Journal of Systems Architecture 160 (2025) 103345

Contents lists available at ScienceDirect

Journal of Systems Architecture

journal homepage: www.elsevier.com/locate/sysarc

A hash-based post-quantum ring signature scheme for the Internet of Vehicles

Shuanggen Liu [a,*], Xiayi Zhou [a], Xu An Wang [b], Zixuan Yan [a], He Yan [a], Yurui Cao [a]

[a] School of Cyberspace Security, Xi'an University of Posts and Telecommunications, Xi'an, Shaanxi, China
[b] Key Laboratory of Network and Information Security, Engineering University of People's Armed Police, Shaanxi, China

ARTICLE INFO

Keywords:
Ring signature
Internet of Vehicles
Merkle tree
Post-quantum digital signature
Hash-based signature scheme

ABSTRACT

With the rapid development of the Internet of Vehicles, securing data transmission has become crucial, especially given the threat posed by quantum computing to traditional digital signatures. This paper presents a hash-based post-quantum ring signature scheme built upon the XMSS hash-based signature framework, leveraging Merkle trees for efficient data organization and verification. In addition, the scheme is applied to the Internet of Vehicles, ensuring both anonymity and traceability while providing robust quantum-resistant security. Evaluation results indicate that, compared to other schemes, the proposed method achieves superior verification speed while ensuring data security and privacy.

1. Introduction

As a fundamental necessity in modern life, the number of vehicles produced worldwide continues to grow. According to relevant statistics, global vehicle production reached 94 million units in 2023 [1]. Additionally, data from the International Organization of Motor Vehicle Manufacturers indicates that there are now 1.3 billion vehicles in use [2]. However, this growth brings various challenges, including network attacks, unauthorized access, and concerns around road safety and privacy. To address these issues, new research fields, such as intelligent transportation systems (ITS) and the Internet of Vehicles (IoV), have emerged. These fields aim to provide safer, more efficient, and more harmonious vehicular environments. Vehicle-to-Everything (V2X) technology enables the effective use of dynamic information from all networked vehicles via on-board devices, facilitating secure, efficient, intelligent, and comfortable services, thereby contributing to the intelligence of social traffic systems [3]. The typical VANET structure is shown in Fig. 1.

With the increasing number of vehicles and the development of the IoV, it is a very important job to ensure the security of the IoV systems. Currently, the security of vehicular networks, whether internal or external, primarily relies on digital signatures or public-key encryption. However, as quantum computing advances, traditional digital signature algorithms are increasingly vulnerable to quantum attacks, making it essential to incorporate post-quantum digital signature algorithms into IoV research. Unlike traditional computers, quantum computers can accelerate the cracking of probabilistic algorithms through parallel computation capabilities [4]. In light of these challenges, post-quantum cryptography has become a critical area of study, with the aim of establishing a resilient foundation for the industry. The National Institute of Standards and Technology (NIST) has been conducting a multi-stage standardization process for post-quantum cryptography. The third round of candidate evaluations has been completed, and algorithms such as SPHINCS+, CRYSTALS-DILITHIUM, and CRYSTALS-KYBER have been standardized. These algorithms achieve varying levels of bit-level security depending on key size and parameter settings, which align with NIST security levels from 1 to 5, representing 128/160/192/224/256-bit security strengths, respectively [5]. A post-quantum digital signature scheme is a digital signature scheme capable of resisting quantum attacks. Among post-quantum digital signature schemes, hash-based schemes are particularly effective and provably secure. Hash-based post-quantum digital signature schemes offer significant advantages over other types of post-quantum schemes due to their high computational efficiency, scalability, maturity, and reliance solely on the preimage resistance of the underlying hash function [6].

In IoV networks, where both privacy and traffic safety are essential, ring signatures are especially suitable. Ring signature schemes offer anonymity by concealing the identity of signer among a group of participants. Using hash-based post-quantum ring signatures, vehicles can sign messages anonymously within a group, ensuring their identities cannot be traced. These signatures also provide unforgeability, collision resistance, resilience against quantum attacks, and low communication overhead. In densely populated cities, managing keys for secure vehicular communications can be challenging, especially given the limited IoV coverage [7]. The Merkle tree structure effectively compresses keys, reducing key management costs [8]. In this study, we propose a

* Corresponding author.
E-mail address: liushuanggen201@xupt.edu.cn (S. Liu).

https://doi.org/10.1016/j.sysarc.2025.103345
Received 11 November 2024; Received in revised form 23 December 2024; Accepted 16 January 2025
Available online 23 January 2025
1383-7621/© 2025 Elsevier B.V. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

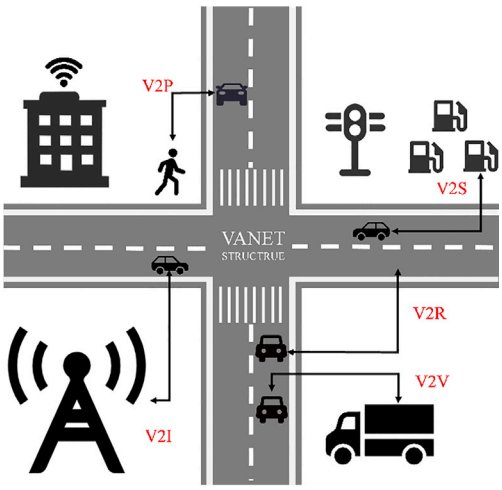

Fig. 1. VANET structure.

hash-based post-quantum ring signature scheme for IoV applications. The ring signature algorithm of Our scheme is based on the **XMSS** algorithm, aiming to enhance data sharing security and efficiency. Merkle trees are used to organize and verify data efficiently, while ring signatures ensure the authenticity and integrity of data within the IoV network without compromising user anonymity.

1.1. Related works

In recent years, hash-based post-quantum digital signature schemes have garnered significant attention within the cryptography community. Following the fourth round of the NIST post-quantum digital signature standardization process, the SPHINCS+ algorithm was introduced as a supplementary standard, featuring a flexible, tunable hash function structure [9]. As the standardization process progresses, researchers have proposed various adaptations, including SPHINCS-a and SPHINCS+-c, which further compress signature sizes and enhance execution speeds [10,11]. Additionally, Sun, Liu, and colleagues developed a domestic signature algorithm based on the post-quantum hash function SM3 [12]. Hülsing and Kudinov provided a rigorous security proof for the SPHINCS+ algorithm, confirming its robustness in a post-quantum environment [13]. The **XMSS** algorithm forms the foundation of SPHINCS+, with its architectural design and security proof presented by Hülsing, Butin, and others [14]. Research on hardware implementations of the **XMSS** algorithm has also advanced, with significant contributions from Thoma and Güneysu [15]. Meanwhile, Sun and Liu investigated the feasibility of replacing the hash function in **XMSS** with the domestic SM3 hash function [16]. An essential component of **XMSS** is WOTS+, a one-time signature algorithm; Hülsing provided its security proof [17], while Zhang, Cui, and colleagues evaluated the efficiency of WOTS+ in tree-based one-time signature algorithms [18]. Currently, research on post-quantum digital signatures primarily concentrates on enhancing signature efficiency and replacing the underlying hash functions. However, there is a scarcity of studies that integrate post-quantum digital signatures with specific application scenarios or explore their variants.

The exploration of post-quantum ring signatures is also accelerating in post-quantum digital signature research. Xie, Wang, and colleagues highlighted that traditional signature algorithms are highly susceptible to quantum computing attacks, and noted that ring signatures offer considerable advantages in blockchain applications, including medical data sharing and vehicular networking, due to their unique properties [19]. Chatterjee and Chung et al. conducted an in-depth analysis on the security of post-quantum ring signature, re-examined the security of classical signature and ring signature in the quantum environment, and proposed two short signature schemes, which were implemented in the quantum random prediction model and the ordinary model respectively [20]. Recent literature has introduced novel architectures, such as linkable ring signatures, threshold ring signatures, and identity-based post-quantum ring signatures, discussing their post-quantum security features [21–23], Similarly, literature [24]systematically reviews the theory and application of linkable ring signatures, providing an in-depth comparison of anonymization and linkability schemes, but these studies lack analysis of specific application scenarios (such as the IoV), and do not fully consider resource-constrained environments and the potential of anti-quantum computing.

In response to the research of NIST on post-quantum algorithms and verification ring signatures, a blockchain-based, post-quantum anonymous, traceable, and verifiable authentication scheme was proposed to mitigate quantum attacks while addressing security and privacy concerns, with an evaluation of its feasibility in IoV environments [25]. The IoV faces significant security and privacy challenges, and blockchain technology offers an effective platform to ensure both user privacy and security [26–28]. Literature [29] proposes an identity authentication and signature scheme for UAV-assisted Vehicular Ad Hoc Networks (VANET), focusing on enhancing network anonymity and user privacy through an efficient authentication mechanism. Literature [30] introduces a distributed message authentication scheme combined with a reputation mechanism to improve the security and trust of the IoV. The scheme uses node credit values to authenticate message validity, effectively preventing malicious attacks and forgery. Literature [31] presents an authentication key negotiation protocol for intelligent transportation systems in vehicle networks, strengthening identity authentication and key exchange mechanisms to prevent security threats such as eavesdropping, tampering, and man-in-the-middle attacks. While these studies address key security challenges in vehicular networks, they often focus on specific aspects, lacking comprehensive and scalable frameworks for real-world scenarios. Furthermore, the integration of post-quantum cryptography and scalability in dynamic, large-scale networks remains underexplored, highlighting opportunities for future research into robust and future-proof solutions. Given the inherent advantages of ring signatures, they are particularly well-suited for applications such as the Internet of Vehicles, making further investigation essential.

