



*Research article***A revocable and efficient attribute-based signature scheme from lattices for data-sharing in VANETs****Xiaopeng Zhou¹, Yang Xin², Haoyang Dang¹, Kaiyan Jin¹, Lei Shi³ and Minggang Zheng^{4,*}**¹ School of Science, Shandong Jianzhu University, Fengming Road, Jinan, 250101, China² Department of Finance of Shandong Province, Jida Road, Jinan, 250002, China³ College of Automotive Studies and Energy Engineering, Tongji University, Cao'an Highway, Shanghai, 201804, China⁴ School of Mechanical Engineering, Shandong Jianzhu University, Fengming Road, Jinan, 250101, China*** Correspondence:** Email: why1318@sdjzu.edu.cn.

Abstract: With the deep iteration of intelligent transportation systems, vehicle ad hoc networks (VANETs) have evolved into a complex ecosystem. However, open wireless channels and high mobility pose severe challenges to secure data sharing. To address issues including inadequate post-quantum security, the absence of dynamic attribute revocation, and resource constraints, this paper proposes a revocable and efficient attribute-based signature scheme from lattices for data sharing in VANETs. The scheme utilizes key-policy attribute-based signature (KP-ABS) to realize attribute authentication of vehicle nodes, and it introduces the complete subtree method for efficient user revocation. Security is proven under the ring-shortest integer solution (ring-SIS) hardness assumption in the random oracle model, achieving strong unforgeability against adaptive chosen-attribute and chosen-message attacks. Experimental results demonstrate that compared with existing schemes, the proposed scheme achieves 27.01% improvement in signing efficiency and reduces vehicle private key storage by approximately 54.17%, making it suitable for resource-constrained vehicular environments.

Keywords: attribute-based signature; authentication; lattice-based cryptography; ring-SIS; vehicle ad hoc networks (VANETs); post-quantum security

Mathematics Subject Classification: 94A60, 94A62

1. Introduction

With the deep iteration of intelligent transportation systems, vehicle ad hoc networks (VANETs) have evolved into a complex ecosystem encompassing intelligent connected vehicles, roadside

infrastructure, data service providers, and cloud platforms [1]. Data transmission is a core function of VANETs, thereby improving both the quality of driving services and the overall travel experience [2]. The VANETs communication model comprises multiple interaction paradigms: vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), vehicle-to-pedestrian (V2P), and vehicle-to-network/cloud (V2N) communications, thereby establishing a modern transportation system characterized by complete environmental perception and intelligent decision-making.

In the practical deployment of VANETs, data service providers (DSPs) play an indispensable role. By delivering diverse data services such as high-precision maps, real-time traffic condition information, traffic safety early warnings, and entertainment information to moving vehicles, they significantly improve driving safety and travel experience [3].

However, inherent characteristics of VANETs such as open wireless communication channels, high-speed movement of nodes, and dynamic changes in network topologies pose severe challenges to the secure transmission of data services [4]. For instance, unauthorized nodes may illegally obtain sensitive information including road data and entertainment data; malicious nodes can forge attribute identities to launch fake data requests. Other potential risks involve the abuse of attribute permissions and the failure to revoke expired permissions in a timely manner. These issues threaten users' privacy. They also impair the legitimate service rights and interests of data service providers. Worse still, they may compromise driving safety and undermine the stable operation of VANETs [5].

Access control is of significant theoretical and practical importance for promoting the large-scale deployment of VANETs [6]. On the premise of protecting vehicle identity privacy, an efficient and reliable message authentication mechanism needs to be established to ensure that only legally authorized vehicle nodes are permitted to access the network and obtain data services. To achieve the above security goals, digital signature technologies have been widely adopted in VANETs [7–9].

Attribute-based signature (ABS) is a special type of digital signature technology, which enables fine-grained access control over signature operations [10]. ABS is categorized into signature-policy ABS (SP-ABS) and key-policy ABS (KP-ABS). SP-ABS associates attributes with signature keys. In contrast, KP-ABS binds attributes to messages. Considering the security and efficiency requirements under the VANET environment, KP-ABS is a suitable solution that satisfies the practical demands of vehicular networks.

In big data sharing and cloud service scenarios, data interaction and privilege management supported by DSP exhibit highly dynamic characteristics. Users' authorization status, attribute information, and private key security status evolve over time along with service cycles and business updates, rather than remaining static throughout the entire service period [11]. For ABS systems deployed in practical DSP environments, dynamic privilege revocation serves as an indispensable security guarantee to enable trustworthy data sharing. Once a user's authorization expires voluntarily, the corresponding service access privileges will be proactively revoked. Alternatively, if a user's private key is leaked or stolen, the system must immediately and accurately revoke the relevant signing privileges to prevent illegal signature operations and unauthorized data access attacks [12].

The data-sharing scheme deployed in VANETs needs to meet the following security requirements:

(1) Data security: The DSP shall guarantee data confidentiality while providing data-sharing services for vehicle terminals to prevent risks and losses from data leakage. Vehicular networks communicate over open public wireless channels, so data security directly determines road traffic safety. As such, complete data security safeguards are essential during data sharing [13].

(2) User revocation: The data-sharing service shall promptly revoke users' access privileges and signature authority once user authorization expires or security compromises occur.

(3) Post-quantum security: The rapid advances in quantum computing have exposed network defense systems built on traditional cryptosystems to the risk of cracking by quantum algorithms. Given the deep integration between VANETs and wireless networks, post-quantum security defense design must be taken into consideration for VANETs [14].

1.1. Our work

To address the challenges encountered in secure data sharing within VANETs, including inadequate post-quantum security, the absence of dynamic attribute revocation mechanisms, and constraints on computational resources, we propose a revocable attribute-based signature scheme from lattices for data sharing in VANETs. The major contributions of this paper are listed as follows:

(1) We propose a revocable and efficient attribute-based signature scheme from lattices for data sharing in VANETs. We utilize KP-ABS to realize attribute authentication of vehicle nodes. DSPs can validate the legitimacy of vehicle attributes in request messages, and all transmitted data are encrypted to secure data sharing over open wireless channels.

(2) We prove that our scheme satisfies strong unforgeability under the random oracle model. It can resist both adaptive chosen-attribute attacks and adaptive chosen-message attacks. Its security foundation is the computational hardness of the ring-shortest integer solution (Ring-SIS) problem. We use lattice-based cryptographic tools in the construction. This design supports post-quantum security.

(3) Efficient user revocation is supported within our construction. Our construction introduces the complete subtree method, which achieves high efficiency in user revocation operations.

(4) We present comprehensive details of security verification and efficiency evaluation in this paper. We implement the proposed algorithm under the Java runtime environment on a hardware device with an AMD Ryzen 7 5800H processor and 16 GB memory. Comparative experimental results indicate that our scheme gains prominent advantages in both computational and storage overhead relative to some known schemes.

1.2. Related work

Data security and user privacy preservation represent core longstanding requirements for VANETs, and digital signature technology serves as the fundamental cryptographic primitive for securing vehicular communications. To balance lightweight computation, anonymous privacy protection, and fine-grained access control in resource-constrained vehicular environments, researchers have extensively investigated two mainstream signature branches: identity-based signatures (IBS) and ABS.

Early IBS and ABS constructions mostly rely on bilinear pairings over elliptic curves, whose security rests on the discrete logarithm and computational Diffie-Hellman problems. Built upon the IBS framework, the ABS primitive was first formalized by Maji et al. [15]. As representative instantiations of pairing-based ABS, the following schemes are all constructed based on elliptic curve bilinear pairings. Gu et al. [16] put forward a traceable attribute-based signature scheme to support the tracing of misbehaving signers. Lin et al. [17, 18] designed a fast and secure ABS tailored for resource-limited mobile platforms. Chen et al. [19] proposed an efficient collusion-resistant ABS scheme dedicated to secure communication in Internet of Vehicles (IoV). Gardham and Manulis [20]

investigated hierarchical ABS with minimized key size and optimal signature length for multilayered access control systems. Su et al. proposed cross-layer collaborative authentication mechanisms, offering new ideas for reducing transmission overhead in vehicular data sharing scenarios [21].

Recent advances in ABS have significantly expanded its application scenarios. Chen et al. [22] devised an attribute-based sanitizable signature with key-exposure resistance for mobile cloud data; Chen et al. [23] constructed a pairing-free ABS with message recovery for the industrial internet of things (IIoT); Li et al. [24] proposed an efficient key escrow-free ABS for anonymous access control in the IIoT; Kang et al. [25] introduced TFS-ABS, a traceable and forward-secure ABS with constant-size signatures; and Chen et al. [26] designed a server-aided verification signature to reduce the verification burden.

Al-Mekhlafi [27] presented a comprehensive survey for VANETs, summarizing the security requirements and threats in vehicular networks. In practical VANET deployment, vehicles may suffer secret key leakage, identity compromise, or expired access permissions, which makes the revocation mechanism an indispensable functional module of vehicular ABS [28, 29]. To fill this gap, a series of revocable ABS schemes built on bilinear pairings have been successively proposed. Wei et al. [30] constructed the pioneering traceable and revocable attribute-based signature (TRABS) scheme for threshold predicates. This scheme utilizes binary tree structures for efficient user revocation and non-interactive witness-indistinguishable proofs to hide signer identities. Bicer and Kupcii [31] designed a modular versatile ABS (VABS) scheme which supports threshold tracing, usage restriction, and dynamic revocation. They proved the scheme's security under standard cryptographic assumptions and implemented the scheme for efficiency evaluation. Al-Mekhlafi et al. [32] further proposed a blockchain-based, certificateless authentication scheme with privacy preservation for 5G vehicular fog computing, which eliminates reliance on centralized certificate authorities through distributed-ledger verification.

Nevertheless, schemes built on traditional cryptographic primitives face fundamental security risks in the quantum computing era [33]. Lattice cryptography, with its native post-quantum security, has been widely introduced into the design of signature primitives. A series of representative lattice-based IBS schemes have been proposed in [34, 35], aiming to realize anonymous authentication with post-quantum security. Studies [36–38] further present lattice-based ABS and IBS schemes for the same purpose.

1.3. Organizations

The remainder of this paper is organized as follows. Section 2 introduces the preliminaries, including lattice-based cryptographic definitions, the discrete Gaussian distribution, g-trapdoor mechanisms, and the complete subtree method for revocation. Section 3 formalizes the system model and the security model, defining the adversarial game for strong unforgeability. Section 4 presents the detailed construction of the proposed revocable attribute-based signature scheme, followed by its correctness analysis, security proof under the ring-SIS assumption, and discussion on resistance to network attacks. Section 5 provides a comprehensive performance evaluation, comparing the proposed scheme with existing works in terms of functional properties, space efficiency, and computational overhead. Finally, Section 6 concludes the paper and outlines future research directions.

2. Preliminaries

2.1. Notation

In this section, we give the preliminaries which are used in the scheme designs. The notations are described in Table 1.

