IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 15 November 2022, accepted 7 December 2022, date of publication 26 December 2022, date of current version 29 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3232301

RESEARCH ARTICLE

A Lightweight BT-Based Authentication Scheme for Illegal Signatures Identification in VANETs

EKO FAJAR CAHYADI[1,2], (Member, IEEE), AND **MIN-SHIANG HWANG**[1,3], (Member, IEEE)

[1]Department of Computer Science and Information Engineering, Asia University, Taichung 41354, Taiwan
[2]Faculty of Telecommunication and Electrical Engineering, Institut Teknologi Telkom Purwokerto, Purwokerto 53147, Indonesia
[3]Department of Medical Research, China Medical University Hospital, China Medical University, Taichung 40402, Taiwan

Corresponding author: Min-Shiang Hwang (mshwang@asia.edu.tw)

This work was supported by the Ministry of Science and Technology, Taiwan, under Contract MOST 109-2221-E-468-011-MY3, Contract MOST 108-2410-H-468-023, and Contract MOST 111-2622-8-468-001-TM1.

ABSTRACT Research related to vehicular ad hoc networks (VANETs) has received significant attention in recent years. Despite all the advantages, the security and privacy in VANETs still become the main challenge that is widely open to discussion. The authentication scheme plays a substantial role to guarantee the security and privacy of information circulation and verification efficiency in VANETs. In this high-density environment, a scalability issue would emerge when the number of message-signature pairs received by a roadside unit (RSU) or vehicles becomes large. This issue happens because those entities cannot sequentially verify each received signature according to the required time limit. Researchers believe that the symmetric cryptography-based authentication scheme provides a lightweight verification operation, which leads to low computation cost. Combined with the batch verification process, this approach can be beneficial. However, to the best of our knowledge, not many of those related schemes provide a realistic scenario regarding illegal signatures' appearance. Could the system identify the forged messages? Is it still efficient enough to do such an operation? In this paper, we propose a lightweight binary tree-based (BT-based) authentication scheme with a batch verification mechanism, that could efficiently identify a modest amount of illegal signatures in the sum of messages. To even improve the operation, we combine our BT-based batch verification scheme with our vehicle reputation scoring system. By this approach, we can guarantee the best-case scenario (the most desirable condition) in our BT-based identification appear as much as possible. Hence, the computation cost can be kept low.

INDEX TERMS Authentication, batch verification, security and privacy, symmetric cryptography, VANETs.

I. INTRODUCTION

Vehicular ad hoc networks (VANETs) have been attracting many researchers since their emergence in early 2000. Its capability in providing information dissemination among the vehicles will become the future of our road transportation systems. This approach aims to improve driving safety as its primary goal. Since VANETs are loaded with intelligent transportation system (ITS) properties, it will make all of these smart vehicles could communicate with each other via vehicle-to-vehicle (V2V) and to the roadside unit (RSU) via vehicle-to-infrastructure (V2I) communications [1], [2], [3].

As depicted in Figure 1, VANETs are composed of three major entities, *i.e.*, trusted authority (TA), RSU, and onboard unit (OBU). TA acts as the trust and security management center of the entire VANETs entities. Its job, including registration and parameters generation for RSUs and OBUs after they join the network. It also revokes nodes in the case of vehicles broadcasting fraud messages or performing malicious behavior [4]. Meanwhile, RSUs are fixed infrastructures located along the road at dedicated locations, such as intersections or parking lots, which are fully controlled by TA [5]. They act as a bridge between TA

The associate editor coordinating the review of this manuscript and approving it for publication was Theofanis P. Raptis.

