TACB: Traceable Anonymous Credentials with Batch Verification for Universal Internet of Vehicles

Jiaxin Yu[1], Xuan Zhao[1], Yanqi Zhao[1], Min Xie[2], Xiaoyi Yang[1]
{yjx06132025@163.com, 17691069283@stu.xupt.edu.cn, zhaoyanqi2019@163.com,
minxie@stu.hit.edu.cn, yangxiaoyi@xupt.edu.cn}

[1]School of Cyberspace Security, xi'an University of Posts and Telecommunications, Xi'an, China
[2]School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China

Abstract. The Universal Internet of Vehicles (UIoV) depends on Roadside Connection Node (RCN) to establish secure Vehicle-to-Everything (V2X) communication, necessitating both efficient and privacy-preserving vehicle authentication mechanisms. However, current anonymous credential schemes in UIoV are prone to uncontrolled anonymity abuse and suffer from high computational overhead. To issue these challenges, we propose TACB, a traceable anonymous credential with batch verification for the UIoV, which balances the privacy protection and regulatory. The TACB supports a conditional traceability mechanism that enables authorized entities to reveal the identities of malicious vehicles while preserving anonymity for legitimate users. We give the construction of TACB, which is based on the PS signatures, but is pairing-free on the vehicle sides. Moreover, efficient batch verification significantly reduces computational overhead. Finally, we conduct an analysis of the computational costs for the TACB and compare it with related works to demonstrate the effectiveness and practicality of TACB.

Keywords: Universal internet of vehicles, Anonymous authentication, Privacy preservation, Traceability

1 Introduction

With the growing number of private vehicles, establishing secure and intelligent transportation systems has become crucial. The Internet of Vehicles (IoV) [1] enables comprehensive connectivity through V2V, V2I, V2P, and V2N communications using sensors, advanced communications, and cloud computing. In the IoV environments, strict vehicle authentication is essential [2]. Without reliable verification, malicious participants can impersonate legitimate vehicles to spread false information [3], for example, fabricating "congestion ahead" messages to cause traffic disruptions. Many IoV services also require the validation of specific attributes such as the subscription status for paid services. Anonymous credential schemes are typically used for such authentication [4]. Vijayakumar et al. proposed an anonymous authentication and key exchange for 6G [5] to reduce RCNs

IIKI 2025, December 18-21, Chengdu, People's Republic of China

DOI 10.4108/eai.18-12-2025.2365283

authentication load in dense areas. Azees et al. introduced an anonymous authentication scheme for VANETs [6] that provides conditional privacy—anonymous normal operation with immediate tracking for rule violators. However, these schemes often employ complex cryptographic primitives that compromise efficiency while achieving anonymity and traceability. How can we design an anonymous credential scheme for UIoV that achieves both privacy protection and traceability while maintaining low computational overhead?

1.1 Our Contributions

This paper proposes TACB, a traceable anonymous credential scheme with batch verification for Universal Internet of Vehicles. The scheme provides privacy preservation, traceability, and batch verification in UIoV. Our main contributions are:

(1) We design the TACB with conditional traceability, allowing authorized entities to reveal malicious vehicle identities while maintaining anonymity for legitimate users, thus balancing privacy and regulation in UIoV.

(2) Based on the PS signatures, our construction eliminates pairing operations on the vehicle side, significantly reducing computational costs. In addition, the scheme supports attribute privacy and efficient batch verification.

(3) Our scheme drastically reduces reliance on bilinear pairings during batch verification. Comparative analysis shows significantly lower computational costs than existing schemes.

1.2 Related Work

Anonymous authentication remains a key concern in IoV. Jiang et al. introduced AAAS [7], which uses regional trust authorities for efficient vehicle authentication. Feng et al. developed BPAS [8], a blockchain-based system that enables privacy-preserving authentication even when trusted parties are offline. Du et al. proposed an anonymous V2V authentication for RCN-less environments using the Authentication Server Function (AUSF) [9]. To reduce communication burden, Han et al. introduced a fog computing-based scheme for VANETs [10] that enables self-authentication between vehicles and RCN. Based on this, Liu et al. created ATRC [11], a blockchain-based credential system that reduces computational costs in the presentation and revocation phases. Bi et al. introduced an anonymous authentication protocol for fog-assisted V2G networks [12] that supports mutual authentication and selective attribute disclosure.

1.3 Organization

In Section II, we review the preliminaries. Subsequently, the system architecture is detailed in Section III. In Section IV, we describe our TACB scheme. The performance analysis is detailed in Section V. We provide conclusions in Section VI.