Table 1. Notations.

Symbol	Description	Symbol	Description
$\mathbf{A}$	Matrix	$\ \cdot\ $	Euclidean norm
$\mathbf{a}$	Vector	s, σ	Gaussian parameters
$\mathbb{R}$	$\mathbb{Z}[X]/(X^n + 1)$	$\mathbf{D}_{*,*}$	Gaussian distribution
$\mathbb{R}_q$	$\mathbb{Z}_q[X]/(X^n + 1)$	T	Timestamp
H	Secure hash function	$\mathbf{u} \leftarrow \mathbb{R}$	Sample an element u randomly from the set $\mathbb{R}$
ID_*	Identity	$PRG(K)$	Pseudo-random generator
$\mathbf{e}$	Signature		

2.2. Lattice

Definition 1. (Lattice [39]) Consider a set of linearly independent vectors $\mathbf{B} = \{\mathbf{b}_1, \dots, \mathbf{b}_n\}$. The generated lattice is defined as follows:

$$\Lambda(\mathbf{B}) = \left\{ \sum_{i=1}^n x_i \mathbf{b}_i \mid x_i \in \mathbb{Z} \right\}.$$

Let $\mathbb{R} = \mathbb{Z}[X]/(X^n + 1)$ and $\mathbb{R}_q = \mathbb{Z}_q[X]/(X^n + 1)$, where n is a power of 2. For any $\mathbf{a} \in \mathbb{R}_q^m$, we define two q -ary lattices:

$$\Lambda_q^\perp(\mathbf{a}) = \{\mathbf{x} \in \mathbb{R}_q^m \mid \mathbf{a}^\top \mathbf{x} = 0 \pmod{q}\},$$

$$\Lambda_q^{\mathbf{u}}(\mathbf{a}) = \{\mathbf{x} \in \mathbb{R}_q^m \mid \mathbf{a}^\top \mathbf{x} = \mathbf{u} \pmod{q}\}.$$

Definition 2. (Gaussian distribution [39]) Let $\sigma > 0$ be a real parameter. We define the discrete Gaussian distribution over the lattice $\Lambda_q^\perp(A)$ via the expression

$$D_{\Lambda_q^\perp(A), \sigma, \mathbf{c}}(\mathbf{x}) = \frac{\rho_{\sigma, \mathbf{c}}(\mathbf{x})}{\rho_{\sigma, \mathbf{c}}(\Lambda_q^\perp(A))}.$$

In the above formulation, $\rho_{\sigma, \mathbf{c}}(\mathbf{x}) = \exp(-\pi \|\mathbf{x} - \mathbf{c}\|^2 / \sigma^2)$ represents an n -variate Gaussian function with center $\mathbf{c}$.

Definition 3. (g-trapdoor [40]) Let $\mathbf{a} \in \mathbb{R}_q^{l+k}$, $\mathbf{g} \in \mathbb{R}_q^k$, and assume a nonzero tag $\mathbf{h} \in \mathbb{R}_q$. A g-trapdoor for vector $\mathbf{a}$ is a matrix $\mathbf{R} = (\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_k) \in \mathbb{R}_q^{l \times k}$ such that

$$\mathbf{a}^T \begin{pmatrix} \mathbf{R} \\ \mathbf{I} \end{pmatrix} = \mathbf{h} \mathbf{g}^T.$$

Definition 4. (Ring-SIS [41]) Let $\mathbf{a} = \{a_1, a_2, \dots, a_m\} \in \mathbb{R}_q^m$ be a uniformly random vector, and let β represent a specified bound. The ring-SIS problem requires finding a nonzero vector $\mathbf{e}$ that belongs to the lattice $\Lambda_q^\perp(\mathbf{a})$, satisfying the constraints $\mathbf{a}^T \mathbf{e} = \mathbf{0}$ and $\|\mathbf{e}\| \leq \beta$.

Lemma 1. (GenTrap [42]) Given a lattice vector $\mathbf{a}_0 = (\mathbf{a}_1, \dots, \mathbf{a}_l)^T \in \mathbb{R}_q^l$, an element $\mathbf{h} \in \mathbb{R}_q$, and a discrete Gaussian distribution χ^{l+k} defined over the matrix space $\mathbb{R}_q^{l \times k}$, the GenTrap algorithm outputs an extended vector $\mathbf{a} = (\mathbf{a}_0^T, \mathbf{h}\mathbf{g}^T - \mathbf{a}_0^T \mathbf{R})^T \in \mathbb{R}_q^{l+k}$ alongside a trapdoor matrix $\mathbf{R} = (\mathbf{r}_1, \dots, \mathbf{r}_k) \in \mathbb{R}_q^{l \times k}$. The produced pair $(\mathbf{a}, \mathbf{R})$ satisfies the tag constraint determined by $\mathbf{h} \in \mathbb{R}_q$.

Lemma 2. (SampleD [40]) We can construct a probabilistic polynomial-time (PPT) algorithm known as SampleD. Taking parameters (n, q, l, k, σ) , a vector $\mathbf{a} \in \mathbb{R}_q^{l+k}$, an identity matrix $\mathbf{I} \in \mathbb{R}_q^{k \times k}$, a matrix $\mathbf{R} \in \mathbb{R}_q^{l \times k}$, and a target element $\mathbf{u} \in \mathbb{R}_q$ as inputs, this algorithm outputs a vector $\mathbf{e} \in \Lambda_q^\perp(\mathbf{a})$. The output vector follows a distribution statistically indistinguishable from the discrete Gaussian distribution and satisfies the norm constraint $\|\mathbf{e}\| \leq \beta$.

2.3. Key-policy attribute-based signature

A key-policy attribute-based signature (KP-ABS) scheme consists of the following four probabilistic polynomial-time (PPT) algorithms:

- $\text{Setup}(1^n) \rightarrow (pp, msk)$: On input the security parameter n , outputs the public parameters pp and the master secret key msk .
- $\text{KeyGen}(msk, f) \rightarrow sk_f$: On input msk and an access policy f , outputs a signing key sk_f that embeds the policy f .
- $\text{Sign}(pp, sk_f, M, \tau) \rightarrow \sigma$: On input pp , a signing key sk_f , a message M and an attribute set τ , outputs a signature σ if $f(\tau) = 1$, and $\perp$ otherwise.
- $\text{Verify}(pp, M, \tau, \sigma) \rightarrow \{0, 1\}$: On input pp , a message M , an attribute set τ and a signature σ , outputs 1 (accept) or 0 (reject).

In a KP-ABS scheme, the access policy is embedded in the *signing key* rather than in the signature itself: A signer holding sk_f is entitled to sign a message M only for an attribute set τ satisfying $f(\tau) = 1$. Correctness requires that for every $(pp, msk) \leftarrow \text{Setup}(1^n)$, every policy f , every $sk_f \leftarrow \text{KeyGen}(msk, f)$, every message M , and every τ with $f(\tau) = 1$, $\text{Verify}(pp, M, \tau, \text{Sign}(pp, sk_f, M, \tau)) = 1$.

2.4. Complete subtree algorithms

To achieve secure signature revocation, we integrate the complete subtree algorithms (CS), which ensures that revoked nodes will not compromise the validity of other nonrevoked nodes.

Let $\mathcal{N}$ be the universe of all system users, with total number $M = |\mathcal{N}|$, and let $\mathcal{R}$ be the subset of revoked users, with $R = |\mathcal{R}|$. The CS algorithm first builds a complete binary tree BT with M leaf nodes, where each leaf node is assigned to one system user. By inputting the revocation list $\mathcal{R}$, the algorithm computes a cover set that exactly represents the set of nonrevoked users $\mathcal{N} \setminus \mathcal{R}$. The full workflow of this cover set calculation is presented in Algorithm 1.

In the algorithm, for any tree node η , η_{LC} and η_{RC} denote its left and right child nodes, respectively, and $\text{Path}(\eta)$ denotes the set of nodes on the path from η up to the root of BT (both inclusive). RL is the

input revocation list, and the final output Y (the cover set) is the set of nodes that collectively cover all unrevoked users.

Algorithm 1 Cover set calculation algorithm $KUNodes(BT, RL)$.

```

 $X \leftarrow \emptyset, Y \leftarrow \emptyset$ 
for  $\eta \in RL$  do
     $X \leftarrow X \cup \text{Path}(\eta)$ 
end for
for  $\eta \in X$  do
    if  $\eta_{LC} \notin X$  then
         $Y \leftarrow Y \cup \{\eta_{LC}\}$ 
    end if
    if  $\eta_{RC} \notin X$  then
         $Y \leftarrow Y \cup \{\eta_{RC}\}$ 
    end if
end for
if  $Y = \emptyset$  then
     $Y \leftarrow \{\text{root}\}$ 
end if
return  $Y$ 

```

3. Problem formulation

3.1. System model

In this section, we describe the proposed data sharing scheme. This scheme consists of three entities: the trusted authority (TA), DSPs, and vehicle nodes.

TA: The TA serves as the primary registration entity possessing maximum privileges in the system. Every vehicle and DSP involved in the data-sharing framework receives its private key upon finishing registration at the TA side. It is assumed that secure communication links exist connecting the TA, vehicles, and DSPs.

Data service provider (DSP): The DSP stores data required by vehicles and shares authorized data to vehicle nodes. Each DSP registers with the TA to obtain its private key, and it is a semi-trusted entity.

Vehicle nodes: Vehicle nodes register their attribute information with the TA to acquire attribute private keys, and their trustworthiness is not assumed. Within the data sharing system, vehicle nodes can securely send data requests to the DSPs.

We adopt the KP-ABS primitive to support authentication in X2C communications. For an attribute vector $\tau = (\tau_1, \tau_2, \dots, \tau_k)$, we define the attribute policy function as $f(\tau_1, \tau_2, \dots, \tau_k) = t_0 + t_1\tau_1 + t_2\tau_2 + \dots + t_k\tau_k \pmod{2}$, where each coefficient $t_i \in \{0, 1\}$. As demonstrated in [43], this function can express arbitrary arithmetic circuits and serves to verify whether an attribute vector satisfies the predefined access policy.