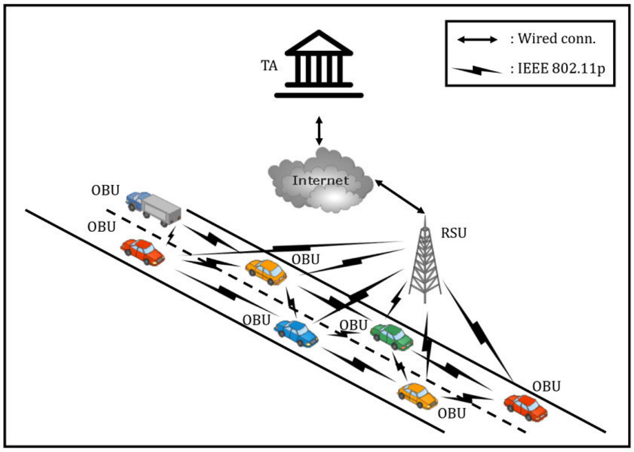


FIGURE 1. The topology of VANETs.

and vehicles (OBUs). RSUs are connected to TA by wire and OBUs by a wireless channel.

In this new environment, a vehicle could broadcast a traffic-related message with hundreds of other vehicles (V2V) or RSUs (V2I) every 100-300 ms [6]. An OBU is equipped in every vehicle as a transceiver unit. It will broadcast information like position, speed, and direction to improve the road environment, traffic safety, and create mutual awareness of the vehicles around local traffic conditions [7].

Despite all its advantages, security and privacy become significant concerns due to its unique characteristics, *e.g.*, open wireless communication, rapid topology shift, and many message exchange [8]. The most common approach to protecting the confidentiality of substantial message exchange in VANETs is by signing each message with a digital signature. Meanwhile, an efficient anonymous authentication scheme for VANETs is required to meet the strict time requirements in VANETs [9].

On the other hand, a scalability issue would emerge when the number of signatures received by a roadside unit (RSU) or vehicles becomes large. Therefore, a batch verification scheme was introduced to reduce the computational overhead in RSU and OBU in verifying a large number of signatures [10]. Batch verification is a method for verifying large amounts of digital signatures at once. This method can reduce the computational cost compared to one-by-one schemes [11]. Without batch verification, a sequentially large number of signatures could take a long time and undeniably cause a bottleneck at the RSUs and OBUs. If roughly 180 vehicles are kept within the communication range of an RSU, and each vehicle is sending a message every 300 ms; this means a verifier (such as an RSU) has to verify 600 messages per second [10].

In this paper, we propose a lightweight symmetric authentication scheme with a binary tree-based (BT-based) batch verification mechanism. Our lightweight authentication scheme is based on Liu et al.'s [12] SEGKA scheme, which has been rectified and improved. In this RSU-centric scheme, RSU has the responsibility to authenticate and compute/update the group key for vehicles in its area. Meanwhile, in our BT-based verification scheme, we apply our reputation scoring mechanism to efficiently reduce the computation cost, particularly in the case when illegal signatures appear in the batch. Illegal signatures produced by adversaries may pose a severe consequence to the recipient. Meanwhile, detecting it in a group of messages can be a difficult and time-consuming process [13]. Therefore, to improve the situation, by implementing our BT-based authentication scheme, RSU will get a substantial assist to speed up the verification process.

For a better understanding, the rest of this paper is organized as follows. In Section II, we review the related work. Section III introduces preliminaries about the system design, security and privacy requirements, concepts of bilinear maps, and a brief explanation about reputation management. Our proposed scheme is conveyed in Section IV. In Section V, we discuss the illegal signature identification scheme with BT-based batch verification. Meanwhile, Section VI discusses the efficiency of the BT-based scheme with our vehicle reputation mechanism is presented. The security and performance analyses of our scheme are in Section VII. Finally, the conclusion is conveyed in Section VIII.