In order to ensure the post-quantum security of data transmission in the IoV environment, researchers have proposed various solutions. The literature [32] recommends the use of lattice-based post-quantum digital signature, but the signature algorithm has not been combined with specific scenarios. Another study [33] proposed a ring-signature scheme based on lattice-based difficult problems and combined it with the vehicle-connected environment, but the quantum anti-attack characteristics of the scheme were not explained in detail. In addition, reducing energy consumption in blockchain has also become a research focus [34]. An energy saving method is adopted to calculate the root of Merkle tree, and a Merkle tree design scheme conforming to the specification is proposed. The effectiveness of this method is verified through experiments. At the same time, the Merkle tree accumulator algorithm proposed by Derler and Ramacher in [35] builds an accumulator that can resist quantum attacks by using only hash function and symmetric meta language, and gives specific operations and definitions. However, the specific algorithm implementation and its combination in practical application scenarios need to be further studied.

1.2. Contributions

Firstly, building on the Merkle tree accumulator algorithm described in Ref. [35], we propose a hash-based ring signature algorithm specifically designed for IOV, we improve the Merkle tree accumulator algorithm to XMSS accumulator algorithm. This algorithm integrates the principles of ring signatures with Merkle tree structures. Unlike

Table 1
Notation for ring signature scheme.

λ	Security parameter
N	The size of the ring
(pk, sk)	Key pair
R	A ring consisting of $(pk_1, pk_2, \ldots\ldots, pk_l)$
m	The message digest
σ	The signature of message

traditional ring signature algorithms, this proposed scheme can resist quantum attacks, thus offering post-quantum security.

Secondly, we construct a new hash-based post-quantum ring signature scheme for application of vehicular network. This scheme enhances the security of data transmission within the vehicular network, providing robust post-quantum security to effectively protect shared data.

1.3. Structure

The remainder of this paper is organized as follows: Chapter 2 provides the necessary foundational knowledge, along with a review of the background and related work relevant to this study. In Chapter 3, we present a post-quantum ring signature algorithm based on Merkle trees and discuss its application within the IoV environment. Chapter 4 offers a security analysis and proof of the robustness of proposed. In Chapter 5, we evaluate the performance of the scheme and compare it with existing alternatives. Finally, Chapter 6 concludes the paper and outlines directions for future research.

2. Preliminaries

2.1. Ring signature

Ring signature is a digital signature scheme introduced by Rivest, Shamir, and Tauman in 2001. A ring is composed of a group of members, allowing any member within the group to sign on behalf of the entire group without revealing the identity of the signing member [36],The main parameters of ring signature are given in Table 1.

Definition 1 (*Ring Signature*)**.** A ring signature scheme consists of three core algorithms: key generation, signature generation, and signature verification. These algorithms are defined as follows:

Step1: Key generation
$(pk, sk) \leftarrow Gen(\lambda, N)$:The size of the ring is N, set the security parameters λ the maximum number of members in the ring N, λ and N as input, the output is the public and private key pair.
Step2: Signature generation
$\sigma \leftarrow Sign(sk, R, m)$: Input private key sk, set of all public keys $R = (PK_1, PK_2, \ldots, PK_L)$, message $m \in M_\lambda$, output signature σ.
Step3: Signature verification
$True/false \leftarrow Ver(R, m, \sigma)$: Input a collection composed of all public keys R, message $m \in M_\lambda$, signature σ, and output $True/false$.

A ring signature must satisfy two critical security properties: anonymity and Unforgeability. Anonymity ensures that while the signature indicates it was generated by a member of the ring, it does not reveal the specific identity of the signer. Unforgeability guarantees that only members of the ring can generate valid signatures; outsiders cannot create valid signatures for the ring.

Definition 2 (*Unforgeability*)**.** Unforgeability ensures that only members of the ring can generate a valid signature. In the unforgeability model, we assume that the attacker has access to a public key and aims to produce a valid ring signature without authorization.

Let the security parameter λ, ring signature $RS = (Gen, sig, Ver)$, algorithm A is polynomial-time algorithm (any PPT adversary A), for any integer s, define the following experiment:

Step 1, the challenger generates s key pairs (pk, sk) in which $i \in [1, s]$, and sends all the public keys PK_i in a set $PK = (PK_1, PK_2, \ldots, PK_s)$ to A.

Step 2, the challenger chooses one PK_i and checks whether PK_i belongs to R, if $Sig(sk_i, R, m) \rightarrow \sigma$ is calculated by the challenger, then the challenger will send σ to A.

Step 3, the attacker outputs the tuple R^*, m^*, σ^*, and the challenger checks it.

If: $R^* \in PK$ Attacker A never performs signature query access to $(sign, R^*, m^*)$,
$Ver(R^*, m^*, \sigma^*)$
And returns a 1 for the experiment, or a 0 otherwise.

$$Adv_{UNF}^{\lambda,s}(A) = Pr[Exp_{UNF}^{\lambda,s}(A) = 1] \leq nelg(\lambda)$$

Definition 3 (*Anonymity*)**.** Anonymity in a ring signature scheme ensures that the identity of signer remains concealed among a group of potential signers, making it impossible to determine who specifically generated the signature. This anonymity is achieved through a ring signature generation process that relies on the public keys of all group members, without revealing the identity of the actual signer.

In the anonymization experiment, the adversary is given a ring signature generated from any two pairs of public and private key pairs, as well as from either of these two private keys, which contains both public keys owned by the adversary, and the goal of adversary is to distinguish which private key was used to generate the ring signature with negligible probability.

Let the security parameter λ, the ring signature $RS = (Gen, sig, Ver)$, algorithm A be a polynomial time algorithm, for any integer s and any bit b, define the experiment as follows:

Step 1, the challenger generates s key pairs (PK_i, SK_i), of which $i \in [1, s]$, and sends all the public keys PK_i to A.

Step 2, A sends (R, m, i_0, i_1) to the challenger, the challenger checks if $pk_{i_0} \in R_2$, $pk_{i_1} \in R_2$, then the challenger calculates $R_2\sigma \leftarrow Sig(sk_{i_b}, R, m)$ and send σ to A.

Step 3, A returns a guess bit b^* where the experiment $b^* = b$ outputs 1 if and 0 otherwise, and RS is considered anonymous if for all s and all polynomial-time algorithms A, the probability of A returning 1 in the $(s, 0)$-anonymous experiment (in the λ) is ignorably close to the probability of A returning 1 in the $(s, 1)$anonymous experiment.

$$Adv_{ANON}^{\lambda,s}(A) = |Pr[Exp_{ANON}^{\lambda,s}(A)] - \frac{1}{2}| \leq nelg(\lambda)$$

2.2. WOTS+

Ralph Merkle pioneered hash-based signature algorithms, as noted in Ref. [37]. Currently, hash-based signature schemes are categorized into three main types: one-time signature schemes (OTS), few-time signature schemes (FTS), and many-time signature schemes (MTS).

The Table 2 below summarizes some of the most widely used hash-based signature schemes. Research on OTS schemes began with the Lamport-Diffie algorithm. This paper adopts the WOTS+ (Winternitz One-Time Signature Plus) scheme, which comprises three main components: key generation (GEN), signature generation (SIG), and signature verification (VER).

The first step is parameter selection, where parameter ω, an integer $\omega \in N$ with $\omega \geq 2$, is determined to set the number of hash iterations required to construct the $n \in N$ public key. Additionally, the hash output length m and security parameter n, where, need to be defined. Next, parameters l_1 and l_2 are computed, which are then summed to obtain l. The calculation method is as follows:

$$l_1 = \left\lceil \frac{m}{\log_2 \omega} \right\rceil, l_2 = \left\lfloor \frac{\log_2 (l_1(\omega - 1)) + \log_2 \omega}{\log_2 \omega} \right\rfloor, l = l_1 + l_2$$

Table 2
Classification table for hash-based signature schemes.

Scheme Type	Scheme Name
OTS	Lamport-Diffe, WOTS, $WOTS^{+}$
FTS	HORS, HORST-T, PORS, PORS-T
MTS	XMSS, SPHINCS, SPHINCS+

Table 3
Parameter descriptions for the WOTS+ algorithm.

$n \in N$	Security parameter
$w \in N$	Winternitz parameter ($w \geq 2$)
$m \in N$	Bit length of the message digest
F_n	A set of functions, $F_n = \{f_k \mid k \in \{0,1\}^n\}$, $f_k : \{0,1\}^n \to \{0,1\}^n$
$h \in N$	Height of the tree
H	Hash function, $H : \{0,1\}^* \to \{0,1\}^m$
$x \in \{0,1\}^n$	Randomly chosen string x, used to construct a one-time verification key

The Table 3 gives the meaning of the parameters in the formula. Next define the operation, WOTS+ uses the function F_n family:

$$F_n : \{0,1\}^n \to \{0,1\}^n$$

Define the function operation:

$$\begin{cases} c^i(x,r) = F(c_k^{i-1}(x,r) \oplus r_i) & i > 0 \\ c^i(x,r) = x, i & i = 0 \end{cases}$$

$$\begin{cases} x \in \{0,1\}^n \\ F = F_n : \{0,1\}^n \to \{0,1\}^n \\ r = (r_1, r_2, \ldots\ldots, r_{2^w-1}) \qquad r \in \{0,1\}^{n\times(2^{\omega}-1)} \end{cases}$$

Step1: Key Generation(GEN)

The process of key generation mainly includes two steps: private key generation and public key generation. The key generation process is shown in Fig. 2.

(1) Private key generation: Using PRG to generate $l + 2^{\omega} - 1$ n bits of random number, the first random number is the private key $sk = (sk_0, sk_1, \ldots\ldots, sk_{l-1})$, and the last $2^{\omega} - 1$ are the mask, $r = (r_1, r_2, \ldots\ldots, r_{2^{\omega}-1})$.

(2) Public key generation: The public key consists of $l + 1$ blocks, the first block is the mask r, the last L blocks are converted by sk, and The public key is composed as follows:

$$pk_i = c^{2^{\omega}-1}(sk_{i-1}, r), \quad i \in [1, l]$$

$$pk = (pk_0, pk_1, \ldots, pk_l) = \left(r, c^{2^{\omega}-1}(sk_0, r), \ldots, c^{2^{\omega}-1}(sk_{l-1}, r)\right)$$

Step2: Message Signature(SIG)

(1) Generate message digest: Generate message digest M that needs to be signed message m through the hash function, and then divide the message digest into l_1 parts, each ω bit, where each ω bit represents the m_i, $i \in [0, l_1 - 1]$ equivalent of an integer. The message digest generation process is shown in Fig. 3, and the overall signature generation process is shown in Fig. 4.