2 Preliminaries

2.1 Bilinear Maps

This work uses a Type-3 bilinear group setting [13], defined by $\mathscr{G} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, \hat{g})$. Here, $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ are distinct cyclic groups of prime order p, with generators g and $\hat{g}$ for $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively. The bilinear pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is efficiently computable and satisfies:

- **Bilinearity**: $\forall x, y \in \mathbb{Z}_p$, $e(g^x, \hat{g}^y) = e(g, \hat{g})^{xy}$.
- **Non-degeneracy**: $e(g, \hat{g})$ generates $\mathbb{G}_T$.
- **Efficiency**: $e(P, Q)$ is computable in polynomial time for all $P \in \mathbb{G}_1$, $Q \in \mathbb{G}_2$.

2.2 Complexity Assumption

PS assumption [14] : Given $(\hat{g}^x, \hat{g}^y)$, an adversary with access to a signing oracle $\mathscr{O}^{PS}(m)$ that returns (u, u^{x+ym}) for random $u \in \mathbb{G}_1$ and $m \in \mathbb{Z}_p$, cannot output a fresh valid tuple $(u', v', m) \in \mathbb{G}_1^2 \times \mathbb{Z}_p$ where $v' = u'^{x+ym}$ and m was not queried.

We use a **generalized PS (GPS) Assumption** with two constrained oracles:

- $\mathscr{O}_0^{GPS}()$: Returns random $u \in \mathbb{G}_1$.
- $\mathscr{O}_1^{GPS}(u, m)$: Returns $v = u^{x+ym}$ only if u was generated by $\mathscr{O}_0^{GPS}$ and not previously used.

The goal remains to forge a new valid tuple (u', v', m) under these constraints.

We further define a **Multi-Message Generalized PS (VMGPS) Assumption** for multiple attributes. The adversary gets $(\hat{g}^x, \hat{g}^{y_1}, \ldots, \hat{g}^{y_n})$ and accesses:

- $\mathscr{O}_0^{VMGPS}()$: Returns random $u \in \mathbb{G}_1$.
- $\mathscr{O}_1^{VMGPS}(g, u, f_1, \ldots, f_n, w_1, \ldots, w_n)$: Returns $v = u^x \prod_{i=1}^n w_i^{y_i}$ only if:

1. u is fresh (not previously queried to $\mathscr{O}_1^{VMGPS}$).
2. u was generated by $\mathscr{O}_0^{VMGPS}$.
3. $\log_g f_i = \log_u w_i$ for all $i \in [1, n]$.

The VMGPS problem is to compute a forgery $(u', v', m_1, \ldots, m_n)$ where $v' = u'^{x+\sum_{i=1}^n y_i m_i}$ for a new message combination. The key difference in VMGPS is that instead of providing scalars m_i directly, the adversary must provide group elements $w_i = u^{m_i}$ along with $(g, f_i = g^{m_i})$, embedding a proof of knowledge of m_i.

2.3 Signature Proof of Knowledge

Signature Proof of Knowledge (SPK) π allows a prover to demonstrate knowledge of a secret x satisfying $y = f(x)$ while signing a message m [15]. This is abstractly denoted as:

$$\pi = SPK\{(x) : y = f(x)\}(m)$$

Concretely, $\pi = (c, s)$ is generated as:

- Choose random rnd $\in \mathbb{G}$
- Compute $c = h(f, y, f(\text{rnd}), m)$
- Compute $s = \text{rnd} - c \cdot x$

Verification checks: $c \stackrel{?}{=} h(f, y, y^c f(s), m)$

Multiple proofs can be combined using the AND operator to prove knowledge of multiple secrets simultaneously.

2.4 PS Signatures

The Pointcheval-Sanders (PS) signature operates in Type-3 bilinear groups and consists of:

- params $\leftarrow$ **Setup**(1^λ): Generate params $= (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2)$ where $g_1 \in \mathbb{G}_1$, and $g_2 \in \mathbb{G}_2$ are generator of $\mathbb{G}_1, \mathbb{G}_2$, respectively.
- $(sk, pk) \leftarrow$ **Keygen**(params): Pick $x, y \in \mathbb{Z}_p$, set $sk = (x, y)$, $pk = (g_2, g_2^x, g_2^y)$.
- $\sigma \leftarrow$ **Sign**(params, sk, m): Pick random $\alpha \in \mathbb{Z}_p$, output $\sigma = (g_1^\alpha, g_1^{\alpha(x+ym)}) = (a, b)$.
- $b' \leftarrow$ **Verify**(params, pk, m, σ): Check $e(a, g_2^x \cdot g_2^{ym}) = e(b, g_2)$. Output 1 if equal, 0 otherwise.

3 System Architecture and Security Requirements

This section introduces the system architecture and the security requirements.

3.1 System Architecture

As shown in Fig. 1, the system includes five entities.

Vehicle Management Platform (VMP): Initializes the system, verifies vehicle legitimacy, and issues valid credentials to authorized vehicles.

Vehicle: Requests and obtains credentials from VMP, then presents them to RCN for service access.

Roadside Connection Node (RCN): Verifies vehicle anonymous credential and provides services after successful authentication.

Traffic Control Department (TCD): Monitors and traces malicious vehicle activities.