II. RELATED WORK

In 2016, Vijayakumar et al. [14] proposed symmetric key-based dual authentication and dual key group management security protocol to improve security in VANETs. The scheme intends to avoid a malicious vehicle $\mathcal{M}$ using the secret key of any legitimate number for participating in VANETs. It relies on the fingerprint and hashes code (HC) for the authentication process. The authors claimed that the mechanism could withstand the replaying attack by appending it with an updated timestamp and the packet's transmission. Another notable authentication scheme based on identity-based cryptography was proposed by Tzeng et al. [15] in 2017. They improved Lee and Lai's [16] scheme by revealing its vulnerability to the identity privacy-preserving attack, the forgery attack, and the anti-traceability attack. It has proven that their scheme is survived against security and privacy requirement issues, such as message authentication, identity privacy-preserving, traceability, non-repudiation, unlinkability, and replay attacks. They also gave a more effective computation and communications delay value compared with any equivalent bilinear identity-based batch verification (IBV) schemes [17].

In 2017, Azees et al. [18] proposed a public key infrastructure-based (PKI-based) efficient anonymous authentication scheme with conditional privacy-preserving (EAAP) in VANETs. EAAP provides both V2I and V2V communications. In EAAP, TA doesn't require storing the vehicle's and RSU's certificates. Instead, it is self-generated by itself. EAAP has two authentications processes: in vehicle side and the RSU side. Vehicle must register themselves to TA before

TABLE 1. Literature survey.

Authors	Literature	Main feature and limitation
Liu *et al.* [12]	A secure and efficient group key agreement scheme for VANET.	**Main feature**: a symmetric cryptography-based with batch verification scheme that provides group key agreement mechanism for entering and leaving vehicles. **Limitation**: suffered from identity-privacy violation, replay, and the denial of service (DoS) attacks. No illegal signatures identification mechanism.
Tzeng *et al.* [15]	Enhancing security and privacy for identity-based batch verification scheme in VANETs.	**Main features**: an identity-based signature (IBS) cryptography with batch verification scheme, that built on bilinear pairings. The most efficient IBV scheme in VANETs. **Limitation**: no illegal signatures identification mechanism.
Azees *et al.* [18]	EAAP: efficient anonymous authentication with conditional privacy-preserving scheme for vehicular ad hoc networks.	**Main features**: a PKI-based authentication scheme for both V2V and V2I communications, with two authentication processes (vehicle and RSU). **Limitations**: vulnerable on anonymity and untraceability. No illegal signatures identification mechanism.
Gu *et al.* [19]	An improved EAAP scheme for vehicular ad hoc networks.	**Main feature**: the scheme improves Azees *et al.*'s scheme by providing certificateless mutual authentication between OBU and RSU. **Limitation**: no illegal signatures identification mechanism.
Jiang *et al.* [20]	BAT: a robust signature scheme for vehicular networks using binary authentication tree.	**Main feature**: an IBS cryptography-based authentication scheme with binary authentication tree (BAT) mechanism for illegal signatures identification. **Limitation**: insecure against forgery attacks, replay attacks, and Sybil attacks [21], [35].
Wang *et al.* [21]	An improved binary authentication tree algorithm for vehicular networks.	**Main feature**: the scheme improves Jiang *et al.*'s BAT scheme by providing random vectors in the batch verification phase.
Shim *et al.* [35]	Reconstruction of a secure authentication scheme for vehicular ad hoc networks using a binary authentication tree.	**Main feature**: an ID-based aggregate signature scheme (CPP-BAT) that improves Jiang *et al.*'s BAT scheme. **Limitation**: no illegal signatures identification mechanism.

TABLE 2. Notations of this paper.

Notation	Definition
TA	Trusted authority
RSU	Roadside unit
s	Master key of TA
P_{pub}	Public key of TA
PID_i	The pseudo identity of vehicle
RID_i	The real identity of vehicle
V_i	The vehicle number
VID_i	V_i's verification identity
$ENC_k(M)$	Encrypting function of M using key k
$DEC_k(M)$	Decrypting function of M using key k
$h(\cdot)$	One secure one-way hash function
$H(\cdot)$	A map-to-point hash function
PK_{RSU}	A public key for the RSU
SK_{RSU}	A private key for the RSU
T_i	The freshness of time
G_1	The cyclic additive group
G_2	The cyclic multiplicative group

getting communicate to another vehicle (V2V). Then vehicles must authenticate themselves to any RSUs in every area, in order to obtain particular location-based safety information (LBSI). The scheme itself was declared secure against impersonation attacks, bogus message attacks, message modification attacks, and providing privacy preservation and anonymity during the authentication of vehicles and RSUs.