The proposed attribute-based data sharing scheme (KP-ABS) comprises five algorithms: Setup, Key generation, Service broadcast, Vehicle request and data-sharing. The scheme model is shown in

Figure 1, and the specific process is as follows:

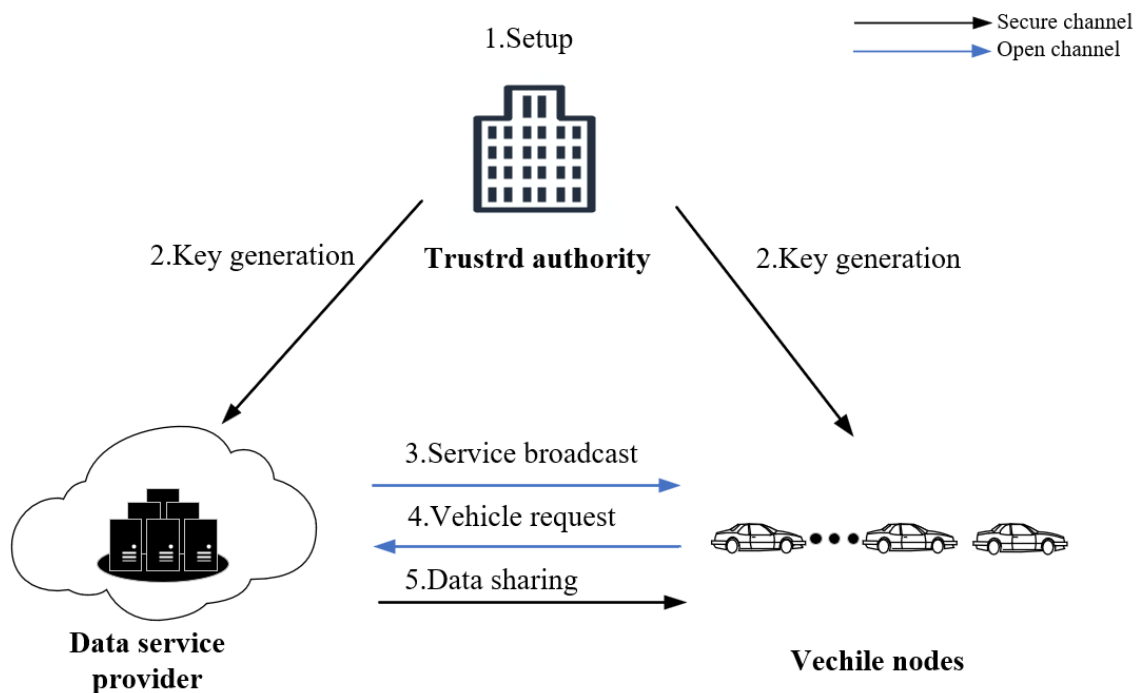


Figure 1. System architecture of the proposed scheme.

(1) **Setup:** Within this stage, the TA receives the security parameter n and extra public parameters as inputs for system parameter generation. Subsequently, the TA leverages the *GenTrap* algorithm presented in Lemma 1 to construct its public and secret key pair (pk_{TA}, sk_{TA}) .

(2) **Key generation:** Within this stage, DSPs and vehicle nodes complete registration with the TA, which generates the DSP private key sk_{ID_i} and vehicle private key sk_{ID_v} for them, respectively. During vehicle registration, given a policy function f , matching attribute vector τ , and system parameters $param$, the TA generates the private signing key sk_{ID_v} for the vehicle. Here, the TA organizes the vehicles as the leaves of a complete binary tree, and using the complete-subtree method, assigns each vehicle an identity matrix S_i together with the node matrices S_η on its leaf-to-root path, which are later used for revocation.

(3) **Service broadcast:** The DSP signs the data list (DL) using its private key sk_{ID_i} to generate a signature e_{DSP} , then broadcasts e_{DSP} together with the DL.

(4) **Vehicle request:** When the vehicle needs data contained in the DL, it first generates a symmetric key K , then encrypts K by utilizing the DSP's public key to generate EK . Afterwards, the vehicle signs the message M using its private key sk_{ID_v} . Finally, the vehicle sends the signed message $\{e, M, EK, \tau, h, T_i\}$ to the DSP.

(5) **Data-sharing:** After the DSP receives the vehicle's data request message $\{e, M, EK, \tau, h, T_i\}$, the DSP verifies the message. When processing a signature, the DSP first checks whether $f(\tau) = 1$ and verifies that the corresponding entity is not contained in the revocation list. If the condition holds, it invokes the Verify algorithm from the KP-ABS scheme to complete signature verification. In the case $f(\tau) = 0$, the CSP outputs $\perp$. For a valid request, the DSP decrypts EK to obtain the symmetric key K ,

then encrypts the data required by the vehicle with this symmetric key and transmits the encrypted data to the vehicle.

3.2. Security model

The security of our proposed scheme is defined based on the strong unforgeability of KP-ABS. We formalize the corresponding adversarial game under the random oracle model to define the rigorous security criteria, where C denotes the polynomial-time challenger, and $\mathcal{A}$ represents the probabilistic polynomial-time (PPT) adversary. We formalize a security Game. If the winning probability of any PPT adversary in Game is negligible, we can prove that the proceeds scheme achieves strong unforgeability against adaptive chosen-message attacks and adaptive chosen-attribute attacks. It releases the challenge attribute τ^* , and the challenger C together with the adversary $\mathcal{A}$ proceed to carry out the subsequent Game. This Game is defined as follows:

(1) Setup: The challenger C executes the system Setup algorithm to output global system public parameters $param$, together with the trusted authority's public/secret key pair (pk_{TA}, sk_{TA}) . Afterwards, C transmits all public materials $(param, pk_{TA})$ to the adversary $\mathcal{A}$.

(2) Hash queries: The adversary $\mathcal{A}$ adaptively issues hash queries to H_1, H_2, H_3, H_4, H_5 , and the challenger C responds to each query accordingly.

(3) Private key extract queries: The adversary $\mathcal{A}$ adaptively makes private key queries for attribute function $(h, f(\tau))$, and the challenger C generates a verifiable secret key sk_{ID_v} as the response.

(4) Signature queries: The adversary $\mathcal{A}$ adaptively queries signatures for attribute τ and message M , and the challenger C generates the corresponding signature as the reply.

(5) Forgery: After finishing all queries, the PPT adversary $\mathcal{A}$ produces a counterfeit signature e^* bound to the target plaintext M^* and target attribute τ^* . We split the forgery scenario into two cases: Case 1: $\mathcal{A}$ has previously acquired a legitimate signature e for the combination (M^*, τ^*) , and the forged signature satisfies $e \neq e^*$.

Case 2: The tuple (M^*, τ^*) has never been submitted to the signature generation oracle by $\mathcal{A}$ in all prior query steps.

The adversary $\mathcal{A}$ is regarded as the winner of Game if the forged signature e^* can pass the verification. The adversarial advantage is formally defined as the probability that $\mathcal{A}$ wins the above security Game.

4. Revocable attribute-based data sharing scheme

We present a detailed description of the proposed approach in this section.

4.1. Construction

Given an attribute vector $\tau = (\tau_1, \dots, \tau_k)^T$ over $\{0, 1\}$, we define a policy function $f(\tau) = t_0 + \sum_{i=1}^k t_i \tau_i \pmod{2}$ and reconstruct it as $g(x) = t_0(1, \dots, 1)^T + \sum_{i=1}^k t_i x_i \pmod{q}$ over $\mathbb{R}_q^k$, where f and g are bijective. Using $g(x)$, we generate attribute-matching private keys.

(1) **Setup:** Taking as input the security parameter n and other public parameters q, l, k, σ , and so forth. as specified in Lemma 1, the system parameters are as follows:

(a) Set five secure hash functions: $H_1 : \mathbb{R}_q^{k \times k} \rightarrow \mathbb{R}_q, H_2 : \mathbb{R}_q \times \{0, 1\}^* \times \{0, 1\}^* \rightarrow \mathbb{R}_q^k, H_3 : \{0, 1\}^* \rightarrow \mathbb{R}_q, H_4 : \mathbb{R}_q^{k \times k} \times \{0, 1\}^* \rightarrow \mathbb{R}_q^{k \times k}, H_5 : \{0, 1\}^* \rightarrow \mathbb{R}_q$.

(b) The TA generates a vector $\mathbf{g} = (1, 2, 4, \dots, 2^{k-1})^T \in \mathbb{R}_q^k$. The GenTrap algorithm is run to output $\mathbf{a} \in \mathbb{R}_q^{l+k}$ and $\mathbf{R} \in \mathbb{R}_q^{l \times k}$ by inputting $\mathbf{g} \in \mathbb{R}_q^k$, $\mathbf{I} \in \mathbb{R}^{k \times k}$, $\mathbf{h} \in \mathbb{R}_q$, and $param$, satisfying $\mathbf{a}^T \begin{pmatrix} \mathbf{R} \\ \mathbf{I} \end{pmatrix} = \mathbf{h}\mathbf{g}^T$. Then, $pk_{TA} = (\mathbf{a}, \mathbf{g})$, and $sk_{TA} = \mathbf{R}$.

(2) **Key generation:** We utilize the complete subtree algorithms to implement the revocation function of the scheme. The TA stores user index information in a complete binary tree with M leaf nodes. Furthermore, the TA selects a random identity matrix $\mathbf{S}_i \in \mathbb{R}_q^{k \times k}$ for each leaf node and a random matrix $\mathbf{S}_\eta \in \mathbb{R}_q^{k \times k}$ for each node $\eta \in BT$.

At this stage, the TA generates private keys $\{sk_{ID_v}, sk_{ID_i}\}$ for vehicles and the DSP, respectively. For vehicles whose attribute set τ satisfies the policy function $f(\tau)$, the TA generates the identity matrix $\mathbf{S}_i$ and the private key $\mathbf{R}_i$ for each vehicle. The detailed process is as follows:

1. For vehicles:

(a) The TA first computes $H_1(\mathbf{S}_i)$. Furthermore, h serves as the identity identifier of the vehicle, where $h = H_1(\mathbf{S}_i)$.

(b) Then, TA computes $H_2(h|i|T)$, $i \in \{1, \dots, k\}$ and $\mathbf{tag} = t_0(1, \dots, 1)^T + \dots + t_k H_2(h|i|T)(mod q)$. Here, T denotes the valid authentication time of the vehicle and $tag \in \mathbb{R}_q^k$. (Notice: The $\mathbf{tag}$ is derived from the modified policy function $g(x)$. The vector $\mathbf{tag} = t_0(1, \dots, 1)^T + \sum_{i=1}^k t_i H_2(h|i|T) \in \mathbb{R}_q^k$ is the target vector of SampleD: The key $\mathbf{R}_i$ is generated to satisfy $\mathbf{a}^T \mathbf{R}_i = \mathbf{tag}^T$. Hence, $\mathbf{tag}$ cryptographically binds the signing key to the policy coefficients $\mathbf{t} = (t_0, \dots, t_k)$, the vehicle identity $h = H_1(\mathbf{S}_i)$ through $H_2(h|i|T)$, and the validity period T .)

(c) Finally, the TA computes $\mathbf{R}_i \leftarrow SampleD(\mathbf{a}, \mathbf{R}, I, \mathbf{tag}, \sigma)$ as the private key of vehicle. ($\mathbf{a}^T \mathbf{R}_i = \mathbf{tag}^T$, Lemma 2)

2. For the DSP:

(a) The TA computes $H_3(ID_i)$ for ID_i .