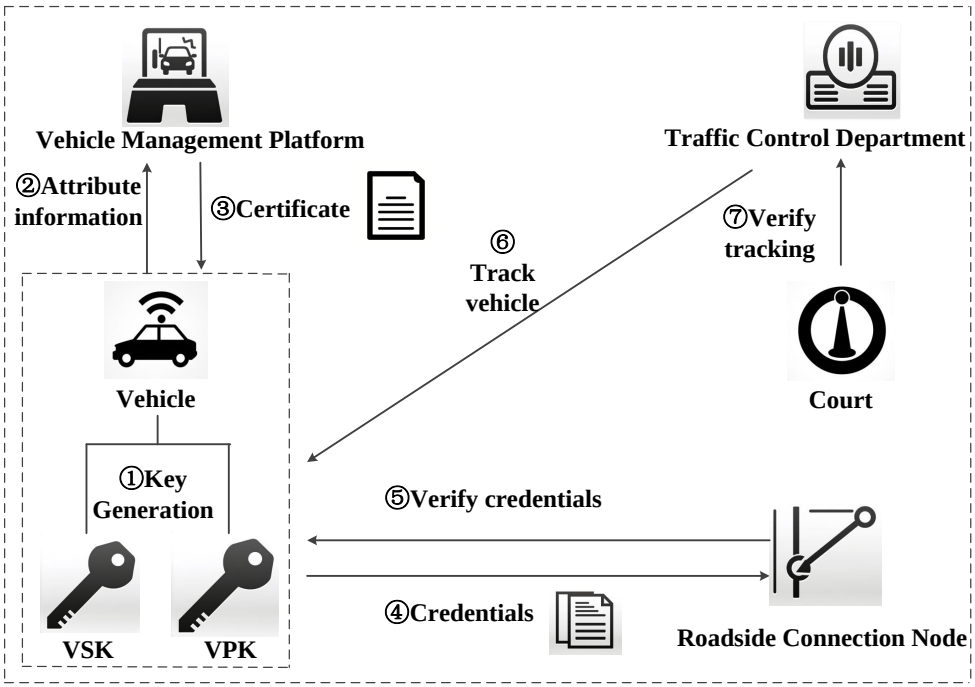


Fig. 1. Overview of system architecture

Court: Validates TCD's identity tracing results to ensure accuracy and prevent misidentification.

The workflow of TACB scheme is as follows: VMP initializes the system and generates public parameters. The vehicle then generates its public-private key pair (Step ①). When a vehicle joins, it submits its attributes to VMP (Step ②). After verification, VMP issues an attribute-based credential (certificate) to the vehicle (Step ③) and records it in the registration list. When requesting roadside services, the vehicle presents its anonymous credential to RCN (Step ④). RCN verify the credential validity (Step ⑤). If malicious activity is detected, the Traffic Control Department (TCD) traces the vehicle's real identity (Step ⑥) and submits the result to Court for verification Step ⑦), ensuring fair identity disclosure.

3.2 Syntax Definition

The TACB scheme comprises polynomial-time algorithms: **Setup**, **Gkg**, **Vkg**, **Request**, **Issue**, **Showing**, **Verify**, **Trace**, **Judge**. It supports efficient batch vehicle verification and ensures full identity traceability.

(1)$(params) \leftarrow$ **Setup**(1^{λ}): Input security parameters λ and generate system public parameters *params*.

(2)$(gpk, ik, tk) \leftarrow$ **Gkg**$(params)$: Input the system's public parameters and outputs the public key gpk, the VMP's secret key ik, the TCD's secret key tk.

(3)$(vpk_j, vsk_j) \leftarrow$ **Vkg**$(params, j)$: Input the security parameters *params*, Vehicle j obtains a pair of Vehicle keys (vpk_j, vsk_j).

(4)$(gsk_j, reg_j) \leftarrow$ (**Request**$(params, gpk, vsk_j, f(m_1, \cdots, m_n)) \Leftrightarrow$ **Issue**$(ik, vpk_j, params)$): VMP executes the interactive protocols with Vehicle. The **Request** algorithm inputs parameters *params*, gpk, vsk_j and Vehicle attributes $f(m_1, \cdots, m_n)$, the **Issue** algorithm inputs vpk_j, ik and *params*. Finally, the vehicle outputs gsk_j and the VMP adds reg_i.

(5)$(\sigma) \leftarrow$ **Showing**$(params, gsk_j, f(m_1, \cdots, m_n))$: Vehicle takes the system parameters *params*, Vehicle attributes $f(m_1, \cdots, m_n)$, and credential gsk_j as inputs, and outputs the anonymous credential σ.

(6)$(b) \leftarrow$ **Verify**$(params, \sigma, f(m_1, \cdots, m_n))$: The RCN takes parameters *params*, revelated attributes $f(m_1, \cdots, m_n)$ and vehicle credentials σ as input, which yields $b \in \{0, 1\}$.