(2) Calculate the checksum:

$$C = \sum_{i=1}^{l_1} (2^{\omega} - 1 - m_i) \leq l_1(2^{\omega} - 1)$$

Divide C into ω bits, and $c = (c_0, c_1, \ldots\ldots, c_{l_2-1})$.

Let $b = (b_0, b_1, \ldots\ldots, b_{l-1})$, that is b be the concatenation of m and c. Signature generation is represented by the following formula:

$$\sigma = (\sigma_0, \sigma_1, \ldots, \sigma_{l-1}) = \left(F^{b_0}(sk_0, r), F^{b_1}(sk_1, r), \ldots, F^{b_{l-1}}(sk_{l-1}, r)\right)$$

Step3: Message verification(VER)

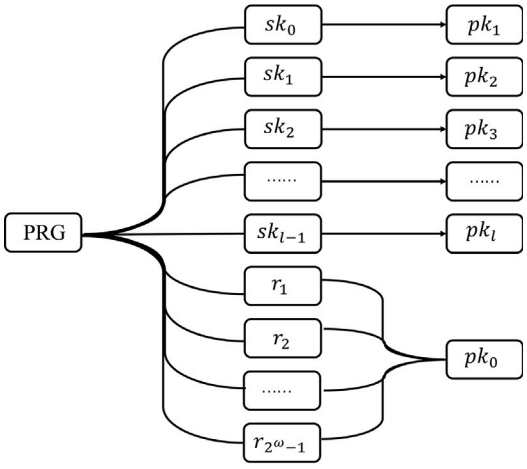

Fig. 2. Key generation process for WOTS+.

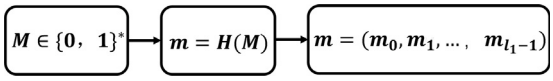

Fig. 3. Message digest generation graph.

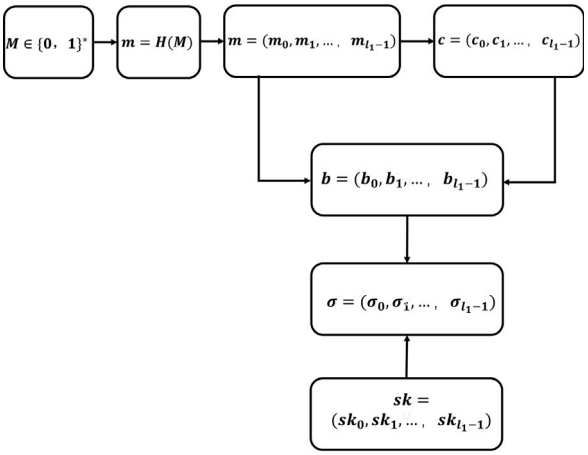

Fig. 4. WOTS+ signature generation diagram.

The message M is converted to $b = (b_0, b_1, \ldots\ldots, b_{l-1})$. Then, the transmitted signature $\sigma = (\sigma_0, \sigma_1, \ldots\ldots, \sigma_{l-1})$ is processed as follows to obtain pk'. If the signature is the same as pk, the signature verification succeeds.

$$pk' = (r, pk'_1, pk'_2, \ldots, pk'_l) = \left(r, F^{2^{\omega}-1-b_0}(\sigma_0), F^{2^{\omega}-1-b_1}(\sigma_1), \ldots, F^{2^{\omega}-1-b_{l-1}}(\sigma_{l-1})\right)$$

2.3. XMSS

2.3.1. Merkle tree

The Merkle Signature Scheme (MSS), proposed by Ralph Merkle in 1979, integrates the Merkle Tree with an OTS algorithm. A Merkle tree is a hierarchical structure where leaf nodes contain hash values of data, and non-leaf nodes store the combined hash values of their child nodes. This structure enables efficient data integrity verification, especially for large-scale datasets. The structure of the Merkle tree is shown in Fig. 5.

According to the Fig. 5, the tree has 3 layers and $2^3 = 8$ leaf nodes, each storing the hash of a one-time signature public key. The leaf nodes,

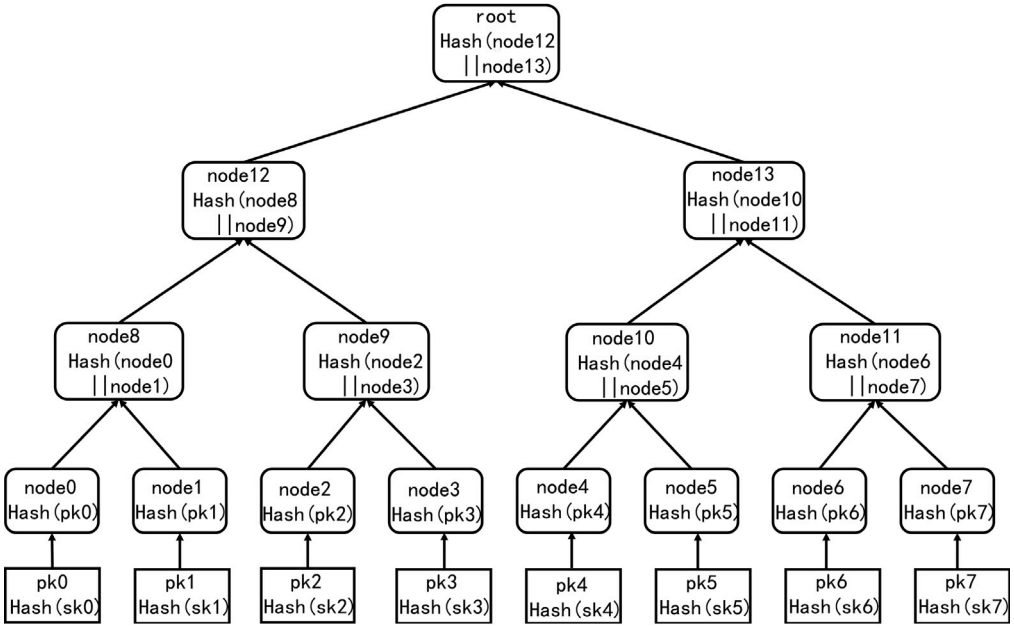

Fig. 5. Merkle tree structure diagram.

labeled node0 to node7, are hashed pairwise to generate the middle nodes. The final root node stores the public key.

The Merkle tree serves two primary functions:

(1) Data Integrity Verification, where users can check if data has been tampered with by recalculating the root hash.

(2) Public Key Size Compression, reducing the storage requirements for numerous public keys by consolidating them into a single root key.

2.3.2. Key generation

The XMSS algorithm deploys 2^h WOTS+ instances as the 2^h leaf nodes of a Merkle tree with height h, with the root node authenticating these instances [38]. The XMSS key consists of multiple OTS keys and the root of the Merkle tree as the public key.

Step1: Select the parameters

Step2: Generate a one-time signature key pair (pk, sk)

Step3: Build the Merkle tree

Use each OTS public key pk_i as a leaf node of the Merkle tree. Each leaf node generates non-leaf nodes through a hash function, which eventually generates the Root node. The parent node in the Merkle tree is generated from the hash of the two child nodes, that is, $Node(i) = H(child(1) \parallel child(i))$, the root node $Root$ serves as the XMSS public key.

Step4: Output the key pair

Public key: $pk = (root, seed)$, the private key consists of the OTS key pairs.

2.3.3. Message signature

To sign a message, an unused WOTS+ private key is selected, and the Merkle tree path proof is generated to output the signature SIG.

Step1: Select WOTS+ key

Choose an unused WOTS+ private key sk_i, ensuring it is used only once.

Step2: Generate WOTS+ one-time signature

Use the WOTS+ private key to sign message M, producing the OTS signature Sig_{OTS}.

Step3: Merkle tree path proof

Hash path from leaf node pk_i to Root node, this path proves that OTS public key is valid.

Step4: Generate XMSS signature

The signature includes: serial number i (using the i th OTS key), OTS signature Sig_{OTS}, and AuthPath for authentication of the Merkle tree $Sig_{XMSS} = (i, Sig_{OTS}, AuthPath)$.

2.3.4. Signature verification

The signature verification process ensures the correctness of the OTS signature and validates that the corresponding OTS public key is consistent with the root of the Merkle tree. The main steps are as follows:

Step1: Extract Information

Extract OTS serial number i, OTS signature Sig_{OTS}, and path proof AuthPath for the Merkle tree from XMSS signature Sig_{XMSS}.

Step2: Verify OTS signature

Using the extracted OTS public key, verify the validity of Sig_{OTS} for the message M. If verification fails, the signature is deemed invalid.

Step3: Compute Merkle Tree Path

Calculate the Merkle tree node of the OTS public key Using OTS public key pk_i and path proof AuthPath, calculate the hash value of the parent node step by step from the leaf node pk_i until the root node $Node(i) = H(child(i) \parallel child(i))$ is calculated.

Step4: Compare Root Nodes

Compare the reconstructed root node with the root node Root from the XMSS public key. If the values match, the signature is valid; otherwise, it is invalid.

3. Hash-based post-quantum ring signature scheme

In addition to its high computational efficiency and excellent scalability, the hash function-based signature scheme exhibits greater algorithmic maturity compared to other post-quantum digital signature schemes, such as XMSS and SPHINCS+. Furthermore, post-quantum ring signatures ensure both the anonymity and unforgeability of signatures. Consequently, in light of the security threats posed by the rapid advancement of quantum computing, it is highly significant to integrate the post-quantum ring signature scheme with vehicle networking.

3.1. Design principles

The Merkle tree is an efficient data structure, a binary hash tree where each node represents the hash value of a data block. The root node represents the hash of the entire data set. The characteristics of the Merkle tree make it a highly efficient method for storing and verifying large amounts of data. In blockchain, Merkle trees are widely used to store transaction data and block hashes. Ring signatures enable

Table 4
Meaning of parameters in the proposed scheme.