(b) Then, the TA computes $\mathbf{R}_{DSP} \leftarrow SampleD(\mathbf{a}, \mathbf{R}, I, H_3(ID_i)g, \sigma)$ as the private key of DSP. ($\mathbf{a}^T \mathbf{R}_{DSP} = H_3(ID_i)g^T$, Lemma 2)

(3) **Service broadcast:** At this stage, the DSP provides a data list (DL), signs it, and broadcasts the signature, so as to provide data services for vehicles. The detailed process is as follows:

(a) The DSP first computes $H_3(ID_i)$, and then calculates $\mathbf{a}_{DSP} = (\mathbf{a}^T | H_3(ID_i)g^T - g^T)^T$.

(b) The DSP calculates $H_5(DL)$, such that $w = H_5(DL)$.

(c) The DSP computes $\mathbf{e}_{DSP} \leftarrow SampleD(\mathbf{a}_{DSP}, \mathbf{R}_{DSP}, H_3(ID_i), w, \sigma)$.

(d) Then, the DSP broadcasts $\{DL, \mathbf{e}_{DSP}, T'\}$. T' denotes the timestamp of the current DSP message.

(4) **Vehicle request:** When a vehicle needs to acquire service data, it will send a request to the DSP using V2X communication after reading and verifying the broadcast message. Prior to this, the vehicle must determine whether its attributes satisfy the access structure. If $f(\tau) \neq 1$, or $h \in RL$, the algorithm outputs $\perp$. Otherwise, the vehicle retrieves the matrix $\mathbf{S}_\eta$, where η is the unique node in $Path(h) \cap KUNodes(BT, RL)$. To ensure data security during communication, the vehicle selects a symmetric encryption key $K \in \{0, 1\}^{nk}$. Accordingly, K is encrypted using the PK_{DSP} to produce the ciphertext EK . The plaintext message M , EK , and timestamp T_i are jointly fed into a cryptographic hash algorithm, which outputs $\tilde{\mathbf{m}} = (m_1, m_2, \dots, m_p)^T \in \{0, 1\}^P$.

(a) The vehicle verifies that the timestamp T_i of the broadcast message $\{DL, \mathbf{e}_{DSP}, T'\}$ is fresh, and checks the broadcast message $\{DL, \mathbf{e}_{DSP}, T'\}$ by $\mathbf{a}_{DSP} \mathbf{e}_{DSP} \stackrel{?}{=} w$.

(b) The vehicle randomly selects a symmetric key $K \in \{0, 1\}^{nk}$ and encrypts it using the public key PK_{DSP} to get EK .

(c) Computes $H_2(h, i, T)$ for $i \in \{1, 2, \dots, k\}$, with T representing the validity timeframe for the vehicle attribute.

(d) Computes $\mathbf{tag} = t_0(1, \dots, 1)^T + \dots + t_k H_2(h|i|T)(\text{mod } q)$ by utilizing the modified function $g(x)$.

(e) Computes $H_4(S_\eta|\tau)$ and $\mathbf{a}_s = (\mathbf{a}^T | g^T H_4(S_\eta|\tau) - \mathbf{tag}^T)^T$.

(f) Computes $H_5(\tau|i|T_i)$ for $i \in \{1, 2, \dots, p\}$, where T_i denotes the current timestamp. Then, the vehicle computes $\mathbf{c} = \mathbf{a}^T \mathbf{u} + \sum_{i=1}^p m_i H_5(\tau|i|T_i) \pmod{q}$, with $\mathbf{u} \in \mathbb{R}_q^{l+k}$ sampled from the Gaussian distribution.

(g) Computes $\mathbf{e}' \leftarrow \text{SampleD}(\mathbf{a}_s, \mathbf{R}_i, h, c, \sigma)$ and $\mathbf{e} = \mathbf{e}' - (\mathbf{u}|0)$.

Finally, the vehicle broadcasts $\{\mathbf{e}, M, EK, \tau, h, T_i\}$.

Remark 1. Each $m_i \in \{0, 1\}$ is a single bit, obtained by hashing the message M (together with EK, T_i) into a p -bit string $\tilde{\mathbf{m}} = (m_1, \dots, m_p)^T$ via a hash $\{0, 1\}^* \rightarrow \{0, 1\}^p$. For the subsequent ring operations, each bit is canonically embedded into $\mathbb{R}_q$ as the constant polynomial m_i (i.e., the coefficient vector $(m_i, 0, \dots, 0)$), so that

$$m_i H_5(\tau|i|T_i) = \begin{cases} H_5(\tau|i|T_i), & m_i = 1, \\ 0, & m_i = 0. \end{cases}$$

(5) **Data-sharing:** After the DSP receives the vehicle's data request message $\{\mathbf{e}, M, EK, \tau, h, T_i\}$, the DSP verifies the message. It first computes $f(\tau)$, and if $f(\tau) = 0$, the protocol terminates; if $f(\tau) = 1$ and h has not been revoked, the DSP proceeds to the following verification steps.

(a) Computes $H_2(h|i|T), i \in \{1, \dots, k\}$ and $\mathbf{tag} = t_0(1, \dots, 1)^T + \dots + t_k H_2(h|i|T)^T$.

(b) Computes $H_4(S_\eta|\tau)$ and $\mathbf{a}_s = (\mathbf{a}^T | g^T H_4(S_\eta|\tau) - \mathbf{tag}^T)^T$.

(c) Computes $H_5(\tau|i|T_i)$ for $i \in \{1, 2, \dots, p\}$ and $\tilde{\mathbf{m}} = (m_1, \dots, m_p)^T$. Then, the DSP computes $\sum_{i=1}^p m_i H_5(\tau|i|T_i) \pmod{q}$.

(d) If $\mathbf{a}_s^T \mathbf{e} = \sum_{i=1}^p m_i H_5(\tau|i|T_i) \pmod{q}$ and $\|\mathbf{e}\| \leq \sigma \sqrt{l+2k} \pmod{q}$, the signature is accepted.

(e) After the signature is verified to be valid, the DSP decrypts EK with its private key sk_{ID_i} to retrieve the symmetric key K .

(f) The DSP encrypts the vehicle requested data ND_i by $E_{ND_i} = \text{PRG}(K) \oplus ND_i \pmod{2}$, and then transmits the tuple (E_{ND_i}, T_i) to the target vehicle.

Remark 2. After the tuple (E_{ND_i}, T_i) arrives, the requesting vehicle recovers the requested data by computing $ND_i = \text{PRG}(K) \oplus E_{ND_i}$. Recall that K was generated by the vehicle itself and was transmitted to the DSP only in the encrypted form EK . Consequently, only the DSP holding sk_{ID_i} can recover K and generate the matching keystream $\text{PRG}(K)$. In this process, the vehicle can authenticate that the received data ND_i really come from the DSP, as no party without sk_{ID_i} can obtain K or produce a valid E_{ND_i} . Moreover, K is freshly sampled for each request and erased after the session, so keystreams of different sessions are independent, and the timestamp T_i , checked within a predefined freshness window, excludes replayed data.

4.2. Correctness

The correctness of the signature verification procedure is established on the following foundations:

1. During the key-generation phase, the TA utilizes the SampleD algorithm in conjunction with the g-trapdoor master secret key $\mathbf{R}$ to derive private keys for both vehicle nodes and the DSP. According to [40], we set $q = 2^k, k = \lceil \log q \rceil, l = 5, \bar{m} = 2n$, and the Gaussian parameter satisfies $\sigma \geq s \sqrt{l+k}$.

2. During the vehicle-request phase, the private key $\mathbf{R}_i$ of the vehicle can be seemed as a g-trapdoor of $\mathbf{a}_s$. Specifically, for $\mathbf{a}_s = (\mathbf{a}^T | g^T H_4(\mathbf{S}_\eta | \tau) - \mathbf{tag}^T)^T$, $\mathbf{a}_s^T \begin{pmatrix} \mathbf{R}_i \\ \mathbf{I} \end{pmatrix} = g^T H_4(\mathbf{S}_\eta | \tau)$.

3. The SampleD algorithm operates correctly in the vehicle-request phase. The private key $\mathbf{R}_i$ can be leveraged to sample signature vectors from the discrete Gaussian distribution $D_{\Lambda^\perp(\mathbf{a}_s), \sigma}$. Following the parameterization convention in [40], we set $q = 2^k$ with $k = \lceil \log q \rceil$, $l = 5$, $\bar{m} = 2n$, and $\sigma \geq s \sqrt{l+k}$.

4. Building on the security analysis above, given the core security parameter n , we instantiate all scheme parameters as follows: The modulus $q = 2^k$ with $k = \lceil \log q \rceil$, the trapdoor dimension $l = 5$, the vector dimension $\bar{m} = 2n$, the smoothing parameter bound $s \geq \omega(\sqrt{\log n})$, and the Gaussian sampling parameter satisfying $\sigma \geq s \sqrt{l+k}$.

Given the parameters defined above, we can prove the correctness of the Data-sharing phase. We know that $\mathbf{a}_s = (\mathbf{a}^T | g^T H_4(\mathbf{S}_\eta | \tau) - \mathbf{tag}^T)^T$, $\mathbf{c} = \mathbf{a}^T \mathbf{u} + \sum_{i=1}^p m_i H_5(\tau | i | T') \pmod{q}$ and $\mathbf{a}_s^T \mathbf{e}' = \mathbf{c}$, so we can verify the equation:

$$\begin{aligned} \mathbf{a}_s^T \mathbf{e} &= \mathbf{a}_s^T (\mathbf{e}' - (\mathbf{u} | \mathbf{0})) = \mathbf{c} - \mathbf{a}_s^T (\mathbf{u} | \mathbf{0}) \\ &= \mathbf{c} - ((\mathbf{a}^T | g^T H_4(\mathbf{S}_\eta | \tau) - \mathbf{tag}^T)(\mathbf{u} | \mathbf{0})) \\ &= \mathbf{a}^T \mathbf{u} + \sum_{i=1}^p m_i H_5(\tau | i | T') - \mathbf{a}^T \mathbf{u} \\ &= \sum_{i=1}^p m_i H_5(\tau | i | T') \pmod{q}. \end{aligned}$$

4.3. Security proof

Theorem 1. *Under the random oracle model, assuming the hardness of the ring-SIS problem (i.e., no probabilistic polynomial-time (PPT) algorithm can solve the ring-SIS problem with non-negligible probability), the scheme proposed in this paper achieves strong unforgeability against adaptive chosen-message attacks and chosen-attribute attacks.*

Specifically, if there exists a PPT adversary $\mathcal{A}$ that can successfully forge a valid attribute-based signature with advantage at least ϵ after making q_1 H_1 hash queries, q_2 H_2 hash queries, q_3 H_3 hash queries, q_4 H_4 hash queries, q_5 H_5 hash queries, q_6 private key extract queries, and q_7 signature queries, then the challenger $\mathcal{C}$ can obtain a solution to the ring-SIS problem with probability $\Pr[\text{Success}] \geq (1 - 2^{-\omega(\log n)})\epsilon$.

