \in \mathbb{G}_1$, $a, b \in \mathbb{Z}_p$, and either $z = u^{ab}$ or z is a randomness in $\mathbb{G}_1$. $\mathcal{S}$ decides which z was given by interacting $\mathcal{A}$.

Setup. $\mathcal{S}$ generates the public system parameter $pp := (\mathbb{G}_1, \mathbb{G}_2, g_1, \dot{g}_1, \ddot{g}_1, g_2, e, \mathrm{H}_1, \mathrm{H}_2, \mathrm{H}_3)$ as usual. In PAP, $\mathcal{RC}$ can be viewed as two entities: an issuer of tokens and accountability authority, where issuing key pair is $(pk_{\mathcal{RC}}, sk_{\mathcal{RC}})$ and accountability key pair is $(ask_{\mathcal{RC}} := a, apk_{\mathcal{RC}} := u^a)$. $\mathcal{A}$ can corrupt entities other than accountability authority to obtain the key pair. Subsequently, $\mathcal{S}$ answers the oracle queries as follows:

-$\mathcal{O}_{\mathrm{H}}$: $\mathcal{S}$ can response to the hash queries for H_1,H_2,H_3 as follows:

1) H_1: only need to ensure its collision resistance.

2) H_2: only need to ensure its collision resistance.

3) H_3: if the input has not been queried before, return $\mathbb{c} \leftarrow \mathbb{Z}_p$, otherwise return the previously queried result.

-$\mathcal{O}_{\mathrm{REG}}$: $\mathcal{A}$ requests to register an identity j. If $j \notin L_H$, $\mathcal{S}$ sets $L_H \leftarrow L_H \cup \{j\}$, and then generates its key pair $(pk_{\mathcal{EV}_j}, sk_{\mathcal{EV}_j})$ and $\mathcal{S}$ receives token $tk_{j'}$ from $\mathcal{A}$. If $j \in L_H$, $\mathcal{S}$ interacts with $\mathcal{A}$ to obtain tk_j.

-$\mathcal{O}_{EV}$: on the input of an identity j, it returns $(tk_j, sk_{\mathcal{EV}_j})$ and add it to T_{tk}.

-$\mathcal{O}_{L_C}$: on the input of an identity j and the public key $pk_{\mathcal{EV}_j}^*$, if $j \notin L_H \cup L_C$, it sets $pk_{\mathcal{EV}_j}^* := pk_{\mathcal{EV}_j}$ and add j to L_C, otherwise returns $\perp$.

-$\mathcal{O}_{\mathrm{ACC}}$: on the input of the authentication information $\pi^{\mathrm{auth}} = (\varphi_1, \varphi_2, \varphi_3, w_z, w_e, w_\eta, w_n, w_o, w_\beta, \mathbb{c}_{auth}, E)$. If $sk_{\mathcal{EV}_j} \in T_{tk}$, $\mathcal{S}$ can compute $s_z := w_z - \mathbb{c}_{auth} \cdot sk_{\mathcal{EV}_j}$, $D_1'^* := e(\varphi_1, g_2)^{-w_\eta} e(\ddot{g}_1, g_2)^{w_n} e(g_1, g_2)^{\mathbb{c}_{auth} \cdot sk_{\mathcal{EV}_j}} e(apk_{\mathcal{RC}}, pk_{\mathcal{RC},1})^{w_o} e(apk_{\mathcal{RC}}, g_2)^{w_\beta} e(g_1, g_2)^{s_z} (\frac{e(\varphi_1, pk_{\mathcal{RC},1})}{e(\dot{g}_1, g_2)})^{-\mathbb{c}_{auth}}$, and then set $\mathrm{H}_3(\varphi_1 \parallel \varphi_2 \parallel \varphi_3 \parallel D_1'^* \parallel D_2' \parallel D_3' \parallel E \parallel \widehat{cred}_j) := \mathbb{c}_{auth}^*$, if $\mathbb{c}_{auth}^* = \mathbb{c}_{auth}$ $\mathcal{S}$ can output the corresponding $tk_j^{\mathrm{acc}*}$ by searching T_{tk}, and stores $(j, tk_j^{\mathrm{acc}*})$ in T_{Auth}, otherwise, return $\perp$.

-$\mathcal{O}_{\mathrm{CH}}$: on input of the identity $j_{b^*}, b^* \in \{0,1\}$, if $j_{b^*} \in L_H$, $\mathcal{S}$ takes out a $tk_{j_{b^*}}^*$ from T_{tk} and simulates $ask_{\mathcal{RC}} := a, apk_{\mathcal{RC}} := v, sk_{ots}^{(1)} := b, apk_{\mathcal{RC}}^{sk_{ots}^{(1)}} := z, (g_1, \varphi_1, \varphi_3) := (u, tk_j^{\mathrm{acc}*} \cdot z, w)$. Next, it chooses $sk_{ots}^{(1)*}, r_v^*, w_z^*, w_e^*, w_\eta^*, w_n^*, w_o^*, w_\beta^*, \mathbb{c}_{auth}^* \leftarrow \mathbb{Z}_p$, if $\mathbb{c}_{auth}^*$ has been queried before, aborts, otherwise it computes $(\varphi_1^* := \varphi_1, \varphi_2^*, \varphi_3^* := \varphi_3, D_1'^*, D_2'^*, D_3'^*, E^*, \widehat{cred}_j^*)$ as usual, sets $\mathrm{H}_3(\varphi_1^* \parallel \varphi_2^* \parallel \varphi_3^* \parallel D_1'^* \parallel D_2'^* \parallel D_3'^* \parallel E^* \parallel \widehat{cred}_j^*) := \mathbb{c}_{auth}^*$, and finally outputs $\pi_{j_{b^*}}^{\mathrm{auth}*}$ and add it to the table T_{Auth}.