Parameter	Description
k	Security parameter
t	Maximum number of elements to accumulate
i	$i \in [0, 2^h - 1]$
$h \in N$	Height of the tree
H	Hash function, $H : \{0,1\}^* \to \{0,1\}^m$
(sk_Ω, pk_Ω)	A key pair
X	The set of $\{x_i \mid i \in [0, 2^h - 1]\}$
Ω	The accumulator
aux	The auxiliary information
wit_{x_i}	The certificate for x_i

a message sender to demonstrate possession of at least one public key within a set while concealing the specific public key used, thus providing anonymity and unlinkability. This feature makes ring signatures particularly valuable in applications centered on privacy and secure communication. Within ring signatures, Merkle trees can be employed to organize the hashes of messages or data blocks into a tree structure, facilitating efficient verification of data integrity and authenticity. Furthermore, ring signatures can leverage Merkle trees to obscure the identity of sender by integrating the public key of signer with those of other members in a ring. Consequently, the signer can validate ownership of at least one public key in the set without disclosing the specific key used. Even if an attacker intercepts the signed message, they would be unable to ascertain the true identity of the signer.

3.2. Scheme description

This scheme is based on the definition of Merkle tree accumulators as described in [35], with slight modifications to accommodate the proposed post-quantum ring signature scheme utilizing hash functions, specifically designed for vehicular networks. This formalism facilitates the restatement of the Merkle tree accumulator algorithm within the current framework. The main parameters of this scheme are given in Table 4.

Definition 4 (*Extend Merkle Tree Accumulator*)**.** The Merkle tree accumulator algorithm (Algorithm 1) comprises the following subroutines (Gen, Eval, WitCreate, Verify), defined as follows:

$Gen(1^k, t)$: The key generation algorithm takes a security parameter k and a parameter t, where t is the upper bound on the number of elements to be accumulated, and returns a key pair (sk_Ω, pk_Ω).

$Eval((sk_\Omega, pk_\Omega), X)$: This algorithm takes the key pair (sk_Ω, pk_Ω) and the set of elements X to be accumulated, returning the accumulator Ω_X and some auxiliary information aux.

$WitCreat((sk_\Omega, pk_\Omega), \Omega_X, aux, x_i)$: This algorithm takes the key pair(sk_Ω, pk_Ω), accumulator Ω_X, auxiliary information aux, and an element x_i. If x_i is not in the set X, it returns false; otherwise, it returns a certificatewit_{x_i} for x_i.

$Verify(pk_\Omega, \Omega_X, wit_{x_i}, x_i)$: This algorithm takes the public key pk_Ω, accumulator $\Omega_X$certificate wit_{x_i}, and element x_i. If wit_{x_i} is a valid certificate for x_i it returns 1; otherwise, it returns 0.

The Merkle tree accumulator ensures both correctness and collision resistance. Collision resistance indicates the difficulty of finding an element $x_{i,j}$ that does not belong to X yet possesses a valid certificate $x_{i,j}$.

Definition 5 (*Collision Resistance*)**.** Collision resistance implies that for an adversary A possessing a valid key pair (sk_Ω, pk_Ω) generated by the Gen algorithm, and under the assumption that intermediate values are correct, the probability of finding an element x_i^* that is not in the accumulator X^* but still produces a verification result of 1 is negligible. Assuming the existence of a negligible function $\varepsilon(k)$, collision resistance is formally defined as follows:

$$Pr\left[\begin{array}{l} Eval_{r_i}((sk_\Omega, pk_\Omega), X^*) \to \Omega^* \\ (\mathrm{Gen}(1^k, t) \to (sk_\Omega, pk_\Omega))(A(pk_\Omega) \to (wit^*_{x_i}, x_i^*, X^*)) \\ Verify(pk^\Omega, \Omega^*, wit^*_{x_i}, x_i^*) = 1 \wedge x_i \in X^* \end{array}\right] \le \varepsilon(k)$$

The implementation of the Merkle tree ring signature is described next, and the whole process is covered in Algorithm 1.

Step1: Key Generation: $Gen(1^k, t)$

First, determine the hash functions $\{H_k\}_{k \in K^K}$, where for any $k \in K^K$, the hash function $H_k : \{0,1\}^* \to \{0,1\}^K$. The hash function can be chosen as SHA functions, SM2, SM3, etc. Determine the parameter N, which represents the number of ring members, and t, the upper bound for accumulating elements. Then, generate the key pairs and return (sk_Ω, pk_Ω).

Step2: Public Key Evaluation Eval: $Eval((sk_\Omega, pk_\Omega), X)$

Parse the number of ring members N. The parsing rule is that if N is not a power of 2, the function returns false, as it must be a perfect binary tree. If N is a power of 2, begin computation from layer 0 (the leaf nodes at the lowest level) and continue until the root (the single node at the top) is obtained. Let $L_{u,v}$ represent the node at layer v and the u-th leaf index. The auxiliary variable aux stores the hash values corresponding to each layer.

Step3: Certificate Creation: $Wit((sk_\Omega, pk_\Omega), \Omega_X, aux_{x_i}, x_i)$

First, parse aux into nodes at each level of the Merkle tree. Then, reconstruct the Merkle tree from bottom to top. The $WitCreat$ algorithm involves using intermediate nodes to build up to the root hash value.

Step4: Certificate Verification: $Verify(pk_\Omega, \Omega_X, wit_{x_i}, x_i)$

The final step is verification. Start by setting the leaves to the hash values of each party and proceed to compute hashes from the bottom up. Check if the final result matches the root node value. If it matches, it verifies that the member is part of the ring. For example, node $l_{0,2}$ is visualized in Fig. 6, showing how node $l_{0,2}$ reconstructs the root node in a Merkle tree with height $h = 3$ and $N = 8$ leaf nodes.

Algorithm 1 Extend Merkle tree accumulator

input: k, t, $\{H_k\}_{k \in K^\kappa}$, $H_k : \{0,1\}^* \to \{0,1\}^\kappa$
output: (sk_Ω, pk_Ω), $L_{u,v}$, wit_{x_i}, 0 or 1

$k \in K^\kappa$ # Key generation $Gen(1^k, t)$
$(sk_\Omega, pk_\Omega) \leftarrow \{H_k\}_{k \in K^\kappa}$
$H_k \leftarrow pk_\Omega$ # Public Key Resolution
$(x_0, x_1, \ldots, x_{n-1}) \leftarrow X$
If $n = 2^k \mid k \in \mathbb{N}$, $v \le k$:
  $H_k(L_{2u,v+1} \| L_{2u+1,v+1})$ **if** $v < k$ **else** $H_k(x_i)$
Else False
$\left(l_{u,v}\right)_{(u \in [n/2^{k-v}])_{v \in [k]}} \leftarrow aux$ # Creates a certificate $WitCreate((pk_\Omega, sk_\Omega), \Omega_X, aux_X, x_i)$
$wit_{x_i} \leftarrow (l_{\lfloor i/2^v \rfloor} + \eta, k - v)$, $0 \le v \le k$
1 **if** $\lfloor i/2^v \rfloor \pmod 2 = 0$ **else** -1
$H_k \leftarrow pk_\Omega$, $L_{0,0} \leftarrow \Omega_X$ # Certificate authentication $Verify(pk_\Omega, \Omega_X, wit_{x_i}, x_i)$
$L_{i,k} \leftarrow H_k(L_{\lfloor i/2^v \rfloor, k-v} \| L_{\lfloor i/2^v \rfloor + 1, k-v})$ **If** $\lfloor i/2^v \rfloor \pmod 2 = 0$ **else** $L_{i,k} \leftarrow H_k(L_{\lfloor i/2^v \rfloor, k-v} \| L_{\lfloor i/2^v \rfloor, k-v})$
1 **if** wit_{x_i} is a valid witness for $x_i \in X$ **else** 0

3.3. Signature algorithm description

The hash-based post-quantum ring signature scheme explored in this work is based on the XMSS algorithm, which incorporates two primary frameworks: the WOTS+ algorithm and the Merkle tree algorithm. Below is an overview of these frameworks.

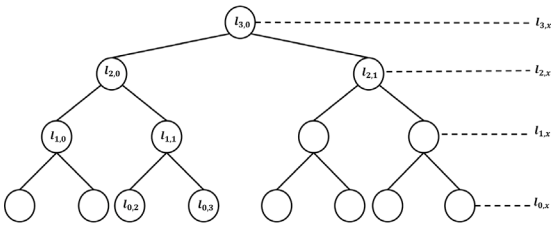

Fig. 6. A Merkle tree with a height of h = 3 and a number of leaf nodes N = 8 visualizes the reconstruction of the root node by $l_{0,2}$ nodes.

Definition 6 (*Merkle Tree Ring Signature Algorithm*)**.** The Merkle tree-based ring signature algorithm comprises four main steps: parameter definition, public key generation, signature generation, and signature verification. These steps are outlined as follows:

Step 1: Parameter Definition

The height h of the tree represents its number of layers, meaning a Merkle tree with height h has 2^h leaf nodes, indicating 2^h ring members and corresponding key pairs $(x_i, y_i), i \in [0, 2^h - 1]$.

In practical application scenarios, if the number of vehicles does not satisfy this condition, it is recommended to either introduce virtual members into the ring or divide the vehicles into multiple rings.

Step 2: Public Key Generation/Merkle Tree Construction

As shown in algorithm 2, in the Merkle tree, all leaf nodes together constitute the ring. Each member in the ring is represented by a public–private key pair corresponding to a leaf node. Each leaf node holds the hash of the public key derived from a one-time signature (OTS) scheme, while each parent node stores the hash of the concatenation of its two child nodes. This process repeats according to the same generation rule until the final root node is formed. The value of the root node is the final public key, while the private key consists of the 2^h OTS private keys x_i. The number of ring members equals the number of leaf nodes in the Merkle tree. It is essential to ensure that the number of participating members in the ring is a power of 2. The public key of each ring member corresponds to the public key from the one-time signature.