However, in 2020, Gu et al. [19] show if Azees et al.'s EAAP is vulnerable against location tracking attacks, and in case of dispute, $\mathcal{M}$ cannot be traced by the TA. Compared to Azees et al.'s [18] scheme, Gu et al.'s scheme realizes a mutual authentication between OBU and RSU, RSU is authenticated without using a certificate, prevents the anonymous identity of the vehicles from being monitored and tracked, and uses a new tracking method for $\mathcal{M}$.

Meanwhile, related to the idea of the BT-based scheme, in 2009, Jiang et al. [20] proposed an idea of a robust signature scheme in V2I communication called binary authentication tree (BAT). The scheme efficiently diminishes the bottleneck issue in batch verification performance and so significantly reduced computational overhead. In BAT, the RSUs can quickly distinguish bogus messages from all the authentic ones, allowing them to withstand message flooding attacks to a great extent. However, in 2012, Wang et al. [21] discovered that Jiang et al.'s BAT cannot resist the forgery attack. They launch two types of attacks on any message, in which the adversary can counterfeit the batch verification and the signatures of the other vehicles. In the first case, any signer can remove any other user's components from the batch verification process. In 2013, Shim [35] also shows that Jiang et al.'s BAT scheme is insecure against forgery attacks, replay attacks, and Sybil attacks. All of the related works are shown in Table 1.

III. PRELIMINARIES

In this section, we introduce the system design, security and privacy requirements, the concept of a bilinear mapping operation, and a brief explanation about reputation management.

A. SYSTEM DESIGN

The two-layer concept in VANETs, with TA on the top, while RSUs and OBUs on the lower layer, have been introduced by Zhang et al. [10]. The task and function of each entity have been briefly described in Section I. Referring to [15], in our VANETs ecosystem, we assume:

1) TA is uncompromised;
2) Only TA that can reveal the real identity of the other entity;
3) TA - RSU communicate through a secured wireline networks;

Vehicle and RSU registration - vehicle signing

TA	$Vehicle$	RSU
	Vehicle registration:	
	Provides personal information PI.	
	← Sends PI	
Assigns n-dimensional column vectors of a_i and b_i.		
Computes $c_i = sH(a_i \oplus RID_i)$.		
Computes $VID_i = a_i \oplus RID_i$.		
Computes $REG_V = RID_i \parallel a_i \parallel b_i \parallel c_i$.		
Sends REG_V →		
	Receives REG_V.	
Computes $REG_{RSU} = VID_i \parallel b_i$.		
	Sends REG_{RSU} →	
		Receives REG_{RSU}.
	Vehicle signing:	
	Picks $r_i \in Z_q^*$ and generates:	
	$PID_i = (PID_{i,1}, PID_{i,2})$;	
	$PID_{i,1} = r_i P$	
	$PID_{i,2} = a_i \oplus RID_i \oplus H(b_i PID_{i,1})$.	
	Computes $\sigma_i = c_i + b_i c_i h(M_i)$, where $M_i = PID_i \parallel T_i$.	
	Computes $X_i = ENC_{PK_{RSU}}(r_i \parallel PID_i \parallel \sigma_i \parallel T_i)$.	
	Sends X_i →	
		Receives X_i.

FIGURE 2. ***Vehicle and RSU registration - vehicle signing* phases.**

4) RSU are semi-trusted (trusted but curious, it may reveal the privacy of the vehicle);
5) TPD is assumed to be credible.

B. SECURITY AND PRIVACY

The following are the description of security and privacy requirements that must hold in VANETs [14], [15], [22].