(7) $(j, \Pi) \leftarrow$ **Trace**$(params, \sigma, f(m_1, \cdots, m_n), tk, reg_j)$: The TCD to track illegal Vehicles. The TCD takes the parameters *param*, attributes $f(m_1, \cdots, m_n)$, the credential σ, the TCD's secret key tk and reg_j as input and outputs a proof of the vehicle j.

(8)$(b) \leftarrow$ **Judge**$(params, (\sigma, \Pi), f(m_1, \cdots, m_n), j, gpk, vpk_i)$: This algorithm is executed by the Court for the purpose of verifying the TCD proof. The algorithm takes parameters *params*, the proof Π, the credential σ, attributes $f(m_1, \cdots, m_n)$, vpk_j and gpk as inputs and outputs $b \in \{0, 1\}$.

3.3 Security Requirements

Anonymity. The adversary cannot link an anonymous credential to its originating vehicle, even when colluding with VMP. Given all keys except two target vehicles, the adversary cannot associate a challenge credential with either target.

Non-frameability. It prevents forging credentials falsely attributed to honest vehicles. Even controlling VMP and TCD, the adversary cannot produce a valid credential-proof pair for a target identity that would be accepted by Judge.

Traceability. Ensures all valid credentials are traceable to registered vehicles. Against an honest VMP, the adversary cannot produce a valid credential that remains untraceable or is rejected by Judge.

4 Scheme Construction

In the section, we give the construction of TACB, which is based on the PS signature, but is pairing-free on the vehicle side.

- **Setup:** Generate the system parameters.
  1. Define the public parameters $pp = (G, H, H_1)$. Here, $G = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, \hat{g})$ represents a Type-3 bilinear group structure. $H : (0,1)^* \rightarrow \mathbb{G}_1$ and $H_1 : (0,1) \rightarrow \mathbb{Z}_p$ are cryptographic hash functions.

- **Gkg:** The key generation does the following.
  1. Select random exponents $x, y_1, \cdots, y_{n+1} \in \mathbb{Z}_p$ and compute the VMP's public key components $\hat{X} = \hat{g}^x$, $\hat{Y}_1 = \hat{g}^{y_1}$, $\hat{Y}_2 = \hat{g}^{y_2}$, $\cdots$, $\hat{Y}_{n+1} = \hat{g}^{y_{n+1}}$.
  2. Select random exponents $z_0, z_1 \in \mathbb{Z}_p$ and compute the TCD's public key components $\hat{Z}_0 = \hat{g}^{z_0}$, $\hat{Z}_1 = \hat{g}^{z_1}$.
  3. The public key is set as $gpk = (\hat{X}, \hat{Y}_1, \hat{Y}_2, \cdots, \hat{Y}_{n+1}, \hat{Z}_0, \hat{Z}_1)$, The VMP's secret key is $ik = (x, y_1, \cdots, y_{n+1})$, and the TCD's secret key is $tk = (z_0, z_1)$.

- **Vkg:** Select a random $vsk_j \in \mathbb{Z}_p$ as the vehicle's secret key. Compute the public key as $vpk_j = \hat{g}^{vsk_j}$, the vehicle j obtains a vehicle key pair (vpk_j, vsk_j).

- **Request $\Leftrightarrow$ Issue:** The vehicle j and the Vehicle Management Platform execute the interactive protocols.
  1. **Request.** The vehicle chooses $m_1, \cdots, m_n, s_0, s_1 \in \mathbb{Z}_p$, randomly and sets $m_{n+1} = vsk_j$.
  (1) The vehicle computes $f_i = g^{m_i}$, for $i = 1, \cdots, n+1$, and computes $H(f_1, \cdots, f_{n+1}) = u$. Then, compute $w_i = u^{m_i}$, for $i = 1, \cdots, n+1$, and compute double encryption of $\hat{S}_0 = \hat{g}^{s_0}$, $\hat{S}_1 = \hat{g}^{s_1}$, $\hat{f}'_0 = \hat{g}^{m_{n+1}} \hat{Z}_0^{s_0}$, $\hat{f}'_1 = \hat{g}^{m_{n+1}} \hat{Z}_1^{s_1}$.
  (2) The vehicle generates a proof to prove it holds the secret values

$$\begin{aligned} \pi_1 = \mathrm{SPK}\{(m_i, s_0, s_1, i = 1, \cdots, n+1) : \\ f_i = g^{m_i} \wedge w_i = u^{m_i} \wedge \hat{S}_0 = \hat{g}^{s_0} \wedge \hat{S}_1 = \hat{g}^{s_1} \wedge \\ \hat{f}'_0 = \hat{g}^{m_{n+1}} \hat{Z}_0^{s_0} \wedge \hat{f}'_1 = \hat{g}^{m_{n+1}} \hat{Z}_1^{s_1}\}. \end{aligned}$$

  (3) The vehicle runs the signature $DSSig$ to obtain the signature $\sigma_{DS} = H(\tau)^{vsk_j}$, where $\tau = (\pi_1, vpk_j)$.
  (4) Finally, it sends the tuple $(f_i, w_i, \hat{S}_0, \hat{S}_1, \hat{f}'_0, \hat{f}'_1, \pi_1, \sigma_{DS}, vpk_j)$ for $i = 1, \cdots, n+1$ to the VMP.