Algorithm 2 Public Key Generation

input: h, SK
output: PK
1. $node_i = Hash\left(node_{2i+1} || node_{2i}\right), i \in [0, 2^h - 1]$
2. Root=Hash(node1|| node2)
3. PK=Root

Step 3: Signature Generation Before executing the ring signature operation, the signer hashes the binary message to generate a message digest $m = H(M)$, where H is the chosen hash function, and M represents the original binary message. This digest m will be used in the subsequent steps of the signature generation process. This process is shown in algorithm 3.

Algorithm 3 Signature generation

input: M, H, one-time signature key pair (x_i, y_i)
output: σ
1 $(x_i, y_i), i \in [0, 2^h - 1]$
2 For x_i
3 Select node to perform a one-time digital signature on message M to generate signature σ_{OTS}
4 Calculate y_i authentication path $auth_i$
5 $\sigma = (i, \sigma_{OTS}, Y_i, auth_i)$

The formal signing process begins by selecting the corresponding one-time signature (OTS) key pair (x_i, y_i), specifically the ith OTS key pair. The signer then uses the private OTS key x_i to sign the message, creating a one-time signature σ_{OTS} and calculating the authentication path. The final signature comprises: the index i, the one-time signature σ_{OTS}, the public key y_i, and the authentication path for y_i, denoted $auth_i$. The signature is formally represented as $\sigma = (i, \sigma_{OTS}, Y_i, auth_i)$. The Fig. 7 illustrates the signing process using leaf nodex_2 as the signing node, where the shaded areas represent the authentication path of the signature.

Step 4: Signature Verification

As shown in Algorithm 4, signature verification begins by first verifying the one-time signature σ_{OTS}. If this check is successful, the next step involves reconstructing the Merkle tree root based on the chosen index i and the public key y_i. The reconstructed root is then compared with the stored public key. If the two match, verification is deemed successful.

Algorithm 4 Signature verification

input: σ
output: true or false
1 If
2 $VER(M, sig(OTS), Y_i) = true$
3 Reconstruct the $root^*$ node of the merkle tree according to i and Yi
4 If
5 $Root' = PK$
6 true
7 Else
8 False
9 Else
10 False

To illustrate the reconstruction process, consider nodex_2 as an example, assuming $i = 2$ and Y_2 known, along with the signature $\sigma = (2, \sigma_{OTS}, Y_2, auth_2)$. Here, $auth_2$ contains values stored in nodes 3, 8, and 13. The root node can be reconstructed as follows: node14=hash(node 12||node13), node12=hash(node8||node9), node9= hash(node2||node3) wh-ere node2 stores the value of Y_2. The computed value of node14 is the value of the reconstructed root $root^*$. This is shown in Fig. 8. By hashing upwards from the leaf nodes, if a match with the stored root node is found, the membership of signer in the ring is verified.

3.4. Application of the scheme in vehicular networks

The proposed hash-based signature scheme offers post-quantum security, protecting against quantum threats, and is highly efficient with compact signatures, ideal for resource-constrained on-board devices in IoV. It supports fast information exchange and verification in dynamic traffic environments, enhancing security and privacy, such as in accident reporting systems, while maintaining reporter anonymity. Overall, it addresses key security, efficiency, and scalability challenges in connected vehicle networks.

The application of ring signatures in IoV involves three main stages: the registration stage, the inter-vehicle communication stage, and the signature tracing and broadcast stage.

Step 1: Registration Stage

This stage consists of three main steps, First, the On-Board Unit (OBU) sends a registration request to the Trusted Authority (TA). Upon receiving the request, the TA generates a public–private key pair (PK_{OBU}, SK_{OBU}) for the OBU. In the final step, the TA returns the private key to the OBU, along with the public key and identity information bound to the blockchain network. The identity information typically includes vehicle certificates, vehicle identification numbers (VIN), and other vehicle-related data. This process ensures that vehicles

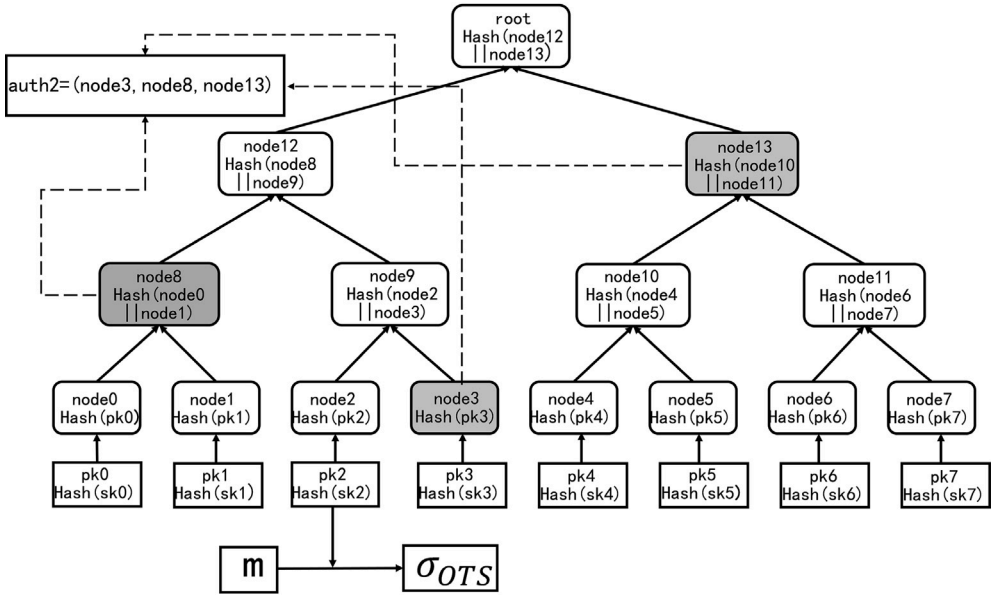

Fig. 7. Diagram of the signature generation process.

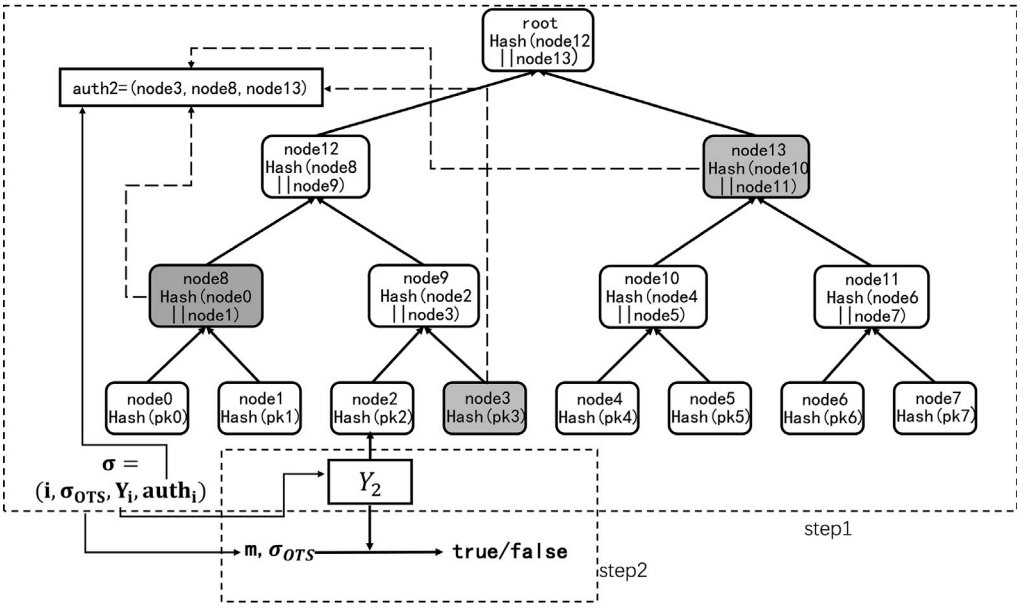

Fig. 8. Signature verification diagram.

are properly registered and recognized within the blockchain network, as illustrated in Fig. 9.

Step 2: Inter-Vehicle Communication Stage

At this stage, the OBU utilizes the public key of the Roadside Unit (RSU) PK_{RSU} to encrypt its own public key and sends it to the RSU, requesting the creation of a ring. Upon receiving the encrypted message, the RSU decrypts it using its private key to obtain PK_{OBU}, which is then added to the ring. When the number of ring members reaches the threshold of 2^h, the RSU broadcasts the ring structure, allowing all ring members to participate in signing processes.

If the threshold is not met, virtual members may be added, or the ring may be split into smaller sub-rings to ensure each ring contains 2^h members. Once the ring is established, the OBU can sign messages using a ring signature and forward them to the RSU. The RSU subsequently broadcasts the signed messages to other OBUs, which can request verification from the Verification Node (VN). The VN validates the signatures and returns the verification results to the requesting OBU, enabling secure and authenticated access to the information. This process is further illustrated in Fig. 10.

Step 3: Signature Tracing and Broadcast Stage

In the event of an accident, the OBU sends accident-related information to the RSU, which then processes and broadcasts the information to other OBUs. At the same time, the RSU forwards the signature of the OBU involved in the accident, denoted as $SIG(OBU^{acc})$ to the TA. The TA uses its private key to identify the relevant vehicle information. If the OBU is determined to be malicious, the TA revokes its identity and public key on the blockchain network. The TA then sends the revoked public key and the adverse record of the malicious OBU to the RSU. The RSU subsequently broadcasts this information to other OBUs, ensuring they are aware of the revoked identity and can exclude the malicious OBU from further network participation. This process is illustrated in Fig. 11.