1) MESSAGE AUTHENTICATION

The implementation of the message authentication method is intended to allow the vehicle or RSU, to differentiate the original message from the bogus message. Furthermore, message authentication is also applied to resist modification and impersonation attacks.

2) NON-REPUDIATION

This requirement will give the message receiver a guarantee about the integrity and authenticity of the information they receive. The sender of the message cannot deny the information they have sent.

3) IDENTITY PRIVACY-PRESERVING

A sender of a message should be anonymous within a set of potential senders. As the user's real identity will be converted to an anonymous identity through TPD assistance. Therefore, without knowing the private master key of the TPD, an adversary cannot reveal the legitimate user's real identity. However, to reach accountability, only conditional anonymity is possible in VANETs, which is also related to traceability.

4) TRACEABILITY

The trusted authority (TA) should be able to reveal the real identities of the anonymous identities of the user in the case of a dispute. Traceability is also called conditional anonymity.

5) REPLAYING ATTACK RESISTANCE

The networks could endure a passive data capture and subsequent retransmission to produce an unauthorized message by the adversaries.

RSU verification - group key generation

$Vehicle$	RSU
Sends X_i →	
	RSU verification: Receives and decrypts X_i, then checks T_i. Single verification: $\hat{e}(\sigma_i, P) = \hat{e}(H(VID_i)(1 + b_i h(M_i)), P_{pub})$ Batch verification: $\hat{e}\left(\sum_{i=1}^{n} v_i \sigma_i, P\right) = \hat{e}\left(\sum_{i=1}^{n} v_i H(VID_i)(1 + b_i h(M_i)), P_{pub}\right)$ **Group key generation:** Selects a random nonce $d_{RSU} \in Z_q^*$ Computes $D_i = d_{RSU} PID_{i,1}$. Computes $D_G = \sum_{i=1}^{n} D_i$. Computes $K_{RSU} = \hat{e}\left(D_G, d_{RSU} P\right)$ Computes $\sigma_{RSU} = SK_{RSU} H(D)$, where $D = D_G \parallel D_1 \parallel D_2 \parallel \cdots \parallel D_n$. Broadcasts $Z = \sigma_{RSU} \parallel D$ to vehicles.
	← Sends Z
Verifies whether $\hat{e}(\sigma_{RSU}, P) = \hat{e}(H(D), PK_{RSU})$. If valid, computes $K_i = \hat{e}\left(D_G, r_i^{-1} D_i\right)$.	

FIGURE 3. ***RSU verification - group key generation*** **phases.**

6) UNLINKABILITY

An adversary vehicle (or RSU) should not link two or more subsequent pseudonym messages of the same vehicle.

C. BILINEAR MAP

The bilinear map $\hat{e}$ could be obtained from the modified Weil [23] or Tate pairings [24] on elliptic curves. Its security and complexity lie in the computational Diffie-Hellman problem (CDHP), which is believed to be hard to solve [25]. Let G_1 be a cyclic additive group generated by P, and G_2 is a cyclic multiplicative group with the same prime order q. Let $\hat{e} : G_1 \times G_1 \rightarrow G_2$ be a bilinear map if it satisfies the following properties:

1) Bilinear: For all $P, Q, R \in G_1$, we have $\hat{e}(Q, P + R) = \hat{e}(P, Q + R) = \hat{e}(Q, P) \cdot \hat{e}(Q, R)$. For any $a, b \in Z_q^*$, $\hat{e}(aQ, bP) = \hat{e}(bQ, aP) = \hat{e}(Q, P)^{ab}$.
2) Non-degenerate: $\hat{e}(P, Q) \neq 1$.
3) Computable: For any $P, Q \in G_1$, there is an efficient algorithm to compute $\hat{e}(P, Q)$.

As G_1 is a cyclic additive group generated by P, given $P, aP, bP \in G_1$, and $a, b \in Z_q^*$ are unknown values. The CDHP is hard, because there is no polynomial time algorithm that can discover $abP \in G_1$.