$Output$. $\mathcal{A}$ outputs $b' \in \{0,1\}$. Let μ be the probability that $\mathcal{A}$ can succeed in breaking the anonymity in authentication phase. If $z = u^{ab}$, then $log_u^v = log_w^z$, and $\Pr(b' = b^*) > \frac{1}{2} + \mu$. If z is random, $\Pr(b' = b^*) = \frac{1}{2}$. Hence, if $\mathcal{A}$ can win the game with a non-negligible probability, then $\mathcal{S}$ can solve the XDH problem with at least $\frac{\mu}{2}$ probability.

Lemma 1.2 *Under the XDH assumption, no PPT adversary $\mathcal{A}$ can break authentication anonymity in the case of linkability* *required with a non-negligible probability.*

Proof. Suppose $\mathcal{A}$ can break the authentication anonymity of our scheme with non-negligible probability. We can build a polynomial-time simulator $\mathcal{S}$ that break XDH assumption as below. $\mathcal{S}$ is given a tuple $(u, v := u^a, w := u^b, z)$, where $u \in \mathbb{G}_1$, $a, b \in \mathbb{Z}_p$, and either $z = u^{ab}$ or z is a randomness in $\mathbb{G}_1$. $\mathcal{S}$ decides which z was given by interacting $\mathcal{A}$.

Setup. $\mathcal{S}$ generates the public system parameter $pp := (\mathbb{G}_1, \mathbb{G}_2, g_1, \dot{g}_1, \ddot{g}_1, g_2, e, \mathrm{H}_1, \mathrm{H}_2, \mathrm{H}_3)$ as usual and creates a special user j' where its secret key $sk_{\mathcal{EV}_{j'}} := \log_u v$, however $\mathcal{S}$ does not know the secret key. $\mathcal{S}$ creates rest of the users by running the $\mathrm{Reg}_{\mathcal{RC}}$ with $\mathcal{A}$.

-$\mathcal{O}_{\mathrm{H}}$: $\mathcal{S}$ can response to the hash queries for H_1,H_2,H_3 as follows:

1) H_1: let q_h be the expected number of unique H_1 queries. $\mathcal{S}$ chooses a random $j \in [1, q_h]$. If the input has been queried before, it returns the previously queried result. Otherwise, if the input is the j-th unique query on H_1, $\mathcal{S}$ chooses a random $r \leftarrow \mathbb{Z}_p$ and returns w^r. For the rest of the queries, $\mathcal{S}$ chooses a random $r \leftarrow \mathbb{Z}_p$ and returns u^r. Let bsn^* denote the j-th unique query.

2) H_2: only need to ensure its collision resistance.

3) H_3: if the input has not been queried before, return $\mathbb{c} \leftarrow \mathbb{Z}_p$, otherwise return the previously queried result.

-$\mathcal{O}_{\mathrm{REG}}$: $\mathcal{A}$ requests to register an identity j, and $\mathcal{S}$ responses $j' \in [1, q]$, q is the number of register requests from $\mathcal{A}$. If $j = j'$, $\mathcal{S}$ sets $pk_{\mathcal{EV}_{j'}} := pk^*_{\mathcal{EV}_j}$ and randomly chooses $c^*, s^*_a \leftarrow \mathbb{Z}_\mathrm{p}$ to compute $A^* := g_1^{s^*_a} pk_{\mathcal{EV}_{j'}}^{-c^*}$. Then, the oracle performs a patching operation by setting $\mathrm{H}_3(pk_{\mathcal{RC}} \parallel pk_{\mathcal{EV}_j} \parallel A^*) := c^*$. If c^* has been queried before, aborts. Otherwise, $\mathcal{S}$ receives token $tk_{j'}$ from $\mathcal{A}$. If $j \neq j'$, $\mathcal{S}$ chooses $sk_{\mathcal{EV}_j} \leftarrow \mathbb{Z}_p$, computes $pk_{\mathcal{EV}_j} := g_1^{sk_{\mathcal{EV}_j}}$. If $pk_{\mathcal{EV}_j} = v$, aborts, otherwise, $\mathcal{S}$ runs the rest of protocol as the honest $\mathcal{EV}$ with $\mathcal{A}$ as the $\mathcal{RC}$, and receives the token.

-$\mathcal{O}_{\mathrm{AUTH}}$: on the input of identity j, if $j \neq j'$, $\mathcal{S}$ is then able to execute the $\mathrm{Auth}_{\mathcal{EV}}(\cdot)$in the authentication process. Otherwise, $\mathcal{S}$ performs as below:

1) If $bsn = \perp$, $\mathcal{S}$ chooses a random $r \leftarrow \mathbb{Z}_p$, and sets $B := u^r$ and $\varphi_4 := v^r$.

2) If $bsn = bsn^*$, aborts.

3) If $bsn \notin \{\perp, bsn^*\}$, $\mathcal{S}$ searches the log of H_1 queries and retrieves r where $\mathrm{H}_1(bsn) = u^r$. $\mathcal{S}$ sets $B := u^r$ and computes $\varphi_4 := v^r$.