  2. **Issue.** Upon receipt, the VMP recomputes $H(f_1, \cdots, f_{n+1}) = u$ and parses $\tau = (\pi_1, vpk_j)$. It then checks the following:
  (1) Check the proof π_1.
  (2) Verify the digital signature σ_{DS} on τ under the vehicle's public key vpk_j.
  (3) If all checks pass, VMP adds $reg_j = (j, \hat{S}_0, \hat{S}_1, \hat{f}'_0, \hat{f}'_1, \tau, \sigma_{DS})$ to the list reg. Then, VMP computes $v = u^x \prod_{i=1}^{n+1} w_i^{y_i}$. To reduce the computation cost and delete the pairing operations on the vehicle side, the VMP generates a proof π_2 for v, where

$$\begin{aligned} \pi_2 = \mathrm{SPK}\{(x, y_i, i = 1, \cdots, n+1) : \\ v = u^x \prod_{i=1}^{n+1} w_i^{y_i} \wedge \hat{X} = \hat{g}^x \wedge \hat{Y}_i = \hat{g}^{y_i}\}. \end{aligned}$$

  (4) Sent v, π_2 to the vehicle.

  3. **Request.** Vehicle checks the validity of v, π_2, which is shown as follows and constructe the $gsk_j = (v, m_1, \cdots, m_{n+1})$.

- **Showing.** The vehicle generates an anonymous credential with $gsk_j = (v, m_1, \cdots, m_{n+1})$.
  1. Select random exponents $r \in \mathbb{Z}_p$, then computes $u' = u^r$, $v' = v^r$, $w'_i = w_i^r$
  2. Construct a proof of knowledge for the vehicle's confidential parameters $m_1, \cdots, m_{n+1}$:

$$\pi_3 = SPK\{(m_1, \cdots, m_{n+1}) : w'_i = u'^{m_i}\}.$$

  The anonymous credential is $\sigma = (u', v', w'_i, \pi_3)$.

- **Verify.** The RCN validates the proof $\sigma = (u', v', w'_i, \pi_3)$ through these steps:
  1. Verify the validity of π_3 concerning (u', w'_i) and m_i. If valid, subsequently check:

$$e(v', \hat{g}) = e(u', \hat{X}) \prod_{i=1}^{n+1} e(w'_i, \hat{Y}_i).$$

  2. Output 1 if both checks pass; otherwise output 0.

- **Trace.** For a valid credential $\sigma = (u', v', w'_i, \pi_3)$, the TCD executes:
  1. For each registry entry $reg_j = (j, \hat{S}_0, \hat{S}_1, \hat{f}'_0, \hat{f}'_1, \tau, \sigma_{DS}) \in reg$:
  (1) Randomly select $b \in \{0, 1\}$.
  (2) Compute $\hat{f}_{n+1} = \hat{f}'_b (\hat{S}_b^{z_b})^{-1}$
  (3) Verify $e(u', \hat{f}_{n+1}) = e(w'_{n+1}, \hat{g})$.
  2. Return $\perp$ if no registry entry satisfies the check.
  3. Upon finding a matching $\hat{f}_{n+1}$ for a certain reg_j, generate proof:

$$\pi_4 = SPK\{(\hat{f}_{n+1}) : e(w'_{n+1}, \hat{g}) = e(u', \hat{f}_{n+1})\}$$

  4. $\Pi = (\tau, \sigma_{DS}, \pi_4)$

- **Judge.** Given $(m_1, m_2, \cdots, m_{n+1}, \sigma, gpk, j, vpk_j, \Pi)$, with valid credential $\sigma = (u', v', w'_i, \pi_3)$ and proof $\Pi = (\tau, \sigma_{DS}, \pi_4)$, this algorithm:
  1. Validate π_4 and $DSVf = (vpk_j, \tau, \sigma_{DS}) = 1$.
  2. Validate $e(\sigma_{DS}, \hat{g}) = e(H(\tau), vpk_j)$. Output 1 if all conditions are satisfied; otherwise 0.

Remark. Multiple anonymous credentials (up to k) from distinct users can be efficiently verified using pairing-based batch computation techniques, as detailed in the Batch Verification Algorithm **BGVF**.