D. REPUTATION MECHANISM

In this paper, we are applying a reputation scoring mechanism for minimizing the computation cost of the BT-based verification scheme. In general, reputation management schemes are used for building trust among entities in VANETs. Based on the reputation values, vehicles may pick trustworthy messages sent by others that are intended for themselves.

In general, the trust models in VANETs can be classified into three categories: (i) entity-centric, (ii) data-centric, and (iii) the combined trust models [26]. Briefly described, entity-centric and data-centric trust management is focused on evaluating the trustworthiness of the vehicles and the received data, respectively. Meanwhile, the combined trust model integrates the entity-centric and data-centric mechanisms to establish trust in VANETs. In this work, we concentrate on the improved entity-based trust management method to aim for faster computation. It would be easier to arrange the signatures sequentially from the highest reputable vehicle to the lowest one by sorting all signature value coming to the batch.

To emphasize our point about reputation management's role in this work, we make assumptions about real-world applications. The first assumption is in VANETs majority of the vehicles are considered honest. So, in the following section, we will work with a small amount number of forged signatures. Second, we argue that vehicles with low-reputation scores tend to be more malicious than the high-reputation ones. Therefore, to increase the efficiency of finding illegal signatures in the batch, the BT-based scheme is used to maximize the opportunity for having the best scenario more often. A detailed explanation of the implemented reputation management system will be discussed in Section VI.

IV. BATCH VERIFICATION FOR TRAFFIC INFORMATION

As mentioned in Section I, our scheme is built based on Liu et al.'s [12] SEGKA scheme. By modifying its *vehicle signing*, *RSU verification*, *group key generation*, *group member joining*, and *group member leaving* phases, we made our improvement. Still adapting the full seven phases of the SEGKA, our proposed scheme consists of: *parameter initialization*, *vehicle and RSU registration*, *vehicle signing*, *RSU verification*, *group key generation*, *group member joining*, and *group member leaving* phases. To comprehend the scheme's procedure, notations throughout this paper are presented in Table 2.

A. PARAMETER INITIALIZATION

In this early phase, TA generates initial system parameters *params* for vehicles and RSU. First, it selects a cyclic additive group G_1 generated by P, and a cyclic multiplicative group G_2 with the same prime order q, to construct a bilinear map $\hat{e} : G_1 \times G_1 \rightarrow G_2$. Then, TA selects a secret parameter $s \in Z_q^*$ as its master key and computes $P_{pub} = sP$ as its public key. TA selects a map-to-point hash function $H(\cdot) : \{0,1\}^* \rightarrow G_1$ and a one-way hash function $h(\cdot) : \{0,1\}^* \rightarrow Z_q^*$. Finally, TA broadcasts $params = \{G_1, G_2, \hat{e}, q, P, P_{pub}, H(\cdot), h(\cdot)\}$ to vehicles and RSU in the network.

B. VEHICLE AND RSU REGISTRATION

Vehicle owners will directly go to the TA during the (offline) registration process. They must provide information such as name, address, email address, phone number, *etc.* to the TA. Then, TA registers both vehicles V_i and RSU for being able to communicate in VANETs. The a_i and b_i denote a shared secret key of TA - V_i and a shared secret key of V_i - RSU, respectively. TA computes $c_i = sH(a_i \oplus RID_i)$ and sends $REG_V = RID_i \parallel a_i \parallel b_i \parallel c_i$ to V_i. Finally, TA computes V_i's verification $VID_i = a_i \oplus RID_i$ and sends $REG_{RSU} = VID_i \parallel b_i$ to RSU. The process of this phase is shown in Figure 2.