Next, $\mathcal{S}$ takes out a $tk_{j'}$ and $cred_{j'}$ from T_{tk} and chooses $sk^{(1)*}_{ots}, r^*_v, w^*_z, w^*_e, w^*_\eta, w^*_n, w^*_o, w^*_\beta, \mathbb{c}^*_{auth} \leftarrow \mathbb{Z}_p$ to compute $\varphi^*_1, \varphi^*_2, \varphi^*_3, \varphi^*_4, \varphi^*_5, \tilde{\sigma}^*_{k,1}, \tilde{\sigma}^*_{k,2}, \sigma'^*_1, \sigma'^*_2, \tilde{\sigma}^*_1, E^*, D'^*_1, D'^*_2, D'^*_3$ as usual. And it patches the hash by setting $\mathrm{H}_3\big(\varphi^*_1 \parallel \varphi^*_2 \parallel \varphi^*_3 \parallel \varphi^*_4 \parallel \varphi^*_5 \parallel D'^*_1 \parallel D'^*_2 \parallel D'^*_3 \parallel E^* \parallel \widehat{cred}^*_j\big) := \mathbb{c}^*_{auth}$. If it fails, $\mathcal{S}$ aborts; otherwise, $\mathcal{S}$ outputs $\pi^{\mathrm{auth}*} := (\varphi^*_1, \varphi^*_2, \varphi^*_3, \varphi^*_4, \varphi^*_5, w^*_z, w^*_e, w^*_\eta, w^*_n, w^*_o, w^*_\beta, \mathbb{c}^*_{auth}, E^*)$.

-$\mathcal{O}_{EV}$: on the input of an identity j, if $j \neq j'$, it returns $sk_{\mathcal{EV}_j}$; otherwise, aborts.

-$\mathcal{O}_{\mathrm{CH}}$: on input of the identity $j_{b^*}, b^* \in \{0,1\}$. If $j' \notin \{j_0, j_1\}$ or $bsn \notin \{\perp, bsn^*\}$, aborts. Otherwise, $\mathcal{S}$ picks $j_{b^*} = j'$, if $bsn = \perp$, $\mathcal{S}$ chooses a random $r \leftarrow \mathbb{Z}_p$, and sets $B := w^r$ and $\varphi_4 := z^r$. If $bsn = bsn^*$, $\mathcal{S}$ searches the log of H_1 queries and retrieves r where $\mathrm{H}_1(bsn) = w^r$. $\mathcal{S}$ sets $B := w^r$ and computes $\varphi_4 := z^r$. And then $\mathcal{S}$ performs the rest protocol as $\mathcal{O}_{\mathrm{AUTH}}$.

Output. $\mathcal{A}$ outputs $b' \in \{0,1\}$. Let μ be the probability that $\mathcal{A}$ can succeed in breaking the anonymity in authentication phase and the probability that $\mathcal{S}$ does not abort is $1/(q_h \cdot q)$. If $z = u^{ab}$, then $log_u^v = log_w^z$, and $\Pr(b' = b^*) > \frac{1}{2} + \mu$. If z is random, $\Pr(b' = b^*) = \frac{1}{2}$. Hence, if $\mathcal{A}$ can win the game with a non-negligible probability, then $\mathcal{S}$ can solve the XDH problem with at least $\frac{\mu}{2}$ probability.

Theorem 2. *Under the PS assumption, the PAP scheme has unforgeability of the credentials. More specifically, if there is a PPT adversary $\mathcal{A}$ that succeeds with a non-negligible probability to break the unforgeability(Definition 8), then there is a polynomial-time algorithm $\mathcal{S}$ that breaks the PS assumption with a non-negligible probability.*

Proof. Suppose $\mathcal{A}$ can break the unforgeability game of our scheme with non-negligible probability. We can build a polynomial-time simulator $\mathcal{S}$ that break PS assumption as below. Let $a \in \mathbb{G}_1$ and the tuple (a, a^{x+my}) be a PS assumption for (g_2^x, g_2^y), where g_2 is the generator of $\mathbb{G}_2$.

Setup. $\mathcal{S}$ initializes an empty keylist $\mathcal{K}$, which stores the public key of issuers and generates the public parameter $pp = (\mathbb{G}_1, \mathbb{G}_2, g_1, \dot{g}_1, \ddot{g}_1, g_2, e, p, \mathrm{H}_1, \mathrm{H}_2, \mathrm{H}_3)$. $\mathcal{A}$ may request to add pk_k that contains q elements to $\mathcal{K}$. Note that whenever $\mathcal{A}$ wishes to add an issuer's public key pk_k to $\mathcal{K}$, it must prove knowledge of the corresponding secret key.

- $\mathcal{O}_{\mathrm{ISS}}$: $\mathcal{A}$ requests to add a signature for attribute-set $\{\mathfrak{a}_i\}$ under pk^* to the aggregate signature $\hat{\sigma}^*$ for attribute-set: $\left(\{\mathfrak{a}_{1,i,q}\}_{i\in[q]}, \{\mathfrak{a}_{2,i,q}\}_{i\in[q]} \dots\dots, \{\mathfrak{a}_{n_q,i,q}\}_{i\in[q]}\right)$ under public keys $(pk_{1,q}, \dots\dots, pk_{N^*_J,q})$, $\mathcal{S}$ verifies $\hat{\sigma}^*$ and requests a signature on the attribute-set $\{\mathfrak{a}_i\}$, which outputs $(\hat{\sigma}_1, \hat{\sigma}_2)$. All public keys were previously certified, so $\mathcal{S}$ knows the corresponding secret keys. Thus, $\mathcal{S}$ chooses $r \leftarrow \mathbb{Z}_p$, and returns $\hat{\sigma}_{q+1} \leftarrow \mathrm{Issue}_{\mathcal{ISS}}\big(pp, pk_{\mathcal{EV}_u}, sk^*, \{\mathfrak{a}_i\}, \hat{\sigma}_q, \pi^{\mathrm{cred}}_u\big)$, which is valid on $\left(\{\mathfrak{a}_{1,i,q}\}_{i\in[q]}, \{\mathfrak{a}_{2,i,q}\}_{i\in[q]} \dots\dots, \{\mathfrak{a}_{n_q,i,q}\}_{i\in[q]}, \{\mathfrak{a}_i\}_{i\in[q]}\right)$ under $(pk_{1,q}, \dots\dots, pk_{n_q,q}, pk^*)$.