- **BGVF.** The algorithm processes k anonymous credentials $\{\sigma_t\}_{t=1}^k$. Define ℓ as a small prime and define the t-th anonymous credential as $\sigma_t = (u'_t, v'_t, w'_{t,i}, \pi_{t,3})$ among $i = 1, \cdots, n+1$. The RCN performs the following steps:
  1. For each $t = 0, \ldots, k$ verify the validity of $\pi_{t,3}$ with respect to $(u'_t, w'_{t,i})$.
  2. If all individual proofs are valid, randomly select exponents $e_1, \ldots, e_k \in \{0, 1\}^\ell$ and compute the batched values: $\{\tilde{u}_t = u_t'^{e_t}, \tilde{v}_t = v_t'^{e_t}, \tilde{w}_{t,i} = w_{t,i}'^{e_t}\}_{t=1}^k$.
  3. Check the following batched pairing equation:

$$e(\prod_{t=1}^{k} \tilde{v}_t, \hat{g}) = e(\prod_{t=1}^{k} \tilde{u}_t, \hat{X}) \cdot \prod_{i=1}^{n+1} e(\prod_{t=1}^{k} \tilde{w}_{t,i}, \hat{Y}_i).$$

4. If all verification steps succeed, output 1; otherwise, output 0.

4.1 Security Analysis

The TACB achieves anonymity, non-frameability, and traceability.

Theorem 1. (Anonymity) TACB scheme provides anonymity under the SXDH assumption, SPKs are simulation-sound extractable NIZKs and H is a random oracle.

Proof. First, by the zero-knowledge property of SPKs, real proofs are replaced with simulated ones. Next, under the $\text{XDH}_{\mathbb{G}_2}$ assumption, tracing components in the credential are randomized. Finally, under the $\text{XDH}_{\mathbb{G}_1}$ assumption, the credential's attribute-related components are replaced with random group elements, fully hiding the user's identity. Since all transitions are computationally indistinguishable, the scheme achieves anonymity.

Theorem 2. (Non-frameability) TACB scheme satisfies non-frameability under the SDL assumption, the EUF-CMA security of the PS signature, and the simulation-sound extractability of the SPKs.

Proof. Any successful framing attack by an adversary must fall into one of three cases: (1) forging the VMP's signature, violating EUF -CMA; (2) breaking the soundness of the SPK for tracing, violating simulation-sound extractability; or (3) solving the SDL problem by extracting a vehicle's secret key from a forged credential. Since all three are computationally hard, non-frameability holds.

Theorem 3. (Traceability) TACB scheme achieves traceability based on the VMGPS assumption and the simulation-soundness of SPK_1.

Proof. Any adversary producing an untraceable valid credential must either (1) forge a credential with new secret attributes—allowing reduction to the VMGPS assumption via extraction from the NIZK proof π_3, or (2) cause the **Judge** to reject a valid trace—violating the soundness of the underlying proof system. Since both are computationally hard, traceability holds.

5 Performance Analysis

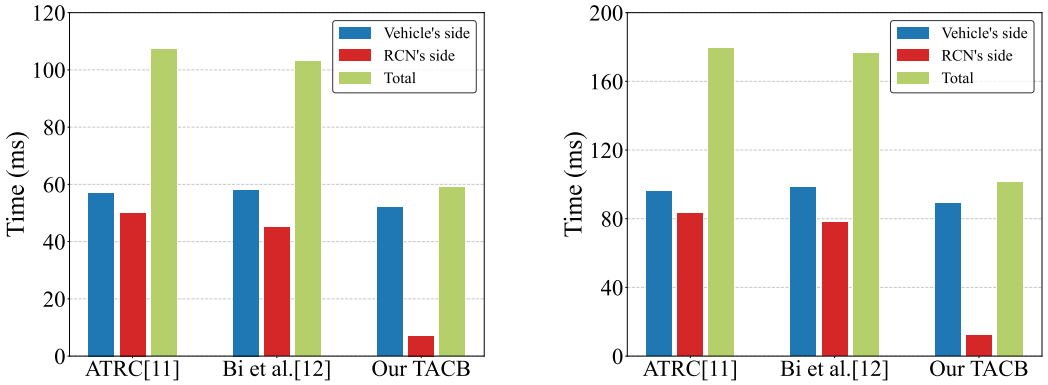
We implement TACB scheme using the Python programming language on the Windows 10 and Raspberry Pi 4B. The Windows system is equipped with an Intel(R) Core(TM) i5-10300H CPU, 16GB of memory, and an NVIDIA GeForce GTX 1650 Ti Intel UHD Graphics. The Raspberry Pi 4B configuration includes a Cortex-A72 processor clocked at 1.5GHz and 4GB of onboard RAM. We mainly compared its performance with ATRC [11] and Bi et al. [12] in terms of computational and communication costs, with the number of attributes fixed to one for fairness.

5.1 Computational Cost Evaluation

We define T_e, T_p, T_h, and T_m as the time for exponentiation, pairing, hash, and point multiplication operations and evaluate the verification phase computational overhead on vehicle side and RCN side. Based on 100 simulation runs, the average operation times on Windows 10 ($T_e = 1.9104$ ms, $T_p = 1.6205$ms, $T_h = 3.9520$ ms, $T_m = 0.0161$ ms) are considerably lower than those on the Raspberry Pi 4B ($T_e = 3.1892$ ms, $T_p = 3.0084$ ms, $T_h = 7.1172$ ms, $T_m = 0.0456$ ms). Table 1 shows the detailed computation cost comparison of algorithms.