C. VEHICLE SIGNING

In this phase, V_i selects a random nonce $r_i \in Z_q^*$ to generates its pseudo-identity $PID_i = (PID_{i,1}, PID_{i,2})$, where $PID_{i,1} = r_iP$ and $PID_{i,2} = a_i \oplus TID_i \oplus H(b_iPID_{i,1})$. Then, V_i computes its signature $\sigma_i = c_i + b_ic_ih(M_i)$, where $M_i = PID_i \parallel T_i$, and T_i is the signing time. Finally, V_i sends $X_i = ENC_{PK_{RSU}}(r_i \parallel PID_i \parallel \sigma_i \parallel T_i)$ to RSU, with $PK_{RSU} = SK_{RSU}P$ is the public key of RSU. The diference towards [12], they do not encrypt $(r_i \parallel PID_i \parallel \sigma_i \parallel T_i)$. The process of this phase is shown in Figure 2.

D. RSU VERIFICATION

Upon receiving X_i from V_i, RSU decrypts X_i using its secret key $DEC_{SK_{RSU}}(ENC_{PK_{RSU}}(r_i \parallel PID_i \parallel \sigma_i \parallel T_i))$ and checks the freshness of T_i. In the single verification mode, RSU verifies σ_i by checking whether (1) holds or not.

$$\begin{aligned}
&\hat{e}(\sigma_i, P)\\
&\quad = \hat{e}(c_i + b_ic_ih(M_i), P)\\
&\quad = \hat{e}(c_i, P) \cdot \hat{e}(b_ic_ih(M_i), P)\\
&\quad = \hat{e}(sH(a_i \oplus RID_i), P) \cdot \hat{e}(b_isH(a_i \oplus RID_i)h(M_i), P)\\
&\quad = \hat{e}(H(VID_i), sP) \cdot \hat{e}(b_iH(VID_i)h(M_i), sP)\\
&\quad = \hat{e}(H(VID_i), P_{pub}) \cdot \hat{e}(b_iH(VID_i)h(M_i), P_{pub})\\
&\quad = \hat{e}(H(VID_i)(1 + b_ih(M_i)), P_{pub}) \qquad (1)
\end{aligned}$$

Meanwhile, in the batch verification mode, RSU verifies σ_i by checking whether (2) holds or not.

$$\begin{aligned}
&\hat{e}\left(\sum_{i=1}^{n} v_i\sigma_i, P\right)\\
&\quad = \hat{e}\left(\sum_{i=1}^{n} v_i(c_i + b_ic_ih(M_i)), P\right)\\
&\quad = \hat{e}\left(\sum_{i=1}^{n} v_ic_i, P\right) \cdot \hat{e}\left(\sum_{i=1}^{n} v_ib_ic_ih(M_i), P\right)\\
&\quad = \hat{e}\left(\sum_{i=1}^{n} v_isH(a_i \oplus RID_i), P\right)\\
&\qquad \cdot \hat{e}\left(\sum_{i=1}^{n} v_ib_isH(a_i \oplus RID_i)h(M_i), P\right)\\
&\quad = \hat{e}\left(\sum_{i=1}^{n} v_iH(VID_i), sP\right)\\
&\qquad \cdot \hat{e}\left(\sum_{i=1}^{n} v_ib_iH(VID_i)h(M_i), sP\right)\\
&\quad = \hat{e}\left(\sum_{i=1}^{n} v_iH(VID_i), P_{pub}\right)\\
&\qquad \cdot \hat{e}\left(\sum_{i=1}^{n} v_ib_iH(VID_i)h(M_i), P_{pub}\right)\\
&\quad = \hat{e}\left(\sum_{i=1}^{n} v_iH(VID_i)(1 + b_ih(M_i)), P_{pub}\right) \qquad (2)
\end{aligned}$$

When both of (1) and (2) are hold, so the vehicles are authenticated. The process of this phase is shown in Figure 3.