Finally, $\mathcal{A}$ breaks the unforgeability of credentials that means it generates an aggregate signature $\hat{\sigma}^*$ on $(\{\mathfrak{a}^*_{i,1}\}_{i\in[q]}, \{\mathfrak{a}^*_{i,2}\}_{i\in[q]} \dots\dots, \{\mathfrak{a}^*_{i,N^*_J}\}_{i\in[q]})$ under public keys $(pk_1, \dots\dots, pk_{N^*_J})$. The adversary succeed if the aggregate signature $\hat{\sigma}^*$ passes $\mathrm{Verify}^{\mathrm{cred}}_{\mathcal{EV}}$, and there exist a value $k^* \in [N^*_J]$, such that $pk_{k^*} = pk^*$ and set $\{\mathfrak{a}^*_{k^*,i}\}_{i\in[q]}$ was not previously submitted to $\mathcal{O}_{\mathrm{Iss}}$ for aggregation by using pk^*. And all other public keys must be added in $\mathcal{K}$, which means that $pk_k \in \mathcal{K}, \forall k \neq k^*$.

The above simulated signature and actual aggregated signatures both have the form (τ, τ'), $\mathcal{S}$ simulates perfectly, where $\tau \in \mathbb{G}_1$ is a uniform element and $\tau' \in \mathbb{G}_1$ is the unique element satisfying: $e(\tau, \prod_{k\in[N_J]} X_k \cdot (pk_{k,0,q})^{sk_{\mathcal{EV}_u}} \cdot (pk_0^*)^{sk_{\mathcal{EV}_u}} \cdot \prod_{k\in[N_J]} \prod_{i\in[q]} (pk_{k,i,q})^{\alpha_{k,i,q}} \cdot \prod_{i\in[q]} (pk_i^*)^{\alpha_i}) = e(\tau', g_2)$, which has the same distribution as the actual one. According to the assumption, $\mathcal{A}$ can output a forged aggregate signature $\hat{\sigma}^* = (\hat{\sigma}_1^*, \hat{\sigma}_2^*)$ on $(\{\alpha_{i,1}^*\}_{i\in[q]}, \{\alpha_{i,2}^*\}_{i\in[q]} \dots\dots, \{\alpha_{i,Q}^*\}_{i\in[q]})$ under public keys $(pk_1, \dots\dots, pk_{N_J^*})$. And $\mathcal{S}$ can extract all corresponding secret keys and compute $\dot{\hat{\sigma}}_2^* \leftarrow \hat{\sigma}_2^* \cdot \prod_{k\neq k^*} \prod_{i\in[q]} (\hat{\sigma}_1^*)^{-(x_k + y_{k,i}\alpha_{k,i}^*)} \cdot pk_{\mathcal{EV}_u}^{-y_{k,0}}$ and finally outputs a PS signature $(\hat{\sigma}_1^*, \dot{\hat{\sigma}}_2^*)$ for $\{\alpha_{k^*,i}^*\}_{i\in[q]}$ under the challenge key pk^*.

According to the above, an adversary that can forge a valid credential can forge a PS signature, and further break PS assumption[30]. Thus, PAP has unforgeability of the credentials, if PS assumption holds.

Theorem 3. *Under the q-SDH and q-MSDH-1 assumptions, the PAP scheme supports accountability(Definition 9) and unforgeability of revocation tokens. More specifically, if there is a PPT adversary $\mathcal{A}$ that succeeds with a non-negligible probability to break the accountability game or unforgeability game, then there is a polynomial-time algorithm $\mathcal{S}$ that solves the q-SDH problem or q-MSDH-1 problem with a non-negligible probability.*

Lemma 3.1 *Under the q-SDH assumption, no PPT adversary $\mathcal{A}$ can break accountability with a non-negligible probability.*

Proof. Given $(\mathbb{G}_1, \mathbb{G}_2, g_1, \dot{g}_1, \ddot{g}_1, g_2, e, \mathrm{H}_1, \mathrm{H}_2, \mathrm{H}_3)$ and $(q-1)$ SDH tuple $\{(S_j, \eta_j)\}_{j=1}^{q-1}$ as input, where $S_j := g^{\frac{1}{\xi_1+\eta_j}}$, $g := \dot{g}_1\ddot{g}_1{}^n$, one more (S, η) can be transformed into a solution to q-SDH problem. We assume that if there exists a PPT adversary $\mathcal{A}$ that can break the accountability of our scheme, $\mathcal{S}$ can solve the q-SDH problem in polynomial time.