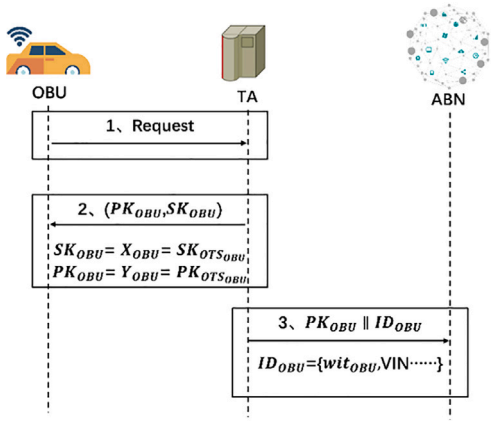

Fig. 9. Registration phase.

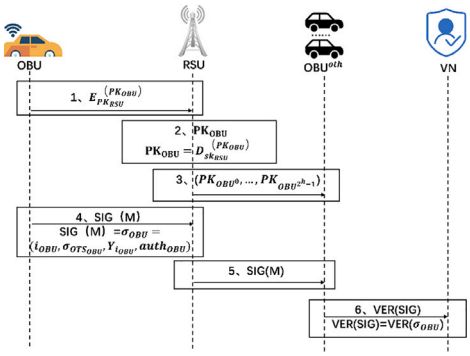

Fig. 10. Information interaction phase.

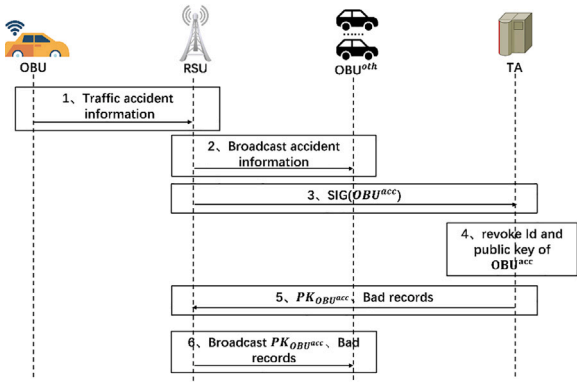

Fig. 11. Signature tracing phase.

When applying this ring signature scheme to a vehicular network system, the overall model framework is shown in Fig. 12. The primary components of the model include:

[1] On - Board Unit (OBU): Responsible for sending requests to the TA, transferring its public key to the RSU, signing messages with the ring signature, and sharing traffic accident information.
[2] Road - Side Unit (RSU): Organizes received public keys into a ring, broadcasts signatures, accident information, and adverse records to other vehicles, and forwards accident-related signatures to the TA.
[3] Trusted Authority (TA): Generates key pairs for the OBU, uploads these to the blockchain network, and, in the event of an accident, sends the public key and adverse record of the vehicle involved to the RSU.
[4] Verification Node (VN): Responsible for verifying signature requests sent by other vehicles.
[5] Anonymous Blockchain Network (ABN): In this model, vehicle public keys are stored in the blockchain network, providing a secure and anonymous framework for identity management.

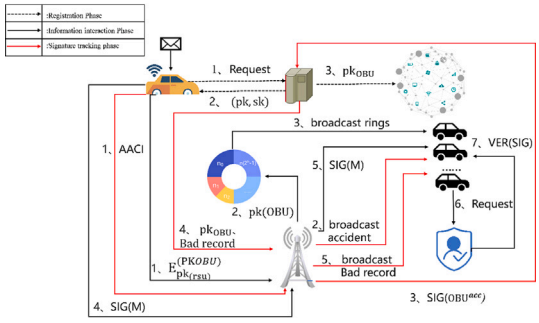

Fig. 12. IOV model based on post-quantum ring signature.

In addition to the interactions between the OBU and the TA, as well as between the OBU and RSU in the aforementioned process, within a specific segment of roadway, the OBU is also capable of engaging with pedestrians, road infrastructure, and stations located within that segment.

In general, the integrity and privacy protection of data transmission are more emphasized in interactions between vehicles and other vehicles, as well as roadside units. However, interactions between vehicles and pedestrians often involve location verification and identity confirmation. In a vehicular networking system, vehicles may need to verify both the identity and location of pedestrians, while using post-quantum ring signatures to ensure the integrity and non-repudiation of pedestrian information.

4. Security analysis

4.1. Safety assessment

The proposed scheme possesses the following characteristics:

(1) Anonymity: Ring signatures inherently support anonymity, protecting the identity of signer. Assuming an attacker has obtained a valid ring signature generated only by members within the ring, if the ring contains n members, the probability that the attacker identifies the true signer is $1/n$. For any member other than the signer, the probability of knowing the identity of signer is $1/n-1$.

(2) Privacy: The generation of a ring signature relies solely on the signer within the ring, with no involvement from other ring members, thus preserving the privacy of the signer.

(3) Post-Quantum Security: This scheme employs a post-quantum ring signature approach based on Merkle trees, leveraging hash-based and post-quantum secure mathematical problems. This design provides robust security against quantum computing threats. The use of hash-based post-quantum ring signatures combines the strong properties of hash functions with quantum-resilient security, maintaining integrity even under potential quantum computing attacks.

(4) Efficiency: The computational efficiency of hash functions makes this scheme suitable for a variety of application scenarios.

(5) Unforgeability: The scheme ensures unforgeability through the one-way and irreversible properties of hash functions in constructing hash chains. Thus, it is highly challenging for anyone other than the legitimate signer to forge a signature within this scheme.

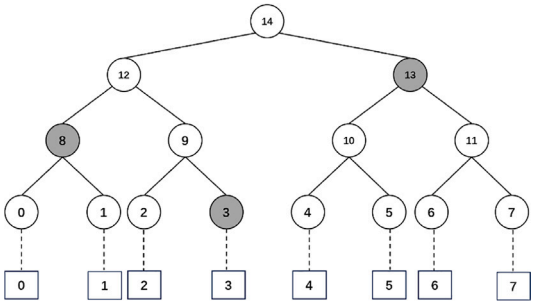

Fig. 13. Authentication path diagram of a node with index i = 2.

4.2. Security proof

The following section provides security proofs and discussions for the proposed scheme:

Lemma 1. *If a one-time signature scheme passes verification and the reconstructed Merkle root Root* matches the original Merkle root Root, then the signature is valid.*

Proof. Suppose the index $i = 2$ is chosen for the one-time signature key used in the message signature. The nodes from index $i = 2$ to the root node traverse nodes [2, 9, 12], with sibling nodes [3, 8, 13], forming a verification path [3, 8, 13], In Fig. 13, we illustrate the verification pathway of the leaf node indexed at 2, which is depicted as the gray node. Reconstructing the root Root* follows these steps:

$$Node(9) = \text{Hash}(node(2) \parallel node(3))$$

$$Node(12) = \text{Hash}(node(9) \parallel node(8))$$

$$Node(14) = \text{Hash}(node(12) \parallel node(13))$$

The value of node 9 is computed from nodes 2 and 3, the value of node 12 is computed from nodes 9 and 8, and the value of the root node Root* (node 14) is computed from nodes 12 and 13. This computed Root* value is then compared with the public key. Clearly, the hash of Root* matches the original public key. The proof process for any other node is identical, thus confirming the correctness of the signature.

Theorem 1. *The proposed post-quantum ring signature scheme preserves anonymity.*

Assuming a valid signature $\sigma = (i, \sigma_{OTS}, Y_i, auth_i)$, where each value of i is within the appropriate range $i \in [0, 2^h - 1]$, the probability that any other person can identify the true signer is $1/2^h$ (for a ring with 2^h members). For other ring members, the probability of knowing the identity of signer is $1/(2^h - 1)$.

Theorem 2. *The proposed ring signature scheme is unforgeable.*

Proof. Suppose an attacker A could successfully forge a ring signature with non-negligible probability P within polynomial time. We construct a simulator S to challenge a ring signature algorithm claimed to be secure by challenger C as follows:

Step 1: The challenger initializes n signing instances with the MSS signing algorithm, generating n key pairs (sk, pk) and sends all public keys pk to simulator S.

Step 2: Upon receiving the public keys, S initializes the ring signature algorithm by randomly selecting additional parameters and forwarding the public keys to attacker A.

Step 3: In the query phase, A selects a message M and sends it to S. Following the ring signature algorithm, S randomly selects a user s to generate the ring signature, computes Y_s, and forwards it to C. C computes the corresponding σ_s, which S returns as a complete ring signature to A.

Step 4: In the challenge phase, A sends M and an unobserved forged ring signature to S, which calculates the corresponding Y_s of the forged signer and submits (Y_s, σ_s) to C. If C verifies Y_s and σ_s as valid, then S has successfully forged a signature, with output 1; otherwise, S fails, outputting 0.

Since A can break the scheme with non-negligible probability P, we deduce that $pr(output(Game) = 1) = p$, allowing S to break the post-quantum ring signature algorithm with non-negligible probability. However, this contradicts the assumed security of scheme, proving that A cannot successfully forge signatures in polynomial time.

Theorem 3. *If the underlying hash function family $\{H_k\}, k \in K_K$ is a collision-resistant family, then the proposed hash-based post-quantum ring signature scheme is collision-resistant.*

Proof. During initialization, this reduction interacts with a collision-resistant hash function challenge to acquire H_k and completes initialization per the original protocol. If an attacker generates a collision within the accumulator, this implies that the reduction knows two distinct inputs that collide under H_k, with the collision probability bounded by the collision resistance of hash function.

Theorem 4. *If the employed hash functions are one-way, then the proposed Merkle-tree-based post-quantum ring signature scheme is unforgeable under chosen-message attacks.*

Let $n, w, m \in N, with w, m = poly(n)$, and let the function family $F_n = f_k : \{0,1\}^n \to \{0,1\}^n$ where $k \in \{0,1\}^n$ satisfy second-preimage resistance and one-way properties. The variable t represents the computational time. The term $\omega \cdot InSec^{UD}(F_n; t^*)$ reflects the undetectability (UD) security of the function family F_n, while $InSec^{OW}(F_n; t')$ represents its one-way(OW) security. Additionally, the term $\omega \cdot InSec^{SPR}(F_n; t')$ denotes the second-preimage resistance(SPR) security, scaled by the parameter ω. The formal definitions of EU-CMA and SPR are provided in [14], and will not be elaborated on here.