Proof. The challenger $\mathcal{C}$ is given a ring-SIS problem instance consisting of parameters $(\mathbf{a} \in \mathbb{R}_q^{l+2k}, q, n, k, \sigma)$, and its objective is to find a vector $\mathbf{e} \in \mathbb{R}_q^{l+2k}$ that meets two critical conditions: $\|\mathbf{e}\| \leq 2\sigma \sqrt{l+2k} \pmod{q}$ and $\mathbf{a}^T \mathbf{e} = 0 \pmod{q}$. We formalize a security game Game between the adversary $\mathcal{A}$ and the challenger $\mathcal{C}$. Let L_1 – L_7 be seven lists used to store the responses of H_1 – H_5 hash queries, private key extraction queries, and signature queries, respectively. The Game proceeds as follows:

(1) **Setup:** The challenger $\mathcal{C}$ constructs $\mathbf{a} = (\mathbf{a}_0 | \mathbf{a}_1)$, where $\mathbf{a}_0 \in \mathbb{R}_q^{l+k}$, and $\mathbf{a}_1 \in \mathbb{R}_q^k$. The remaining setup parameters are produced in accordance with the procedures illustrated in the Setup section. $\mathcal{C}$ transmits the tuple $(\text{param}, pk_{TA} = \mathbf{a}_0)$ to the adversary $\mathcal{A}$. Before performing the hash query, the challenger $\mathcal{C}$ first selects an attribute t_{j^*} such that $f(t_{j^*}) = 1$.

(2) **Hash H_1 queries:** Adversary $\mathcal{A}$ makes q_1 times H_1 queries. For the Hash H_1 query of $\mathbf{S}_i$, C randomly chooses a $S_i \leftarrow D_{\mathbb{R}_q^{k \times k}, \sigma}$ and computes $h = H_1(\mathbf{S}_i)$ as answer. Then, C stores $(h, \mathbf{S}_i)$ to list L_1 . Whenever a repeated query is detected, C delivers the previously stored result back to the adversary.

(3) **Hash H_2 queries:** Without loss of generality, we assume that the adversary $\mathcal{A}$ completes the corresponding H_1 hash queries prior to issuing each H_2 query. The adversary $\mathcal{A}$ can issue up to q_2 hash queries to the oracle H_2 . When $\mathcal{A}$ transmits the tuple $(h|i|T)$ to C for an H_2 query, C searches for $(h|i|T)$ within the list L_2 and returns the value $H_2(h|i|T)$ to $\mathcal{A}$. If the query is newly submitted, C executes the following operations:

Case 1: If $i = j^*$, C randomly generates $\mathbf{R}_{j^*} \leftarrow D_{\mathbb{R}_q^{(l+k) \times k}, \sigma}$ and computes $H_2(h|i|T)^T = \mathbf{a}_0^T \mathbf{R}_{j^*} + t_0(1, \dots, 1)^T \pmod{q}$. Then, C sends it to $\mathcal{A}$ as answer and stores it to list L_2 .

Case 2: If $i \neq j^*$, C generates $\mathbf{R}_i \leftarrow D_{\mathbb{R}_q^{(l+k) \times k}, \sigma}$ and computes $H_2(h|i|T)^T = \mathbf{a}_0^T \mathbf{R}_i \pmod{q}$ as the answer. Here, the Gaussian parameter σ exceeds the smoothing parameter of $\Lambda_q^\perp(\mathbf{a}_0)$. Then, C sends it to $\mathcal{A}$ and stores $(h, i, T, \mathbf{R}_i, H_2(h|i|T))$ to list L_2 .

(4) **Hash H_3 queries:** Without loss of generality, when $\mathcal{A}$ issues an H_3 query, we assume it has completed the corresponding H_2 queries. At this stage, $\mathcal{A}$ inputs (ID_i) to C , and C outputs $H_3(ID_i)$. C computes $H_3(ID_i) \leftarrow D_{\mathbb{R}_q, \sigma}$ as the answer. Then, C stores it to list L_3 .

(5) **Hash H_4 queries:** Without loss of generality, when $\mathcal{A}$ issues an H_4 query, we suppose that the corresponding H_3 queries have already been completed. Prior to this step, the challenger C first searches $H_2(h|k|T)$ from list L_2 and computes the **tag** $= t_0(1, \dots, 1)^T + \dots + t_k H_2(h|k|T)$. Then, the adversary $\mathcal{A}$ inputs $(\mathbf{S}_\eta, \tau)$ to the challenger C , and C returns $H_4(\mathbf{S}_\eta|\tau)$, where $H_4(\mathbf{S}_\eta|\tau)^T g = \mathbf{a}_1 + \mathbf{tag} \pmod{q}$. Subsequently, C stores the tuple $(H_4(\mathbf{S}_\eta|\tau), h, T, \mathbf{S}_\eta, \tau, \mathbf{tag})$ to list L_4 .

(6) **Hash H_5 queries:** Without loss of generality, when $\mathcal{A}$ submits an H_5 query, we assume that it has completed the corresponding H_4 queries. When the adversary $\mathcal{A}$ issues a query on (τ, i, T_i) , the challenger C retrieves the value $H_5(\tau|i|T_i)$ from list L_5 as the answer. There are two cases as follows:

Case 1: If $i = j^*$, C randomly generates $\mathbf{e}_{j^*} \leftarrow D_{\mathbb{R}_q^{l+2k}, \sigma}$ and computes $H_5(\tau|i|T_i) = \mathbf{a}^T \mathbf{e}_{j^*} - \mathbf{a}_0^T \mathbf{u}_i \pmod{q}$ where $\mathbf{u}_i \leftarrow D_{\mathbb{R}_q^{l+k}, \sigma}$. Then, C sends it to $\mathcal{A}$ as answer and stores it to list L_5 .

Case 2: If $i \neq j^*$, C generates $\mathbf{e}_i \leftarrow D_{\mathbb{R}_q^{l+2k}, \sigma}$ and computes $H_5(\tau|i|T_i) = \mathbf{a}^T \mathbf{e}_i \pmod{q}$ as the answer. Here, the Gaussian parameter σ exceeds the smoothing parameter of $\Lambda_q^\perp(\mathbf{a}_0)$. Then, C sends it to $\mathcal{A}$ and stores the tuple $(H_5(\tau|i|T_i), \tau, i, T_i, \mathbf{e}_i)$ to list L_5 .

Private key extract queries: When the adversary $\mathcal{A}$ issues a private key query on $(h, f(\tau))$, the challenger C retrieves all corresponding $\mathbf{R}_i$ from list L_2 and computes $\mathbf{R} = t_1 \mathbf{R}_1 + \dots + t_k \mathbf{R}_k \pmod{q}$. Then, C sends it to $\mathcal{A}$ and stores the tuple $(h, f(\tau), \mathbf{R})$ to list L_6 . Prior to this, $\mathcal{A}$ has already completed the H_2 hash queries and can verify that $\mathbf{a}^T \mathbf{R} = t_1 H_2(h|1|T)^T + \dots + t_k H_2(h|k|T)^T \pmod{q}$ and that $\|\mathbf{R}\| \leq \|\mathbf{R}_1\| + \dots + \|\mathbf{R}_k\| \leq \sigma \sqrt{l+k} \pmod{q}$.

Signature queries: During this phase, the adversary $\mathcal{A}$ requests the signature for the requested message M with vehicle attribute τ in an adaptive manner. There are two cases need to be considered:

Case 1: If $f(\tau) = 1$, C queries $\mathbf{e}_i$ from list L_5 and computes $\mathbf{e} = \sum_{i=1}^p m_i \mathbf{e}_i \pmod{q}$ as the answer. From Definition [40], it is known that the distribution of $\mathbf{e}$ is consistent with the genuine signature.

Case 2: If $f(\tau) \neq 1$, C queries $H_5(\tau|i|T_i)$ from list L_5 and queries $\mathbf{R}_i$ from list L_2 . Then, C calculates $\mathbf{c} = \sum_{i=1}^p m_i H_5(\tau|i|T_i) \pmod{q}$, and uses $\text{SampleD}(\mathbf{a}, \mathbf{R}, h, \mathbf{c}, \sigma) \rightarrow \mathbf{e}$. Then C stores the tuple $(\tau, h, M, \mathbf{e})$ to list L_7 .

Forgery: After finishing all of its queries, the adversary $\mathcal{A}$ forges a message M representing the vehicle with attribute τ . We consider the following two cases:

Case 1. If $\mathcal{A}$ forges a signature $\mathbf{e}$ on message M^* for attribute τ^* , and $\mathbf{e}^*$ has been queried in list L_7 where $\mathbf{e} \neq \mathbf{e}^*$, then C can obtain a solution to the ring-SIS problem. Here, $\mathbf{a}^T(\mathbf{e}^* - \mathbf{e}) = 0(\text{mod } q)$, and $\|\mathbf{e}^* - \mathbf{e}\| \leq 2\sigma \sqrt{l+2k}$.

Case 2. If the signature $\mathbf{e}$ on message M^* for attribute τ^* has never been queried, C queries $\mathbf{e}_i$ from L_5 and computes $\mathbf{e} = \sum_{i=1}^p m_i \mathbf{e}_i$, where $\|\mathbf{e}\| \leq \frac{p}{2} \|\mathbf{e}_i\| \leq \sigma \sqrt{l+2k}$. Then, the probability of $\mathbf{e}^* \neq \mathbf{e}$ is larger than $1 - 2^{-\omega(\log n)}$. Similar to the analysis for Case 1, C obtains *ring-SIS* solution.

Analysis:

1. Based on the following analysis, the simulation implemented by C is statistically indistinguishable from the Game.
2. In Hash H_1 queries phase, since $\mathbf{S}_i \leftarrow D_{\mathbb{R}_q^{k \times k}, \sigma}$, $(h, \mathbf{S}_i)$ is random and uniform.
3. In Hash H_2 queries, since matrices $\mathbf{R}_i \leftarrow D_{\mathbb{R}_q^{(l+k) \times k}, \sigma}$ are sampled uniformly at random, and the Gaussian parameter σ exceeds the smoothing parameter of lattice $\Lambda_q^\perp(\mathbf{a}_0)$; the value $\mathbf{a}_0^T \mathbf{R}_i$ approximates the uniform distribution. Then, $H_2(h|i|T)$ is random [40].
4. In Hash H_3 queries phase, $H_3(ID_i) \leftarrow D_{\mathbb{R}_q}$. Because the Gaussian parameter σ exceeds the smoothing parameter of lattice $\Lambda_q^\perp(\mathbf{a}_0)$, $H_3(ID_i)$ is statistically uniform and random [40].
5. In Hash H_4 queries phase, $\mathbf{a}_1$ and tag are random, and $H_4(\mathbf{S}_\eta|\tau)$ obtained from $H_4(\mathbf{S}_\eta|\tau)^T g = \mathbf{a}_1 + \text{tag}(\text{mod } q)$ is random.
6. In Hash H_5 queries phase, $\mathbf{e}_i \leftarrow D_{\mathbb{R}_q^{l+2k}, \sigma}$ is random, and $H_5(\tau|i|T_i)$ obtained from $H_5(\tau|i|T_i) = \mathbf{a}^T \mathbf{e}_i(\text{mod } q)$ is random.
7. In private key extract queries phase, since $\mathbf{R}_i \leftarrow D_{\mathbb{R}_q^{(l+k) \times k}, \sigma}$, and $\mathbf{R}$ obtained from $\mathbf{R} = t_1 \mathbf{R}_1 + \dots + t_k \mathbf{R}_k(\text{mod } q)$ is random.
8. During the signature query phase, $\mathbf{e}_i$ is sampled directly from $D_{\mathbb{R}_q^{l+2k}, \sigma}$ under Case 1. For Case 2, $\mathbf{e}$ is generated by the *SampleD* algorithm, whose output distribution approximates the Gaussian distribution. Therefore, Cases 1 and 2 are statistically indistinguishable.