Table 1: Computation Cost Comparison of algorithms

Scheme	Request	Issue	Showing	Verify	Trace	Judge
ATRC[11]	$2T_e + T_m$	$21T_e$	$9T_e$	$26T_e$	$7T_e + 2T_m + 2T_h$	-
Bi et al.[12]	$7T_e + 4T_p$	$2T_e + 2T_p$	$18T_e + 2T_p$	$17T_e + 8T_p$	-	-
Our TACB	$17T_e + 2T_h + 4T_m$	$18T_e + 11T_m$	$4T_e + T_h$	$T_e + 3T_p + T_m$	$4T_e + 3T_p + 2T_m$	$T_e + 4T_p + T_m$



(a) On Windows 10 **(b)** On Raspberry Pi 4B

Fig. 2. The computation cost of Vehicle, RCN and Total.

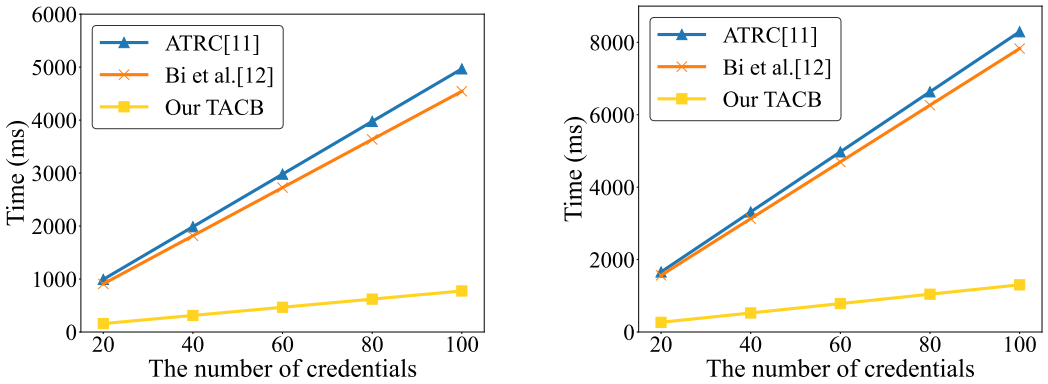
According to the time of four operations and the Table 1, on windows 10, the time of vehicle's side in TACB (**Request** and **Showing**) is 52 ms, while the time of RCN's side (**Verify**) is 7 ms, totaling 59 ms. In comparison, ATRC [11] requires 107 ms, and Bi et al. [12] requires 103 ms. On Raspberry Pi 4B, the time of vehicle's side in TACB is 89 ms, while the time of RCN's side is 12 ms, totaling 101 ms. ATRC [11] requires 179 ms, and Bi et al. [12] requires 176 ms. The computation cost of vehicle, RCN and Total is shown in Fig. 2, whether on Windows 10 or on the Raspberry Pi 4B, TACB significantly reduces computational overhead on both sides.

The computation cost comparison of batch verification is shown in Table 2. Our TACB significantly reduces intensive operations (pairings, exponentiations) during verification, yielding lower computational costs and higher message throughput versus other schemes.

Fig. 3 shows the time comparison of batch verification. TACB maintains significantly higher efficiency with growing credential numbers. For 100 vehicles, our verification on Windows 10 requires only 775 ms versus over 4000 ms for others. On Raspberry Pi 4B, verifying 100 vehicles requires 1303 ms versus about 8000 ms for others. It is evident that the verification time on the Raspberry Pi is twice that of the Windows 10 system. Despite this, our solution maintains superior

Table 2: Computation Cost Comparison of Batch Verification

Scheme	Verify a single credential	Verify n credentials
ATRC[11]	$26T_e$	$26nT_e$
Bi et al.[12]	$17T_e + 8T_p$	$17nT_e + 8nT_p$
Our TACB	$T_e + 3T_p + T_m$	$4nT_e + 3T_p + (4n-3)T_m$



(a) on Windows 10 **(b)** on Raspberry Pi 4B

Fig. 3. The time comparison of batch verification.

verification efficiency in both experimental environments.

5.2 Communication Complexity

This subsection compares communication overhead during RCN identity verification among our TACB, ATRC [11], and Bi et al. [12]. Wlog, the group elements and $\mathbb{Z}_p$ elements are set to 32 bytes ($|\mathbb{G}|$) and 20 bytes ($|p|$), respectively. Our anonymous credential $\sigma = (u', v', w'_i, \pi_3)$ is 6 group elements (168 bytes). In ATRC [11], the credential $(path, g^{e_i}h^{w'}, attri/ra, \pi_s)$ includes a Merkle path (400 bytes for height h=20, supporting million-scale vehicles), a group element, attribute (20 bytes) and a proof (60 bytes), totaling 512 bytes. In Bi et al. [12], credential π_{auth} contains 4 group and 7 $\mathbb{Z}_p$ elements (268 bytes). So, our TACB scheme achieves the lowest communication overhead.