Setup. It sets that $g = \varphi(g_2), pk_{\mathcal{RC},1} := g_2^{\xi_1}$ and $S_j := g^{\frac{1}{\xi_1+\eta_j}}$, while ξ_1 is unknown to $\mathcal{S}$. After that, $\mathcal{S}$ selects $a, b, \eta \leftarrow \mathbb{Z}_p$, and computes that $g_1 := \varphi((pk_{\mathcal{RC},1}^{1+d\cdot n} \cdot g_2{}^{\eta})^{\frac{b}{a}} \cdot g_2{}^{-\frac{1}{a}} := g^{\frac{b(\xi_1+\eta)-1}{a}}$. Finally, $\mathcal{S}$ sends $(\mathbb{G}_1, \mathbb{G}_2, g, g_1, g_2, e, pk_{\mathcal{RC},1})$ to $\mathcal{A}$.

-$\mathcal{O}_{L_H}$: Upon the input of the identity j, if $j \neq j'$, $\mathcal{S}$ proceeds as usual. Otherwise, it returns $(pk_{\mathcal{EV}_j}, \eta_j)$.

-$\mathcal{O}_{L_C}$: If $\mathcal{S}$ receives a query on an honest j, $\mathcal{S}$ returns $sk_{\mathcal{EV}_j}$ in the case of $j \neq j'$ and otherwise, aborts.

-$\mathcal{O}_{\mathrm{H}}$: $\mathcal{S}$ can response to the hash queries for H_1,H_2,H_3 as follows:

1) H_1: only need to ensure its collision resistance.

2) H_2: only need to ensure its collision resistance.

3) H_3: if the input has not been queried before, return $\mathbb{c} \leftarrow \mathbb{Z}_p$, otherwise return the previously queried result.

-$\mathcal{O}_{\mathrm{REG}}$: $\mathcal{A}$ requests to register a new $\mathcal{EV}_j$,and $\mathcal{S}$ randomly responses $j' \in [1, q]$. If $\mathcal{A}$ inputs j, $\mathcal{S}$ chooses $sk_{\mathcal{EV}_j}$ and stores $(j, sk_{\mathcal{EV}_j}, pk_{\mathcal{EV}_j}, \rho_j)$, and if $\mathcal{A}$ inputs j and $sk_{\mathcal{EV}_j}$, $\mathcal{S}$ adds $(j, sk_{\mathcal{EV}_j}, \pi_j^{\mathrm{tok}}, \rho_j)$ to L_C and runs as usual. In the case of $j \neq j'$, $\mathcal{S}$ sets $\dot{tk}_j^{\mathrm{acc}*} = g^b, \eta^* = \eta$ and sends $(\eta^*, \dot{tk}_j^{\mathrm{acc}*})$ to $\mathcal{A}$. Otherwise, either $\mathcal{S}$ chooses $sk_{\mathcal{EV}_j} \leftarrow \mathbb{Z}_p$ or $\mathcal{A}$ sends a $sk_{\mathcal{EV}_j}$, and $\mathcal{S}$ sets $\dot{tk}_j^{\mathrm{acc}*} := (g \cdot g_1{}^{sk_{\mathcal{EV}_j}})^{\frac{1}{\xi_1+\eta_j}} := S_j{}^{1-\frac{sk_{\mathcal{EV}_j}}{a}+\frac{sk_{\mathcal{EV}_j}\cdot b(\eta-\eta_j)}{a}} \cdot g^{sk_{\mathcal{EV}_j}\cdot b/a}$. If $sk_{\mathcal{EV}_j}$ is from $\mathcal{A}$, $\mathcal{S}$ adds $(j, sk_{\mathcal{EV}_j}, pk_{\mathcal{EV}_j}, \dot{tk}_j^{\mathrm{acc}*}, \rho_j)$ to L_C.

-$\mathcal{O}_{\mathrm{ISS}}$: $\mathcal{S}$ simulates the $\mathcal{I}ss$ to execute $\mathrm{Issue}_{\mathcal{RC}}(\cdot)$ protocol and can also act as any honest j', if $j' \neq j$. Otherwise, it simulates the proof of knowledge of $sk_{\mathcal{EV}_j}$ and forge a credential $cred_j^*$.

-$\mathcal{O}_{\mathrm{ACC}}$: on input of a proof π_j^{auth}, it runs $\mathrm{Acc}_{\mathcal{RC}}(\cdot)$ and returns its output.