We define the unforgeability insecurity under chosen-message attack of WOTS+ as follows:

$$\begin{aligned}
&\text{lnSec}^{\text{EU-CMA}}(\text{WOTS}^{+}(1^n, w, m); t, 1) \\
&\quad \le w \cdot \ln \text{Sec}^{\text{UD}}(F_n; t^*) + wl \\
&\qquad \cdot \max\{\ln \text{Sec}^{\text{OW}}(F_n; t'), w \cdot \ln \text{Sec}^{\text{SPR}}(F_n; t')\} \text{ with } t' \\
&\quad = t + 3lw \text{ and } t^* \\
&\quad = t + 3lw + w - 1
\end{aligned}$$

For WOTS+ combined with Merkle trees, the non-forgeability under chosen-message attacks on the Merkle tree can be defined as follows:

$$\begin{aligned}
&\text{InSec}^{\text{EU-CMA}}\left(\text{Merkle-tree}\left(1^n, T = 2^h\right); t, 1\right) \\
&\qquad \le 2 \cdot \max\left\{2^{h+\log_2 \ell - 1} \cdot \right. \\
&\qquad\qquad \left. \text{InSec}^{\text{SPR}}\left(\text{WOTS+}\left(1^n, \omega, m\right); t, 1\right)\right\}
\end{aligned}$$

Using the derived insecurity function for the Merkle tree combined with W-OTS, which employs pseudorandom key generation and Gen_{2^h} we arrive at the following results:

$$\begin{aligned}
&\text{InSec}^{\text{EU-CMA}}\left(\text{XMSS}(1^n, T = 2^h); t, 1\right) \\
&\le \text{InSec}^{\text{EU-CMA}}\left(\text{WOTS+}(1^n, \omega, m); t, 1\right) \\
&\quad + \text{InSec}^{\text{EU-CMA}}\left(\text{Merkle-tree}(1^n, T = 2^h); t, 1\right) \\
&= \text{InSec}^{\text{PRF}}(F_n, t' + 2^h, 2^h) \\
&\quad + 2 \max\left\{\begin{array}{l}
(2^{h+\log_2 l - 1}) \cdot \text{InSec}^{\text{SPR}}(H_n, t'), \\
2^h \cdot \text{InSec}^{\text{PRF}}(F_n; t' + l, l) + \\
\quad \omega \cdot \text{InSec}^{\text{UD}}\left(F_n; t^* + \max\left\{\begin{array}{l}\text{InSec}^{\text{OW}}(F_n; t'), \\ \text{InSec}^{\text{SPR}}(F_n; t')\end{array}\right\}\right)
\end{array}\right\}.
\end{aligned}$$

Table 5
Test 16 XMSS-SHA2_10_256 signatures.

Number	Signature time	Verification time
0	1.990014	0.001119
1	1.980151	0.000947
2	1.969849	0.001210
3	1.965888	0.001184
4	1.969898	0.001056
5	1.980296	0.001144
6	2.017889	0.001093
7	2.054971	0.001101
8	2.016147	0.001241
9	2.020737	0.001267
10	1.954583	0.001016
11	2.021315	0.001060
12	2.029765	0.001043
13	2.057487	0.001016
14	1.958401	0.001081
15	1.990919	0.001053

To prove XMSS is unforgeable under chosen-message attacks, we consider the following factors:

Random Oracle Model: Assuming the hash function behaves as a random oracle, an attacker has no foreknowledge of input–output pairs.

Irreversibility: WOTS+ security relies on the irreversibility of hash chains; given a hash value $H_i(x)$, finding the predecessor $H_{i-1}(x)$ is infeasible.

Collision Resistance: The hash function must resist collisions, making it nearly impossible for an attacker to produce distinct messages that yield identical hash chains.

5. Performance analysis

This study evaluates the performance of proposed scheme in densely trafficked urban areas, focusing particularly on resistance to quantum attacks. The experiments are based on the Merkle tree-ring signature scheme, with a primary emphasis on security strength, as attacks in the IoV environments are expected to become increasingly complex, especially with the advent of quantum attacks. Consequently, a high-security, quantum-resistant signature scheme is essential for the IoV systems.

The primary operations in the signature scheme include generating public and private keys, measuring the time required for message signing and verification, and instantiating the SHA-256 function as the underlying hash function. Key parameters include the security parameter n, the Winternitz parameter ω, and the number of ring members, with specific values assigned to each. These operations allow us to measure metrics such as key generation time, signature generation time, and signature verification time.

In this scheme, the digital signature algorithm is set to XMSS-SHA2-10-256, utilizing the SHA-256 hash function with a Merkle tree height of 10, enabling a maximum of $2^{10} = 1024$ possible ring signatures. The number of signature tests is set to 16 to balance efficiency and data stability, ensuring valid results without excessive resource consumption.

To present the data more intuitively, the experimental results of the 16 tests shown in Table 5 are depicted in graphical form, resulting in Fig. 14 and Fig. 15. Fig. 14 illustrates the signature generation times across the 16 tests, while Fig. 15 displays the signature verification times. These figures show that both the signature generation time and verification time fluctuate within a certain range, indicating variability rather than fixed values. Select one of the 16 test results to compare with relevant literature studies. The attributes of comparison include key generation time, signature generation time, signature verification time, resistance to quantum attacks, anonymity, traceability, and application to the IoV. The comparison results are drawn in Tables 6 and 7, In our scheme, we set the parameters as n = 32, $\omega = 16$, the height of Merkle tree as 10, and the number of ring members as 2^{10}. Among them, HBS stands for the scheme based on hash and LBS stands for a scheme based on lattices.

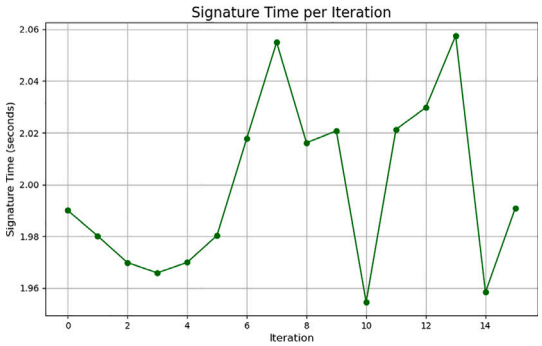

Fig. 14. Signature generation time of 16 test results.

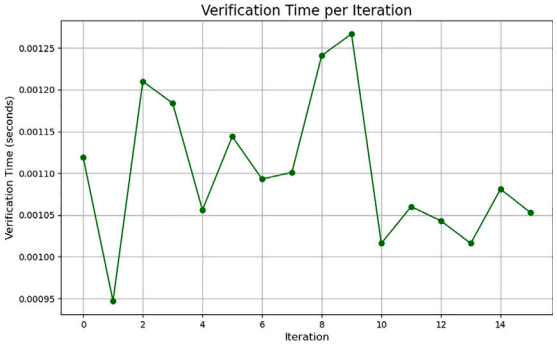

Fig. 15. Signature verification time of 16 test results.

Table 6
Signature efficiency comparison table.

	Scheme	Number of Members	Key generation time/s	Signature time/s	Verification time/s
OURS	HBS	2^{10}	2.06	1.97	9.47e−04
[33]	LBS	10	0.07	0.06	0.04
[32]	LBS	–	34.1e−06	9.59e−05	3.49e−05
[25]	HBS	2^{10}	–	0.16	0.11

Table 7
Function comparison table of the scheme.

	Scheme	Post-quantum security	Anonymity	Traceability	Application to IOV
OURS	HBS	YES	YES	YES	YES
[33]	LBS	NO	YES	YES	YES
[32]	LBS	YES	NO	NO	YES
[25]	HBS	YES	YES	YES	NO

Comparing the scheme proposed in this paper with the scheme in [33], it can be seen that the post-quantum ring signature scheme based on Merkle tree has great advantages. First, in this evaluation, the number of ring members our scheme can accommodate is 2^{10}, which is much larger than the number of ring members evaluated in [33]. When the road section is wider and crowded, the scheme proposed in this paper is more suitable. Secondly, this scheme has post-quantum security, which is more secure; Moreover, although the key generation time of our scheme is slightly longer than that of the scheme with fewer ring members in [33], it is much faster in terms of signature time and verification time, especially the verification time is nearly 44 times faster than that of [25].

Compared with the scheme in [32], the outstanding feature of the scheme in this paper is ring signature, which has anonymity and traceability, making it more suitable for the Internet of vehicles environment. In addition, the scheme in this paper uses Merkle tree structure, which reduces the storage cost of public key and signature. In general, lattice signature may require special optimization in high performance computing. The algorithm maturity is not high, but the underlying hash function of the post-quantum ring signature scheme in this paper is SHA-256, and the SHA-256 function has passed the test of time in many practical applications, and has high algorithm maturity.

Comparing the scheme in this paper with the scheme in [25], it can be seen that both papers are based on hash function. The advantages of the scheme in this paper are as follows: First, although the time of signature generation in [25] is nearly 12 times faster than that in this paper, the time of signature verification in this paper is nearly 100 times faster than that in [25]. In addition, the scheme in this paper is also applied to the vehicle networking model.

As shown in Table 7, this study compares the attributes of "Post-quantum", "Anonymity", "Traceability", and "Application to IoV". The comparison reveals that our scheme offers post-quantum security, anonymity, traceability, and the ability to apply to IoV, with the advantages of our proposed scheme becoming more evident through this comprehensive comparison.

6. Conclusion

The hash-based post-quantum ring signature scheme offers advantages such as high signature efficiency, good scalability, and independence from complex mathematical assumptions. In the context of increasing security threats posed by advancements in quantum computing, applying post-quantum ring signatures in IoV can enhance anonymity and privacy protection while ensuring quantum-resistant security. This paper presents a hash-based post-quantum ring signature scheme built on the XMSS algorithm and demonstrates its application in the IoV system. The proposed scheme is analyzed and proven secure. Performance analysis is conducted following 16 experimental tests, with comparisons made to other similar schemes. The results show that the proposed scheme exhibits significant advantages in signature verification time compared to other approaches. This is due to the efficient hash computations and Merkle tree verification paths, which maintain low time complexity and high efficiency even with large data sets. Moreover, the scheme satisfies the properties of quantum resistance, anonymity, traceability, and applicability to IoV.