The challenger C successfully solves the ring-SIS problem if and only if the following three independent events all occur.

E_1 : $\mathcal{A}$ outputs a valid forgery:

$$\Pr[E_1] = \epsilon.$$

E_2 : The forged signature satisfies $\mathbf{e}^* \neq \mathbf{e}$ (so that $\mathbf{e}^* - \mathbf{e} \neq \mathbf{0}$ is a valid ring-SIS solution):

$$\Pr[E_2] \geq 1 - 2^{-\omega(\log n)}.$$

E_3 : The simulation does not fail. Each of the $q_1 + \dots + q_7$ queries introduces a statistical error of at most $2^{-\omega(\log n)}$ from the discrete Gaussian sampling and *SampleD*; by the union bound,

$$\Pr[E_3] \geq 1 - (q_1 + \dots + q_7) 2^{-\omega(\log n)}.$$

Then, we obtain

$$\Pr[\text{Success}] \geq \epsilon \cdot (1 - 2^{-\omega(\log n)}) \cdot (1 - (q_1 + \dots + q_7) 2^{-\omega(\log n)}) = (1 - 2^{-\omega(\log n)}) \epsilon,$$

where the last equality holds because $q_1, \dots, q_7$ are all $\text{poly}(n)$, and hence, $(q_1 + \dots + q_7) 2^{-\omega(\log n)} = 2^{-\omega(\log n)}$.

According to [39], we get the probability of C to solve *ring-SIS* problem as $\Pr[\text{Success}] \geq (1 - 2^{-\omega(\log n)}) \epsilon$. Therefore, this probability cannot be ignored.

4.4. Mutual message authentication

Vehicles can verify the legitimacy of the DSP by checking the validity of the verification equation $\mathbf{a}_{DSP} \mathbf{e}_{DSP} = w$. The DSP can also check the equation $\mathbf{a}_s^T \mathbf{e} = \sum_{i=1}^p m_i H_5(\tau|i|T') \pmod{q}$ to verify the legitimacy of vehicle request messages. Therefore, we have implemented the mutual authentication between the vehicle and the DSP.

4.5. Resistance to network attacks

Man-in-the-middle attack: It is evident that our scheme requires joint participation of the TA, the DSP and the vehicle. Due to the hardness of the ring-SIS problem, adversaries are unable to forge signatures without the private key. Accordingly, man-in-the-middle attacks pose no threat to our scheme.

Replay attack: In our scheme, we introduce timestamps T' and T_i into the message $\{DL, \mathbf{e}_{DSP}, T'\}$ broadcast by the DSP and the vehicle request message $\{\mathbf{e}, M, EK, \tau, h, T_i\}$, respectively. Therefore, our scheme can effectively resist replay attacks.

Impersonation and tampering attacks: In our scheme, forging a legitimate signed message requires satisfying the verification equation $\mathbf{a}_s^T \mathbf{e} = \sum_{i=1}^p m_i H_5(\tau|i|T_i) \pmod{q}$ and $\|\mathbf{e}\| \leq \sigma \sqrt{l+2k} \pmod{q}$. Furthermore, any unauthorized modification to the request message can be immediately detected by verifying this equation. The probability that an adversary successfully forges a signed message is equivalent to the probability of solving the ring-SIS problem. Therefore, the proposed scheme inherently resists both impersonation and tampering attacks.

4.6. Resistance to additional attacks

Collusion of revoked users: The complete-subtree method is secure against arbitrary collusion of revoked users. Each vehicle stores the node keys $\mathbf{S}_\eta$ along its leaf-to-root path; the cover set $Y = \text{KUNodes}(BT, RL)$ is constructed so that no node $\eta \in Y$ lies on the path of any revoked user. Hence, the union of all revoked users' key material does not contain any updated cover-node key, and colluding revoked users cannot forge valid signatures after the key update.

Malicious DSP: The DSP is semi-trusted. A malicious DSP may refuse service or tamper with the data list, but it does not hold any vehicle private key and thus cannot forge vehicle signatures. Tampering with the broadcast DL is detected because the DSP signs it with its own key $\mathbf{e}_{DSP}$, which every vehicle verifies. Furthermore, the DSP is subject to the supervision of the TA. Any noncompliant or malicious behavior by the DSP will result in it being blacklisted.

Attribute-key leakage: If a vehicle's attribute private key is leaked, the adversary can forge signatures only for that vehicle's policy and only until the vehicle is revoked. The leakage does not affect other vehicles, as each vehicle's key material is generated independently. Prompt revocation and key update immediately contain the damage.

Compromise of the complete-subtree material: Each vehicle stores $O(\log M)$ node keys $\mathbf{S}_\eta$ on its leaf-to-root path. If this material is compromised, the adversary can only impersonate that vehicle (using its $\mathbf{S}_\eta$ to produce a signature) until it is revoked. The compromise does not undermine other vehicles: After revocation, the cover set Y avoids every node on the compromised path, so the updated keys are not derivable from the leaked material.

5. Performance

In this section, we conduct a functional and efficiency comparison of existing schemes.

5.1. Function and security properties comparison

We define the existential unforgeability and the strong unforgeability as EU and SU, respectively. In Table 2, we compare the function and security properties between our proposed scheme and other existing schemes. Several early schemes [19, 20, 30, 31] were constructed based on classical cryptography and cannot resist potential attacks from quantum computers. In terms of security level, the scheme in [20, 30, 31, 34] only achieves EU security. The works in [35–37] are also restricted to EU security. In contrast, both the scheme in [19, 38] and our proposed scheme achieve SU security. In terms of the user revocation function, the schemes in [30, 31, 35] and our proposed scheme support dynamic user revocation. In terms of mutual authentication, partial designs in [31, 34, 36] and our proposed scheme realize bidirectional authentication between vehicles and the authority. As shown in Table 2, our proposed scheme provides comprehensive functional and security guarantees, including stronger security, user revocation, mutual recognition and other features.

Table 2. Function and security properties comparison.

Schemes	Post-quantum	Security level	User revocation	Mutual recognition
[19]	×	SU	×	×
[20]	×	EU	×	×
[30]	×	EU	√	×
[31]	×	EU	√	√
[34]	√	EU	×	√
[35]	√	EU	√	×
[36]	√	EU	×	√
[37]	√	EU	×	×
[38]	√	SU	×	×
Ours	√	SU	√	√

5.2. Space efficiency comparison

In this section, we compare the space efficiency of our proposed scheme with existing lattice-based schemes [34–36]. Table 3 provides a comparison in terms of the TA's public key size, the TA's private key size, the vehicle's private key size, and the signature length. According to Table 3, our proposed scheme achieves superior performance in terms of signature length and other metrics. To provide a fair and concrete comparison, all schemes are instantiated with the same parameters $n = 256, m = 12288, q = 2^{17}, k = 17, p = 7, l = 5, \sigma = 17, N = n \lceil \log q \rceil$. According to Figure 2, the storage space required for our vehicle private key is smaller, making our scheme more suitable for resource-constrained OBUs. The storage size of the private key in our proposal is about 54.17% smaller than that of the schemes in [34, 35]. According to Figure 3, the signature length of our proposed

authentication scheme is shorter than that of the other authentication schemes. In addition, the message signature length of our scheme is 20.72 KB, which is 14.87 KB shorter than those the shortest signature schemes [34, 35]. In fact, our scheme is constructed based on $\mathbb{R}_q$, whereas the other competing schemes [34–36] are designed based on $\mathbb{Z}_q$. Therefore, our scheme achieves higher efficiency with compact keys, short signatures, and reduced computational overhead.

Table 3. Space efficiency comparison.

Schemes	TA's public key size	TA's private key size	Vehicle's private key size	Signature length
[34]	$mn \log q$	$m^2 \log q$	$mk \log q$	$(m + 2n + nk) \log q$
[35]	$mn \log q$	$mn \log q$	$mk \log q$	$(m + 2n + nk) \log q$
[36]	$mn \log q$	$m^2 \log q$	$[l + n(m + N)] \log q$	$[2m + 2(l + 1)N] \log q$
Ours	$(l + 2k)n \log q$	$lkn \log q$	$(l + k)kn \log q$	$(l + 2k)n \log q$

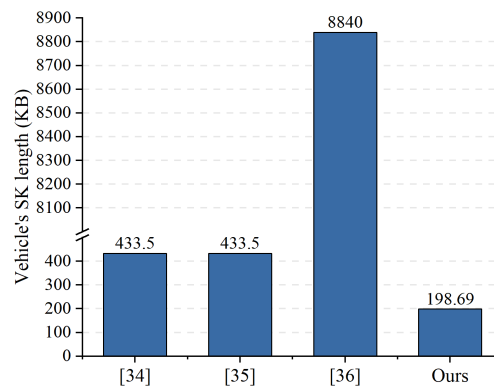


Figure 2. Vehicle's SK length.

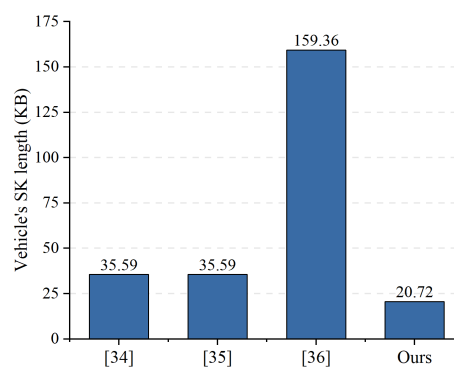


Figure 3. Signature length.

5.3. Revocation performance analysis

We analyze the complete-subtree (CS) revocation mechanism with respect to the five metrics requested. Let M be the number of vehicles (leaf nodes), R the number of revoked users, $d = \lceil \log_2 M \rceil$ the tree height, and $B = k^2 n \lceil \log_2 q \rceil$ the bit-length of one node matrix $\mathbf{S}_\eta \in \mathbb{R}_q^{k \times k}$.