Finally, $\mathcal{A}$ outputs $(\ddot{\pi}_j^{\mathrm{auth}'}, \widehat{\ddot{cred}}_j')$ and $\mathcal{S}$ can extract $(sk_{\mathcal{EV}_j}^*, \eta^*, \ddot{tk}_j^{\mathrm{acc}*})$. Based on forking lemma[58], $\mathcal{S}$ can choose $\mathbb{c}'_{auth,1}$, $\mathbb{c}'_{auth,2}$ and control the hash values corresponding to $(\varphi_1^*, \varphi_2^*, \varphi_3^*, \varphi_4^*, \varphi_5^*, D_1^*, D_2^*, D_3^*, E^*, \widehat{\ddot{cred}}_j')$. At the same time, $\mathcal{A}$ responses $(w'_{z,1}, w'_{e,1}, w'_{\eta,1}, w'_{n,1}, w'_{o,1}, w'_{\beta,1})$ and $(w'_{z,2}, w'_{e,2}, w'_{\eta,2}, w'_{n,2}, w'_{o,2}, w'_{\beta,2})$. Then, $\mathcal{S}$ can compute $sk_{\mathcal{EV}_j}^* := |w'_{z,1} - w'_{z,2}| / |\mathbb{c}'_{auth,1} - \mathbb{c}'_{auth,2}|$, $\eta^* := |w'_{\eta,1} - w'_{\eta,2}| / |\mathbb{c}'_{auth,1} - \mathbb{c}'_{auth,2}|$ and $\ddot{tk}_j^{\mathrm{acc}*} := \ddot{\varphi}'_1 / (\ddot{\varphi}'_3)^{ask_{\mathcal{RC}}}$. Additionally, if $\ddot{\eta}' \notin \{\eta, \eta_1, \dots, \eta_{q-1}\}$, $\mathcal{S}$ can obtain another valid tuple $(\ddot{S}' := (\ddot{tk}_j^{\mathrm{acc}*} \cdot g^{-sk_{\mathcal{EV}_j}^* \cdot \frac{b}{a}})^{a/(a - sk_{\mathcal{EV}_j}^* + b\cdot sk_{\mathcal{EV}_j}^*(\eta-\eta^*))}, \eta^*)$. And if $\ddot{tk}_j^{\mathrm{acc}*} \in \{\dot{tk}_j^{\mathrm{acc}*}\} \wedge \eta^* \neq \eta$, $\mathcal{S}$ aborts. Otherwise, $\mathcal{S}$ can obtain $(\ddot{S}' := (\ddot{tk}_j^{\mathrm{acc}*} \cdot g^{-sk_{\mathcal{EV}_j}^* \cdot \frac{b}{a}})^{\frac{a}{a - sk_{\mathcal{EV}_j}^*}} := g^{\frac{1}{\xi_1+\eta^*}}, \eta^*)$, which is a SDH tuple.

Therefore, if $\mathcal{A}$ breaks accountability with a non-negligible probability, $\mathcal{S}$ can solve the q-SDH problem with a non-negligible probability.

Lemma 3.2 *Under the q-MSDH-1 assumption, no PPT adversary $\mathcal{A}$ can break unforgeability of revocation tokens with a non-negligible probability.*

Proof. The proof of the unforgeability of tk_u^{rev} relies on q-MSDH-1 assumption, which can refer to [56].

Theorem 4. *Under the XDH assumption, the PAP scheme is anonymous in payment phase. More specifically, if there is a PPT adversary $\mathcal{A}$ that succeeds with a non-negligible probability to break the payment anonymity(Definition 10), then there is a polynomial-time algorithm $\mathcal{S}$ that solves the XDH problem with a non-negligible probability.*

Proof. Suppose $\mathcal{A}$ can break the anonymity game of our scheme with non-negligible probability. We can build a polynomial-time simulator $\mathcal{S}$ that break XDH assumption as below. $\mathcal{S}$ is given a tuple $(u, v := u^a, w := u^b, z)$, where $u \in \mathbb{G}_1$, $a, b \in \mathbb{Z}_p$, and either $z = u^{ab}$ or z is a randomness in $\mathbb{G}_1$. $\mathcal{S}$ decides which z was given by interacting $\mathcal{A}$.

Setup. $\mathcal{S}$ generates the public system parameter $pp := (\mathbb{G}_1, \mathbb{G}_2, g_1, \dot{g}_1, \ddot{g}_1, g_2, e, \mathrm{H}_1, \mathrm{H}_2, \mathrm{H}_3)$ as usual. $\mathcal{A}$ can corrupt entities other than $\mathcal{C}mt$, and the key pair of $\mathcal{C}mt$ is $(pk_{\mathcal{C}mt} := v, sk_{\mathcal{C}mt} := a)$. Subsequently, $\mathcal{S}$ answers the oracle queries as follows:

-$\mathcal{O}_{Addr}$: on the input of an identity j, it generates $(pk_{\mathcal{EV}_j}^{\text{addr}}, sk_{\mathcal{EV}_j}^{\text{addr}})$ and stores in T_{Addr}.

-$\mathcal{O}_{\text{H}}$: $\mathcal{S}$ can response to the hash queries for H_1,H_2,H_3 as follows:

1) H_1: only need to ensure its collision resistance.

2) H_2: only need to ensure its collision resistance.

3) H_3: if the input has not been queried before, return $\mathbb{c} \leftarrow \mathbb{Z}_p$, otherwise return the previously queried result.

- $\mathcal{O}_{\text{VER}}$: on input of $tx \coloneqq (\text{info}_{pay}, \pi^{\text{ch}} \coloneqq (\mathbb{c}_{pay}, w_1, w_2, R_2), \pi^{\text{auth}}, \sigma_{ots}, \widetilde{pk_{\mathcal{EV}}^{\text{addr}}})$, if $sk_{\mathcal{EV}_j}^{\text{addr}} \in T_{Addr}$, $\mathcal{S}$ can compute $sk_{ots}^{(2)} \coloneqq w_2 + \mathbb{c}_{pay} \cdot sk_{\mathcal{EV}_j}^{\text{addr}}$, $R_3^* \coloneqq g_1^{sk_{ots}^{(2)}}$, $\mathbb{c}'_{pay} \coloneqq \text{H}_3(\text{info}_{pay} \parallel \widetilde{pk_{\mathcal{EV}_u}^{\text{addr}}} \parallel pk_{\mathcal{CS}} \parallel pk_{\mathcal{C}mt} \parallel R'_1 \parallel R_2 \parallel R_3^*)$, if $\mathbb{c}_{pay} = \mathbb{c}'_{pay}$, it can output the corresponding $pk_{\mathcal{EV}_J}^{\text{addr}*}$ and stores $(pk_{\mathcal{EV}_J}^{\text{addr}*}, tx_j)$ in T_{tx}.