Future research will aim to further improve the practicality and security of the scheme in response to the evolving threats posed by quantum computing, and second, interdisciplinary collaboration can be strengthened in future research to provide valuable insights for optimizing solutions in real-world scenarios.

CRediT authorship contribution statement

Shuanggen Liu: Conceptualization. **Xiayi Zhou:** Writing – original draft. **Xu An Wang:** Supervision. **Zixuan Yan:** Investigation. **He Yan:** Formal analysis. **Yurui Cao:** Resources.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant No. 62172436.The first author and the third author are the corresponding authors of this paper.

Data availability

No data was used for the research described in the article.

References

[1] I. Wanger, Car production: Number of cars produced worldwide, Statista (2020).
[2] Patrick Miner, Barbara M. Smith, Anant Jani, Geraldine McNeill, Alfred Gathorne-Hardy, Car harm: A global review of automobility's harm to people and the environment, J. Transp. Geogr. 115 (2024) 103817.
[3] Juan Contreras-Castillo, Sherali Zeadally, Juan Antonio Guerrero-Ibañez, Internet of vehicles: Architecture, protocols, and security, IEEE Internet Things J. 5 (5) (2018) 3701–3709, http://dx.doi.org/10.1109/JIOT.2017.2690902.
[4] David Deutsch, Quantum theory, the Church–Turing principle and the universal quantum computer, Proc. R. Soc. A 400 (1818) (1985) 97–117.
[5] Rasha Shajahan, Kurunandan Jain, Prabhakar Krishnan, A survey on NIST 3 rd round post quantum digital signature algorithms, in: 2024 5th International Conference on Mobile Computing and Sustainable Informatics, ICMCSI, IEEE, 2024, pp. 132–140.
[6] David A. Cooper, Daniel C. Apon, Quynh H. Dang, Michael S. Davidson, Morris J. Dworkin, Carl A. Miller, et al., Recommendation for stateful hash-based signature schemes, NIST Spec. Publ. 800 (208) (2020) 208–800.
[7] Samira El Madani, Saad Motahhir, Abdelaziz El Ghzizal, Internet of vehicles: concept, process, security aspects and solutions, Multimedia Tools Appl. 81 (12) (2022) 16563–16587.
[8] Cesar Castellon, Swapnoneel Roy, Patrick Kreidl, Ayan Dutta, Ladislau Bölöni, Energy efficient merkle trees for blockchains, in: 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom, IEEE, 2021, pp. 1093–1099.
[9] Daniel J. Bernstein, Andreas Hülsing, Stefan Kölbl, Ruben Niederhagen, Joost Rijneveld, Peter Schwabe, The SPHINCS+ signature framework, in: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, 2019, pp. 2129–2146.
[10] Kaiyi Zhang, Hongrui Cui, Yu Yu, SPHINCS-α: A compact stateless hash-based signature scheme, 2022, Cryptology ePrint Archive.
[11] Mikhail Kudinov, Andreas Hülsing, Eyal Ronen, Eylon Yogev, SPHINCS+ C: Compressing SPHINCS+ with (almost) no cost, 2022, Cryptology ePrint Archive.
[12] Sun Siwei, Liu Tianyu, Guan Zhi, SM3-based post-quantum digital signature schemes, J. Cryptologic Res. 10 (1) (2023) 46.
[13] Andreas Hülsing, Mikhail Kudinov, Recovering the tight security proof of SPHINCS+, in: International Conference on the Theory and Application of Cryptology and Information Security, Springer, 2022, pp. 3–33.
[14] Andreas Hülsing, Denis Butin, Stefan Gazdag, Joost Rijneveld, Aziz Mohaisen, XMSS: Extended Merkle Signature Scheme, Technical Report, 2018.
[15] Jan Philipp Thoma, Tim Güneysu, A configurable hardware implementation of XMSS, 2021, Cryptology ePrint Archive.
[16] Siwei Sun, Tianyu Liu, Zhi Guan, Yifei He, Jiwu Jing, Lei Hu, Zhenfeng Zhang, Hailun Yan, XMSS-SM3 and MT-XMSS-SM3: Instantiating extended Merkle signature schemes with SM3, 2022, Cryptology ePrint Archive.
[17] Andreas Hülsing, W-OTS+–shorter signatures for hash-based signature schemes, in: Progress in Cryptology–AFRICACRYPT 2013: 6th International Conference on Cryptology in Africa, Cairo, Egypt, June 22-24, 2013. Proceedings 6, Springer, 2013, pp. 173–188.
[18] Kaiyi Zhang, Hongrui Cui, Yu Yu, Revisiting the constant-sum winternitz one-time signature with applications to SPHINCS+ and XMSS, in: Annual International Cryptology Conference, Springer, 2023, pp. 455–483.
[19] Xie Jia, Liu Shizhao, Wang Lu, Research progress and prospects of ring signature technology., J. Front. Comput. Sci. Technol. 17 (5) (2023).
[20] Rohit Chatterjee, Kai-Min Chung, Xiao Liang, Giulio Malavolta, A note on the post-quantum security of (ring) signatures, in: IACR International Conference on Public-Key Cryptography, Springer, 2022, pp. 407–436.
[21] Yuxi Xue, Xingye Lu, Man Ho Au, Chengru Zhang, Efficient linkable ring signatures: new framework and post-quantum instantiations, in: European Symposium on Research in Computer Security, Springer, 2024, pp. 435–456.
[22] Abida Haque, Alessandra Scafuro, Threshold ring signatures: new definitions and post-quantum security, in: Public-Key Cryptography–PKC 2020: 23rd IACR International Conference on Practice and Theory of Public-Key Cryptography, Edinburgh, UK, May 4–7, 2020, Proceedings, Part II 23, Springer, 2020, pp. 423–452.
[23] Maxime Buser, Joseph K. Liu, Ron Steinfeld, Amin Sakzad, Post-quantum id-based ring signatures from symmetric-key primitives, in: International Conference on Applied Cryptography and Network Security, Springer, 2022, pp. 892–912.
[24] J. Odoom, X. Huang, Z. Zhou, et al., Linked or unlinked: A systematic review of linkable ring signature schemes, J. Syst. Archit. 134 (2023) 102786.
[25] Shiwei Xu, Tao Wang, Ao Sun, Yan Tong, Zhengwei Ren, Rongbo Zhu, Houbing Herbert Song, Post-quantum anonymous, traceable and linkable authentication scheme based on blockchain for intelligent vehicular transportation systems, IEEE Trans. Intell. Transp. Syst. (2024).

[26] Nyothiri Aung, Tahar Kechadi, Tao Zhu, Saber Zerdoumi, Tahar Guerbouz, Sahraoui Dhelim, Blockchain application on the internet of vehicles (iov), in: 2022 IEEE 7th International Conference on Intelligent Transportation Engineering, ICITE, IEEE, 2022, pp. 586–591.

[27] Haibin Zhang, Jiajia Liu, Huanlei Zhao, Peng Wang, Nei Kato, Blockchain-based trust management for internet of vehicles, IEEE Trans. Emerg. Top. Comput. 9 (3) (2020) 1397–1409.

[28] Mirador Labrador, Weiyan Hou, Implementing blockchain technology in the internet of vehicle (IoV), in: 2019 International Conference on Intelligent Computing and Its Emerging Applications, ICEA, IEEE, 2019, pp. 5–10.

[29] Y. Liu, Q. Xia, X. Li, et al., An authentication and signature scheme for UAV-assisted vehicular ad hoc network providing anonymity, J. Syst. Archit. 142 (2023) 102935.

[30] X. Feng, X. Wang, K. Cui, et al., A distributed message authentication scheme with reputation mechanism for internet of vehicles, J. Syst. Archit. 145 (2023) 103029.

[31] S. Thapliyal, M. Wazid, D.P. Singh, et al., Robust authenticated key agreement protocol for internet of vehicles-envisioned intelligent transportation system, J. Syst. Archit. 142 (2023) 102937.

[32] Nikhil Verma, Swati Kumari, Pranavi Jain, Post quantum digital signature change in iota to reduce latency in internet of vehicles (iov) environments, in: 2022 International Conference on IoT and Blockchain Technology, ICIBT, IEEE, 2022, pp. 1–6.

[33] Cui Yongquan, Cao Ling, Zhang Xiaoyu, Privacy protection of internet of vehicles based on lattice-based ring signature, Chinese J. Comput. 42 (5) (2019) 980–992.

[34] Cesar Castellon, Swapnoneel Roy, Patrick Kreidl, Ayan Dutta, Ladislau Bölöni, Energy efficient merkle trees for blockchains, in: 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom, IEEE, 2021, pp. 1093–1099.

[35] David Derler, Sebastian Ramacher, Daniel Slamanig, Post-quantum zero-knowledge proofs for accumulators with applications to ring signatures from symmetric-key primitives, in: Post-Quantum Cryptography: 9th International Conference, PQCrypto 2018, Fort Lauderdale, FL, USA, April 9-11, 2018, Proceedings 9, Springer, 2018, pp. 419–440.

[36] Xinyu Zhang, Ron Steinfeld, Joseph K. Liu, Muhammed F. Esgin, Dongxi Liu, Sushmita Ruj, DualRing-PRF: Post-quantum (linkable) ring signatures from Legendre and power residue PRFs, in: Australasian Conference on Information Security and Privacy, Springer, 2024, pp. 124–143.

[37] David A. Cooper, Daniel C. Apon, Quynh H. Dang, Michael S. Davidson, Morris J. Dworkin, Carl A. Miller, et al., Recommendation for stateful hash-based signature schemes, NIST Spec. Publ. 800 (208) (2020) 208–800.

[38] Ralph C. Merkle, A certified digital signature, in: Conference on the Theory and Application of Cryptology, Springer, 1989, pp. 218–238.