6 Conclusion

This paper proposes the TACB scheme, a traceable anonymous credential scheme with batch verification specifically designed for the Universal Internet of Vehicles (UIoV) environment. TACB enables rapid vehicle-side authentication without requiring expensive pairing operations, thereby

significantly reducing computational overhead. The scheme preserves user privacy by allowing vehicles to present anonymous credentials, while the Traffic Control Department (TCD) retains the ability to identify and trace malicious vehicles under proper oversight. This dual capability effectively deters potential misuse and safeguards legitimate users. Compared with existing approaches, TACB strikes an optimal balance among strong privacy protection, full traceability, and low resource consumption. Our future research will focus on two key directions. First, we aim to enhance the scalability of the tracking mechanism to support large-scale, dynamic vehicular networks, including efficient revocation and multi-authority cooperation. Second, we will improve the practical deployability of the solution by optimizing system parameters, reducing communication overhead, and integrating lightweight cryptographic primitives suited for resource-constrained onboard units. These efforts will further advance TACB toward real-world UIoV environments, ensuring both security and efficiency in next-generation intelligent transportation systems. TACB is thus a promising solution.

Acknowledgments

This work is supported by the Major Program of Shandong Provincial Natural Science Foundation for the Fundamental Research (ZR2022ZD03), the Key Research and Development Program of Shaanxi (2024GX-ZDCYL-01-15), Natural Science Basic Research Program of Shaanxi (2025JC-YBQN-814).

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] Deng Y, Zhang L, Li J. Overview of research on privacy protection of Internet of Vehicles. Application Research of Computers. 2022;39(10):2891-906.

[2] Yang H, Li Y. A blockchain-based anonymous authentication scheme for internet of vehicles. Procedia Computer Science. 2022;201:413-20.

[3] Wu T, Li G, Wang J, Xiao B, Song Y. PPCA: Privacy-Preserving Continuous Authentication Scheme with Consistency Proof for Zero-Trust Architecture Networks. IEEE Internet of Things Journal. 2025;12(11):17596-609.

[4] Zhang J, Wang X, Cui J, Li R, Zhong H. A Decentralized Threshold Credential Management with Fine-Grained Authentication for VANETs. IEEE Transactions on Information Forensics and Security. 2025;20:5818-31.

[5] Vijayakumar P, Azees M, Kozlov SA, Rodrigues JJPC. An anonymous batch authentication and key exchange protocols for 6G enabled IoT. IEEE Transactions on Intelligent Transportation Systems. 2022 Oct;23(2):1630-8.

[6] Azees M, Vijayakumar P, Deboarh LJ. EAAP: Efficient anonymous authentication with conditional privacy-preserving scheme for vehicular ad hoc networks. IEEE Transactions on Intelligent Transportation Systems. 2017 Sep;18(9):2467-76.

[7] Jiang Y, Ge S, Shen X. AAAS: An anonymous authentication scheme based on group signature in VANETs. IEEE Access. 2020 May;8:98986-98.

[8] Feng Q, He D, Zeadally S, Liang K. BPAS: Blockchain-Assisted Privacy-Preservation Authentication System for Vehicular Ad Hoc Networks. IEEE Transactions on Industrial Informatics. 2019 Oct;16(6):4146-55.

[9] Du Q, Zhou J, Ma M. EAIA: An efficient and anonymous identity-authentication scheme in 5G-V2V. Sensors. 2024 Aug;24(16):5376.

[10] Han M, Liu S, Ma S, Wan A. Anonymous-authentication scheme based on fog computing for VANET. PLoS ONE. 2020 Feb;15(2):e0228319.

[11] Liu Y, He D, Luo M, Wang H, Liu Q. ATRC: An anonymous traceable and revocable credential system using blockchain for VANETs. IEEE Transactions on Vehicular Technology. 2024 Feb;73(2):2482-94.

[12] Bi X, Yue X. Anonymous Authentication Scheme for Fog-Assisted V2G Networks. Computer Science and Application. 2024;14(6):25-31.

[13] Barreto PSLM, Naehrig M. Pairing-friendly elliptic curves of prime order. In: Selected Areas in Cryptography (SAC 2005). vol. 3897 of Lecture Notes in Computer Science. Springer; 2006. p. 319-31.

[14] Pointcheval D, Sanders O. Short Randomizable Signatures. In: Sako K, editor. Topics in Cryptology – CT-RSA 2016. vol. 9610 of Lecture Notes in Computer Science. Springer; 2016. p. 111-26.

[15] Camenisch J, Stadler M. Efficient group signature schemes for large groups. In: Advances in Cryptology - CRYPTO '97. vol. 1294 of Lecture Notes in Computer Science. Springer; 1997. p. 410-24.