-$\mathcal{O}_{\text{CH}}$: on input of the identity $j_{b^*}, b^* \in \{0,1\}$, if $j_{b^*} \in L_H$, $\mathcal{S}$ simulates $sk_{\mathcal{C}mt} \coloneqq a$, $pk_{\mathcal{C}mt} \coloneqq v$, $sk_{ots}^{(1)} \cdot sk_{\mathcal{EV}_J}^{\text{addr}} \coloneqq b$, $pk_{\mathcal{C}mt}^{sk_{ots}^{(1)} \cdot sk_{\mathcal{EV}_J}^{\text{addr}}} \coloneqq z$, $(g_1, Q_j, Q'_j) \coloneqq \left(u, w, pk_{\mathcal{EV}_J}^{\text{addr}} \cdot z\right)$, $\widetilde{pk_{\mathcal{EV}_J}^{\text{addr}}}^* \coloneqq (Q_j, Q'_j)$. Next, $\mathcal{S}$ chooses $w_1^*, w_2^*, \mathbb{c}_{pay}^* \leftarrow \mathbb{Z}_p$ and computes R_1^*, R_2^*, R_3^* and patches the hash value by setting $\text{H}_3\left(\text{info}_{pay} \parallel \widetilde{pk_{\mathcal{EV}_J}^{\text{addr}}}^* \parallel pk_{\mathcal{CS}} \parallel pk_{\mathcal{C}mt} \parallel R_1^* \parallel R_2^* \parallel R_3^* \parallel \pi_j^{\text{auth}*}\right) \coloneqq \mathbb{c}_{pay}^*$, and finally outputs tx_{ch}^* to $\mathcal{A}$ and adds to T_{tx}.

Output. $\mathcal{A}$ outputs $b' \in \{0,1\}$. Let μ be the probability that $\mathcal{A}$ can succeed in breaking the anonymity in payment phase. If $z = u^{ab}$, then $log_u^v = log_w^z$, and $\Pr(b^* = b') > \frac{1}{2} + \mu$. If z is random, $\Pr(b^* = b') = \frac{1}{2}$. Hence, if $\mathcal{A}$ can win the game with a non-negligible probability, then $\mathcal{S}$ can solve the XDH problem with at least $\frac{\mu}{2}$ probability.

Theorem 5. *Under the DL problem, the PAP scheme has non-frameability(Definition 11). More specifically, if there is a PPT adversary $\mathcal{A}$ that succeeds with a non-negligible probability to break the non-frameability, then there is a polynomial-time algorithm $\mathcal{S}$ that solves the DL problem with a non-negligible probability.*

Proof. Let $(g_1, g_1{}^z)$ be a DL challenge in $\mathbb{G}_1$. We assume that there exists an adversary $\mathcal{A}$ that can break the *non-frameability* of our privacy-preserving scheme and a simulator $\mathcal{S}$ that solves the DL problem in polynomial time. And $\mathcal{S}$ simulates adversary $\mathcal{A}$'s environment.

Setup. $\mathcal{S}$ generates the public system parameter $pp \coloneqq (\mathbb{G}_1, \mathbb{G}_2, g_1, \dot{g}_1, \ddot{g}_1, g_2, e, \text{H}_1, \text{H}_2, \text{H}_3)$ as usual and $\exists j' \in L_H, sk_{\mathcal{EV}_{j'}} \coloneqq z$, $\mathcal{A}$ will try to impersonate the j'-th honest $\mathcal{EV}$ without knowing its secret key. Then $\mathcal{S}$ answers the oracle queries as follows.

-$\mathcal{O}_{\text{H}}$: $\mathcal{S}$ can response to the hash queries for H_1,H_2,H_3 as follows:

1) H_1: only need to ensure its collision resistance.

2) H_2: only need to ensure its collision resistance.

3) H_3: if the input has not been queried before, return $\mathbb{c} \leftarrow \mathbb{Z}_p$, otherwise return the previously queried result.

-$\mathcal{O}_{\text{REG}}$: $\mathcal{A}$ requests to register an identity j, and $\mathcal{S}$ responses $j' \in [1, q]$. If $j = j'$, $\mathcal{S}$ sets $pk_{\mathcal{EV}_{j'}} \coloneqq pk_{\mathcal{EV}_j}^*$ and randomly chooses $c^*, s_a^* \leftarrow \mathbb{Z}_p$ to compute $A^* \coloneqq g_1^{s_a^*} pk_{\mathcal{EV}_{j'}}^{-c^*}$. Then, the oracle performs a patching operation by setting $\text{H}_3(pk_{\mathcal{RC}} \parallel pk_{\mathcal{EV}_j} \parallel A^*) \coloneqq c^*$. Next, $\mathcal{S}$ receives token $tk_{j'}$ from $\mathcal{A}$. If $j \neq j'$, $\mathcal{S}$ interacts with $\mathcal{A}$ to obtain tk_j.

-$\mathcal{O}_{\text{ISS}}$: $\mathcal{A}$ requests to register an identity j, and $\mathcal{S}$ responses $j' \in [1, q]$. If $j = j'$ $\mathcal{S}$ simulates the proof of knowledge of $sk_{\mathcal{EV}_{j'}}$ as $\mathcal{O}_{\text{REG}}$ performed and receives $cred_{j'}$ from $\mathcal{A}$ by executing $\text{Issue}_{\mathcal{RC}}$. If $j \neq j'$, $\mathcal{S}$ interacts with $\mathcal{A}$ to obtain $cred_j$.