1. Revocation-list size: The revocation list RL stores the indices (or hashes $h = H_1(\mathbf{S}_i)$) of the R revoked users, occupying $R \lceil \log_2 M \rceil = O(R \log M)$ bits.
2. Cover-set size: By the CS method, the cover set $Y = \text{KUNodes}(BT, RL)$ satisfies

$$|Y| \leq R \left\lceil \log_2 \frac{M}{R} \right\rceil \quad (R \geq 1), \quad |Y| = 1 \quad (R = 0);$$

that is $|Y| = O(R \log(M/R))$.

3. Key storage and update communication: Each vehicle stores its leaf matrix $\mathbf{S}_i$ together with the node matrices $\mathbf{S}_\eta$ on its leaf-to-root path, that is, $d + 1$ matrices, so the per-vehicle key size is $(d + 1)B = O(\log M \cdot k^2 n \log q)$ bits. Because the node matrices are predistributed at registration and remain static, revocation requires *no key redistribution*: Each nonrevoked vehicle and the DSP recompute the cover set Y locally from the updated revocation list RL . The only per-revocation communication is therefore the updated list itself, of size $R \lceil \log_2 M \rceil = O(R \log M)$ bits.
4. Revocation computation cost: Computing the cover set requires marking the $d + 1$ nodes on the path of each revoked user and collecting the uncovered siblings, which costs $O(R \log M)$ node operations, independent of n and q .
5. Scalability: The per-vehicle key size grows only logarithmically in M ; the revocation-list and cover-set sizes grow linearly in R (with a factor $\log(M/R)$); because the keys are static, revocation involves no rekeying and its communication is independent of the key size. Hence, the scheme scales gracefully with the number of vehicles and revoked users.

Table 4 summarizes these metrics.

Table 4. Analytical performance of the CS revocation mechanism.

Metric	Formula	Asymptotic
Revocation-list size	$R \lceil \log_2 M \rceil$	$O(R \log M)$
Cover-set size $ Y $	$\leq R \lceil \log_2(M/R) \rceil$	$O(R \log(M/R))$
Key size per vehicle	$(d + 1)B$	$O(\log M \cdot k^2 n \log q)$
Update communication	$R \lceil \log_2 M \rceil$ (list only)	$O(R \log M)$
Revocation computation	$O(R \log M)$ node ops	$O(R \log M)$

5.4. Computation cost comparison

To comprehensively evaluate the computational efficiency of our proposed signature scheme, we conduct both theoretical complexity analysis and practical runtime measurement under a unified experimental environment equipped with an AMD Ryzen 7 5800H processor and 16 GB RAM, where SHA-512 is uniformly adopted for all hash operations. Table 5 compares the theoretical primitive operation costs of signature generation (Signing cost) and verification (Verification cost) between our proposed scheme and the competing schemes proposed in [34–36]. We define $T_{mm}, T_{mv}, T_h, T_{sl}, T_{pe}$,

and T_{sd} as the running time of matrix multiplication, matrix-vector multiplication, hash computation, the SampleL algorithm, the public evaluation algorithm (PubEval), and the SampleD algorithm, respectively. Here, PubEval denotes the public evaluation algorithm of the compared key-policy ABS scheme [36], which publicly evaluates the signing policy over the attribute set. Figure 4 further illustrates the signing cost and verification cost obtained from 50 repeated tests, with the relative confidence interval of all measurements controlled within 5%. As shown in Figure 4, in terms of signing cost, our scheme outperforms the best-performing scheme [34] among the compared works [34–36] with an efficiency improvement of 27.01%. For verification cost, our scheme achieves a 71.02% efficiency gain compared with the optimal baseline scheme [36] in [34–36]. The above results verify the superiority of our signature authentication scheme in computational efficiency. Our scheme can support low-latency computation. This performance perfectly fits the operating demands of resource-constrained edge and mobile terminals.

Remark 3. Hash function instantiation. The codomains of $H_1, \dots, H_5$ are algebraic because their outputs are directly consumed in ring operations: H_3 and H_5 return ring elements that are scaled and summed, H_2 returns a coordinate vector in $\mathbb{R}_q^k$, and H_4 returns a matrix in $\mathbb{R}_q^{k \times k}$. In practice, each H_i is realized by SHA-512: A ring element is first encoded canonically as a binary string (concatenating the $\lceil \log_2 q \rceil$ -bit coefficients of its entries); and hashed together with a distinct domain-separation prefix for each H_i , and the output is expanded in counter mode to $n \lceil \log_2 q \rceil$ bits, which are then parsed into n coefficients of k bits.

Table 5. Computation efficiency comparison.

Schemes	Signing cost	Verification cost
[34]	$T_{mm} + T_{mv} + 3T_h$	$3T_{mv} + 3T_h$
[35]	$T_{mm} + 2T_{mv} + 3T_h$	$3T_{mv} + 3T_h$
[36]	$2T_{sl} + T_{pe}$	$2T_{mv} + T_{pe}$
Ours	$T_{mv} + T_{sd} + (p + k + 1)T_h$	$T_{mv} + 2T_h$

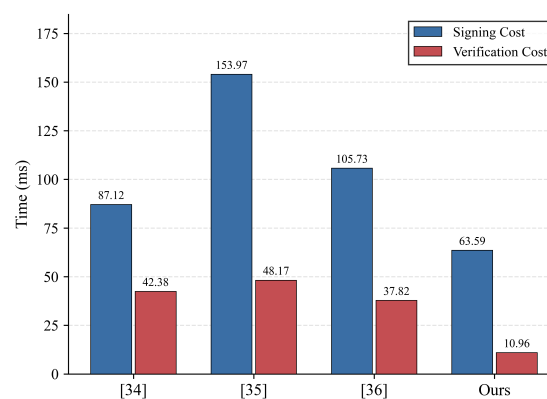


Figure 4. Computational cost of signature.

5.5. Communication overhead and practical considerations

Comparison with realistic V2X message sizes: Standard V2X messages are small: A basic safety message (BSM) is about 100–500 bytes, a cooperative awareness message (CAM) about 100–800 bytes, and a decentralized environmental notification message (DENM) about 300–800 bytes. Our signature is thus roughly one order of magnitude larger than a single BSM/CAM. However, we stress that the signature is generated *once per data-sharing session* for authentication, and is not a periodic beacon; it is not transmitted at the 1–10 Hz rate of BSMs/CAMs.

Bandwidth and transmission time: Over an IEEE 802.11p/DSRC channel at a typical data rate of 6 Mbps, transmitting the 20.72 KB signature takes about

$$20.72 \times 1024 \times 8 / 6 \times 10^6 \approx 28 \text{ ms.}$$

For higher channel rates of 12 Mbps and 27 Mbps, this transmission time falls to roughly 14 ms and 6 ms, respectively.

Packet fragmentation: At the standard 1500-byte MTU, the signature is split into about 15 fragments. For practical deployment, we can tune the Gaussian parameter σ as well as lattice dimensions k and l to reduce signature size.

Latency: The dominant latency component is the transmission time above (tens of milliseconds at low data rates), plus fragmentation/reassembly and MAC contention. This is acceptable for the non-safety-critical data-sharing service, though it would be too large for latency-critical safety beacons.

Channel congestion: Because the signature is transmitted once per session rather than periodically, its contribution to the channel load is bounded and amortized over the session; the subsequent data exchange uses only short symmetric-key ciphertexts.

6. Conclusions

This paper proposed a revocable and efficient attribute-based signature scheme from lattices for secure data sharing in VANETs. The scheme leverages KP-ABS for fine-grained attribute authentication and integrates the complete subtree method to enable efficient dynamic user revocation. Under the random oracle model, the scheme is proven to achieve strong unforgeability based on the hardness of the ring-SIS problem, thereby providing post-quantum security. Comprehensive performance evaluation demonstrates that the proposed scheme outperforms existing lattice-based schemes in terms of both computational efficiency and storage overhead. Specifically, the signing cost is reduced by 27.01% compared to the best-performing competing scheme, and the vehicle private key size is reduced by approximately 54.17%. These advantages make the proposed scheme particularly suitable for deployment in resource-constrained vehicular network environments. Future work will advance in three concrete directions: integrating with National Institute of Standards and Technology (NIST)-standardized post-quantum algorithms to improve interoperability, extending to multiauthority and decentralized ABS architectures for distributed vehicular fog computing, and implementing and optimizing the scheme on resource-constrained hardware. On this basis, we will continue to explore lightweight signature constructions for low-latency VANET demands [44, 45].

Author contributions

Xiaopeng Zhou: Conceptualization, methodology, investigation, validation, writing-original draft, writing-review & editing; Yang Xin: Conceptualization, investigation; Minggang Zheng: Conceptualization, methodology, investigation; Haoyang Dang: Software, supervision; Kaiyan Jin: Investigation, supervision; Lei Shi: Software. All authors have read and approved the final version of the manuscript for publication.

Use of Generative-AI tools declaration

The author declare that they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work was supported by the “20 New Policies for Higher Education Institutions” project of Jinan City [Grant No: 2021GXRC064].

Conflict of interest

The authors declare no conflict of interest.

References

1. M. A. Al-Shareeda, M. Anbar, I. H. Hasbullah, S. Manickam, Survey of authentication and privacy schemes in vehicular ad hoc networks, *IEEE Sens. J.*, **21** (2021), 2422–2433. <http://dx.doi.org/10.1109/JSEN.2020.3021731>
2. H. Shahwani, S. A. Shah, M. Ashraf, M. Akram, J. P. Jeong, J. Shin, A comprehensive survey on data dissemination in Vehicular Ad Hoc Networks, *Veh. Commun.*, **34** (2022), 100420. <http://dx.doi.org/10.1016/j.vehcom.2021.100420>
3. Z. Lu, G. Qu, Z. Liu, A survey on recent advances in vehicular network: security, trust, and privacy, *IEEE T. Intell. Transp.*, **20** (2019), 760–776. <http://dx.doi.org/10.1109/TITS.2018.2818888>
4. S. Dong, H. Su, Y. Xia, F. Zhu, X. Hu, B. Wang, A comprehensive survey on authentication and attack detection schemes that threaten it in vehicular ad-hoc networks, *IEEE T. Intell. Transp.*, **24** (2023), 13573–13602. <http://dx.doi.org/10.1109/TITS.2023.3297527>
5. Y. Zhang, C. Jiang, P. Zhang, Security-aware resource allocation scheme based on drl in cloud-edge-terminal cooperative vehicular network, *IEEE Internet Things*, **11** (2024), 95–104. <http://dx.doi.org/10.1109/JIOT.2023.3293497>
6. C. Li, J. Xie, X. Zhang, *Performance evaluation based on open source cloud platforms for high performance computing*, In: 2013 6th International Conference on Intelligent Networks and Intelligent Systems (ICINIS), IEEE, 2013, 90–94. <http://dx.doi.org/10.1109/ICINIS.2013.30>