-$\mathcal{O}_{L_C}$: If $\mathcal{S}$ receives a corruption query on an honest $\mathcal{EV}_j$, $\mathcal{S}$ returns $(pk_{\mathcal{EV}_j}, sk_{\mathcal{EV}_j})$ in the case of $j \neq j'$ and otherwise, aborts.

- $\mathcal{O}_{EV}$: on the input of an identity j, it returns $(cred_j, tk_j, sk_{\mathcal{EV}_j})$ and add it to T_{tk}.

-$\mathcal{O}_{\text{AUTH}}$: If the showed attestation belongs to $j \neq j'$, $\mathcal{S}$ is then able to execute the $\text{Auth}_{\mathcal{EV}}(\cdot)$in the authentication process. Otherwise, $sk_{\mathcal{EV}_{j'}}$ is unknown to $\mathcal{S}$ and $\mathcal{S}$ takes out a $tk_{j'}$ and $cred_{j'}$ from T_{tk} and chooses $sk_{ots}^{(1)*}, r_v^*, w_z^*, w_e^*, w_\eta^*, w_n^*, w_o^*, w_\beta^*, \mathbb{c}_{auth}^* \leftarrow \mathbb{Z}_p$ to compute $\varphi_1^*, \varphi_2^*, \varphi_3^*, \varphi_4^*, \varphi_5^*, \tilde{\sigma}_{k,1}^*, \tilde{\sigma}_{k,2}^*, \sigma_1'^*, \sigma_2'^*, \tilde{\sigma}_1^*, E^*, D_1'^*, D_2'^*, D_3'^*$ as usual. And it patches the hash by setting $\text{H}_3\left(\varphi_1^* \parallel \varphi_2^* \parallel \varphi_3^* \parallel \varphi_4^* \parallel \varphi_5^* \parallel D_1'^* \parallel D_2'^* \parallel D_3'^* \parallel E^* \parallel \widehat{cred}_{j'}^*\right) \coloneqq \mathbb{c}_{auth}^*$. If it fails, $\mathcal{S}$ aborts; otherwise, $\mathcal{S}$ computes $\dot{F} \coloneqq e\left(X \cdot \tilde{\sigma}_1{}^* \prod_{i \in I,k} Y_{k,i}^{\mathfrak{a}_{k,i}}, (\sigma_1{}'^*)^{-1}\right)$, and $e\left(\prod_k Y_{k,0}^{w_z^*}, \sigma_1'^*\right) \cdot E^{*-1} = \left(\dot{F} \cdot e(\sigma_2'^*, g_2)\right)^{\mathbb{c}_{auth}^*}$. Then $\mathcal{S}$ can set $\pi_{j'}^{\text{auth}*} \coloneqq (\varphi_1^*, \varphi_2^*, \varphi_3^*, \varphi_4^*, \varphi_5^*, w_z^*, w_e^*, w_\eta^*, w_n^*, w_o^*, w_\beta^*, \mathbb{c}_{auth}^*, E^*)$, $\widehat{cred}_{j'}^* \coloneqq (\{\tilde{\sigma}_{k,1}^*, \tilde{\sigma}_{k,2}^*\}_k, \sigma_1'^*, \sigma_2'^*, \tilde{\sigma}_1^*)$, output $(\pi_{j'}^{\text{auth}*}, \widehat{cred}_{j'}^*)$ to $\mathcal{A}$.

Once $\mathcal{A}$ presents its $\left(\pi_{j'}^{\text{auth}*}, \widehat{cred}_{j'}^*\right)$, $\mathcal{S}$ can extract the knowledge of $sk_{\mathcal{EV}_j}^*$. By using the forking lemma[58], $\mathcal{S}$ can rewind $\mathcal{A}$ and control the result of the hash by choosing two different values $\mathbb{c}'_{auth,1}, \mathbb{c}'_{auth,2}$ to the same $(\varphi_1^*, \varphi_2^*, \varphi_3^*, \varphi_4^*, \varphi_5^*, D_1'^*, D_2'^*, D_3'^*, E^*, \widehat{cred}_{j'}^*)$ while $\mathcal{A}$ responses $(w'_{z,1}, w'_{\eta,1}, w'_{e,1}, w'_{n,1}, w'_{o,1}, w'_{\beta,1})$ and $(w'_{z,2}, w'_{\eta,2}, w'_{e,2}, w'_{n,2}, w'_{o,2}, w'_{\beta,2})$. Next, $\mathcal{S}$ obtains $sk_{\mathcal{EV}_j}^* \coloneqq |w'_{z,1} - w'_{z,1}| / |\mathbb{c}'_{auth,1} - \mathbb{c}'_{auth,2}|$ and returns it as a valid solution to the DL problem.

Therefore, if $\mathcal{A}$ succeeds to impersonate an honest $\mathcal{EV}$ with a non-negligible probability that pass $\text{Verify}_{\mathcal{CS}}^{\text{auth}}$, $\mathcal{S}$ can solve the DL problem with a non-negligible probability.

Theorem 6. *Under the PS assumption, the PAP scheme has payment binding(Definition 12). More specifically, if there is a PPT adversary $\mathcal{A}$ that succeeds with a non-negligible probability to break the payment binding, then there is a polynomial-time algorithm $\mathcal{S}$ that breaks the PS assumption with a non-negligible probability.*

Proof. The OTS of PAP in payment phase relies on the unforgeability of PS signature, which can refer to [30] and since PS signature cannot be forged, our σ_{ots} supports unforgeability, thus, no PPT adversary can break the payment binding.