7. A. Imghoure, F. Omary, A. El-Yahyaoui, Schnorr-based conditional privacy-preserving authentication scheme with multisignature and batch verification in VANET, *Internet Things*, **23** (2023), 100850. <http://dx.doi.org/10.1016/j.iot.2023.100850>
8. I. Ali, F. Li, An efficient conditional privacy-preserving authentication scheme for vehicle-to-infrastructure communication in VANETs, *Veh. Commun.*, **22** (2020), 100228. <http://dx.doi.org/10.1016/j.vehcom.2019.100228>
9. F. Wang, Y. Xu, H. Zhang, Y. Zhang, L. Zhu, 2plip: A two-factor lightweight privacy-preserving authentication scheme for VANET, *IEEE T. Veh. Technol.*, **65** (2016), 896–911. <http://dx.doi.org/10.1109/TVT.2015.2402166>
10. Y. Kong, M. Jiang, H. Ge, Y. Yan, *An attribute-based signature scheme with flexible access control on lattice*, In: 2024 2nd International Conference on Computer, Vision and Intelligent Technology (ICCVIT), IEEE, 2024, 1–6. <http://dx.doi.org/10.1109/ICCVIT63928.2024.10872549>
11. S. R. Tate, R. Vishwanathan, Expiration and revocation of Keys for attribute-based signatures, *IACR Cryptology ePrint Archive*, **2015** (2015), 400. <http://dx.doi.org/10.1016/j.vehcom.2019.100228>
12. D. Gardham, M. Manulis, Revocable hierarchical attribute-based signatures from lattices, *Appl. Cryptogr. Netw. Secur.*, **13269** (2022), 459–479. http://dx.doi.org/10.1007/978-3-031-09234-3_22
13. L. Wei, J. Cui, Y. Xu, J. Cheng, H. Zhong, Secure and lightweight conditional privacy-preserving authentication for securing traffic emergency messages in VANETs, *IEEE T. Inf. Foren. Sec.*, **16** (2021), 1681–1695. <http://dx.doi.org/10.1109/TIFS.2020.3040876>
14. M. Al Shareeda, M. Anbar, S. Manickam, A. A. Yassin, VPPCS: VANET-based privacy-preserving communication scheme, *IEEE Access*, **8** (2020), 150914–150928. <http://dx.doi.org/10.1109/ACCESS.2020.3017018>
15. H. K. Maji, M. Prabhakaran, M. Rosulek, *Attribute-based signatures*, Topic in Cryptology-CT-RSA 2011, **6558** (2011), 376+.
16. K. Gu, K. Wang, L. Yang, Traceable attribute-based signature, *J. Inf. Secur. Appl.*, **49** (2019), 102400. <http://dx.doi.org/10.1016/j.jisa.2019.102400>
17. G. Lin, Y. Xia, C. Ying, Z. Sun, F2p-abs: A fast and secure attribute-based signature for mobile platforms, *Secur. Commun. Netw.*, **2019** (2019), 5380710. <http://dx.doi.org/10.1155/2019/5380710>
18. J. Li, Y. Chen, J. Han, C. Liu, Y. Zhang, H. Wang, Decentralized attribute-based server-aid signature in the internet of things, *IEEE Internet Things*, **9** (2022), 4573–4583. <http://dx.doi.org/10.1109/JIOT.2021.3104585>
19. B. Chen, T. Xiang, X. Li, M. Zhang, D. He, Efficient attribute-based signature with collusion resistance for internet of vehicles, *IEEE T. Veh. Technol.*, **72** (2023), 7844–7856. <http://dx.doi.org/10.1109/TVT.2023.3240824>
20. D. Gardham, M. Manulis, Hierarchical attribute-based signatures: Short keys and optimal signature length, *Appl. Cryptogr. Netw. Secur.*, 2019, 89–109. http://dx.doi.org/10.1007/978-303021568-2_5
21. H. Su, S. Dong, N. Wang, T. Zhang, An efficient privacy-preserving authentication scheme that mitigates TA dependency in VANETs, *Veh. Commun.*, **45** (2024), 100727. <http://dx.doi.org/10.1016/j.vehcom.2024.100727>

22. Y. Chen, J. Li, Y. Zhang, J. Shen, Attribute-based sanitizable signature with key-exposure resistance for mobile cloud data, *IEEE T. Depend. Secure*, **23** (2026), 2117–2130. <http://dx.doi.org/10.1109/TDSC.2025.3623729>
23. Y. Chen, J. Li, Y. Lu, Y. Zhang, Pairing-free attribute-based signature with message recovery for industrial internet of things, *IEEE T. Depend. Secure*, **22** (2025), 6797–6808. <http://dx.doi.org/10.1109/TDSC.2025.3590954>
24. J. Li, Y. Chen, Y. Lu, J. Ning, Y. Zhang, J. Shen, Efficient Key escrow-free attribute-based signature for anonymous access control in IIoT, *IEEE Internet Things*, **12** (2025), 55567–55580. <http://dx.doi.org/10.1109/JIOT.2025.3623701>
25. Z. Kang, J. Li, J. Shen, J. Han, Y. Zuo, Y. Zhang, TFS-ABS: Traceable and forward-secure attribute-based signature scheme with constant-size, *IEEE T. Knowl. Data En.*, **35** (2023), 9514–9530. <http://dx.doi.org/10.1109/TKDE.2023.3241198>
26. Y. Chen, J. Li, C. Liu, J. Han, Y. Zhang, P. Yi, Efficient attribute based server-aided verification signature, *IEEE T. Serv. Comput.*, **15** (2022), 3224–3232. <http://dx.doi.org/10.1109/TSC.2021.3096420>
27. Z. G. Al-Mekhlafi, Software-defined vehicular networks (SDVN), *Int. J. Comput. Sci. Net.*, **22** (2022), 231–243. <http://dx.doi.org/10.22937/IJCSNS.2022.22.9.33>
28. P. S. K. Oberko, V. K. S. Obeng, H. Xiong, S. Kumari, A survey on attribute-based signatures, *J. Syst. Architect.*, **124** (2022), 102396. <http://dx.doi.org/10.1016/j.sysarc.2022.102396>
29. S. Chawla, N. Gupta, *An analytical view of revocable attribute based encryption schemes*, In: 2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC), IEEE, 2021, 1–8. <http://dx.doi.org/10.1109/ICESC51422.2021.9532873>
30. J. Wei, X. Huang, W. Liu, X. Hu, Practical attribute-based signature: Traceability and revocability, *Comput J.*, **59** (2016), 1714–1734. <http://dx.doi.org/10.1093/comjnl/bxw045>
31. O. Bicer, A. Kupcu, *Versatile ABS: Usage limited, revocable, threshold traceable, authority hiding, decentralized attribute based signatures*, Cryptology ePrint Archive, Paper 2019/203, 2019. Available from: <https://eprint.iacr.org/2019/203>.
32. Z. G. Al-Mekhlafi, A. J. Alrashdan, K. A. Al-Dhlan, H. A. Al-Reshidi, Blockchain-based certificateless authentication with a privacy-preserving scheme in 5G-based vehicular fog computing, *Int. J. Adv. Appl. Sci.*, **13** (2026), 115–124. <http://dx.doi.org/10.21833/ijaas.2026.06.011>
33. P. Shor, *Algorithms for quantum computation: Discrete logarithms and factoring*, In: Proceedings 35th Annual Symposium on Foundations of Computer Science, IEEE, 1994, 124–134. <http://dx.doi.org/10.1109/SFCS.1994.365700>
34. Q. Li, D. He, Z. Yang, Q. Xie, K. R. Choo, Lattice-based conditional privacy-preserving authentication protocol for the vehicular ad hoc network, *IEEE T. Veh. Technol.*, **71** (2022), 4336–4347. <http://dx.doi.org/10.1109/TVT.2022.3147875>
35. A. A. Almazroi, M. A. Alqarni, M. A. Al-Shareeda, S. Manickam, L-cppa: Lattice-based conditional privacy-preserving authentication scheme for fog computing with 5G-enabled vehicular system, *PLOS ONE*, **18** (2023), 1–23. <http://dx.doi.org/10.1371/journal.pone.0292690>

36. L. Liu, C. Hsu, M. H. Au, L. Harn, J. Cui, Z. Zhao, A revocable and comparable attribute-based signature scheme from lattices for IoMT, *J. Syst. Architect.*, **154** (2024), 103222. <http://dx.doi.org/10.1016/j.sysarc.2024.103222>
37. F. Luo, S. Al-Kuwari, W. Susilo, D. H. Duong, Attribute-based proxy re-signature from standard lattices and its applications, *Comput. Stand. Inter.*, **75** (2021), 103499. <http://dx.doi.org/10.1016/j.csi.2020.103499>
38. S. Mukherjee, D. S. Gupta, G. P. Biswas, An efficient and batch verifiable conditional privacy-preserving authentication scheme for VANETs using lattice, *Computing*, **101** (2019), 1763–1788. <http://dx.doi.org/10.1007/s00607-018-0689-3>
39. C. Gentry, C. Peikert, V. Vaikuntanathan, *Trapdoors for hard lattices and new cryptographic constructions*, Cryptology ePrint Archive, Paper 2007/432, 2007. Available from: <https://eprint.iacr.org/2007/432>.
40. D. Micciancio, C. Peikert, *Trapdoors for lattices: Simpler, tighter, faster, smaller*, Advances in Cryptology–EUROCRYPT 2012, 2012, 700–718. http://dx.doi.org/10.1007/978-3-642-29011-4_41
41. L. Ducas, D. Micciancio, *Improved short lattice signatures in the standard model*, Advances in Cryptology–CRYPTO 2014, 2014, 335–352. http://dx.doi.org/10.1007/978-3-662-44371-2_19
42. M. Ajtai, Generating hard instances of the short basis problem, *Automata Lang. Program.*, (1999), 1–9. http://dx.doi.org/10.1007/3-540-48523-6_1
43. A. El Kaafarani, S. Katsumata, Attribute-based signatures for unbounded circuits in the ROM and efficient instantiations from lattices, *IACR Cryptology ePrint Archive*, **2018** (2018), 22. <http://eprint.iacr.org/2018/022>
44. H. Su, S. Dong, T. Zhang, A hybrid blockchain-based privacy-preserving authentication scheme for vehicular ad hoc networks, *IEEE T. Veh. Technol.*, **73** (2024), 17059–17072. <http://dx.doi.org/10.1109/TVT.2024.3424786>
45. P. Thanalakshmi, N. Anbazhagan, G. P. Joshi, E. Yang, A quantum resistant universal designated verifier signature proof, *AIMS Math.*, **8** (2023), 18234–18250. <http://dx.doi.org/10.3934/math.2023927>



AIMS Press

© 2026 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<https://creativecommons.org/licenses/by/4.0>)