E. GROUP KEY GENERATION

After σ_i is authenticated, the RSU will generate the group key for vehicles in its area. RSU selects a random nonce $d_{RSU} \in Z_q^*$, and computes $D_i = d_{RSU}PID_{i,1}$ and $K_{RSU} = \hat{e}(D_G, d_{RSU}P)$, with $D_G = \sum_{i=1}^{n} D_i$. In this phase, our modification towards the SEGKA, D_G is computed in the RSU rather than in V_i. Then, RSU computes its signature

$\sigma_{RSU} = SK_{RSU}H(D)$, where $D = D_G \parallel D_1 \parallel D_2 \parallel \cdots \parallel D_n$, and broadcasts $Z = \sigma_{RSU} \parallel D$ to vehicles in its area. After receiving Z, V_i verifies σ_{RSU} by checking whether $\hat{e}(\sigma_{RSU}, P) = \hat{e}(H(D), PK_{RSU})$ holds or not. If yes, V_i computes the group key $K_i = \hat{e}\left(D_G, r_i^{-1}D_i\right)$. The process of this phase is shown in Figure 3.

F. GROUP MEMBER JOINING

When a new vehicle V_a joins the network, it will selects a random nonce $r_a \in Z_q^*$ to generates its pseudo-identity $PID_a = (PID_{a,1}, PID_{a,2})$, where $PID_{a,1} = r_aP$ and $PID_{a,2} = a_a \oplus RID_a \oplus H(b_aPID_{a,1})$. Then, V_a calculates its signature $\sigma_a = c_a + b_ac_ah(M_a)$, where $M_a = PID_a \parallel T_a$, and sends $X_a = ENC_{PK_{RSU}}(r_a \parallel PID_a \parallel \sigma_a \parallel T_a)$ to RSU. After receiving X_a, RSU decrypts it using its secret key $DEC_{SK_{RSU}}(ENC_{PK_{RSU}}(r_a \parallel PID_a \parallel \sigma_a \parallel T_a))$ and check the freshness of T_a. The RSU verifies whether $PID_{a,2} = VID_a \oplus H(b_aPID_{a,1})$. If holds, RSU verifies σ_a by checking whether $\hat{e}(\sigma_a, P) = \hat{e}(H(VID_a)(1 + b_ah(M_a)), P_{pub})$ holds or not. If holds, RSU allows V_a for joining the network. When V_a joins the network, RSU will update the group key by selects a random nonce $d'_{RSU} \in Z_q^*$, recomputes $D'_i = d'_{RSU}PID_{i,1}$, with $(1 \leq i \leq n)$ and $D_a = d'_{RSU}PID_{a,1}$. Then, RSU computes $K'_{RSU} = \hat{e}\left(D'_G, d'_{RSU}P\right)$, with $D'_G = \sum_{i=1}^{n} D'_i + D_a$, and its new signature $\sigma'_{RSU} = SK_{RSU}H(D')$, where $D' = D'_G \parallel D'_1 \parallel D'_2 \parallel \cdots \parallel D'_n \parallel D_a$. RSU broadcasts $Z' = \sigma'_{RSU} \parallel D'$ to the new group of vehicles. Upon receiving Z', vehicles will check whether $\hat{e}(\sigma'_{RSU}, P) = \hat{e}(H(D'), PK_{RSU})$ holds or not. If holds, compute the new group key $K'_i = \hat{e}\left(D'_G, r_i^{-1}D'_i\right)$.

G. GROUP MEMBER LEAVING

When V_i leaves the network, RSU updates K_i for the remaining $n-1$ vehicles. RSU selects $d'_{RSU} \in Z_q^*$ and computes $D'_i = d'_{RSU}PID_{i,1}; (1 \leq i \leq n-1)$. Then, RSU computes $K'_{RSU} = \hat{e}\left(D'_G, d'_{RSU}P\right)$, with $D'_G = \sum_{i=1}^{n-1} D'_i$, and its new signature $\sigma'_{RSU} = SK_{RSU}H(D')$, where $D' = D'_G \parallel D'_1 \parallel D'_2 \parallel \cdots \parallel D'_{n-1}$. RSU broadcasts $Z' = \sigma'_{RSU} \parallel D'$ to the remaining vehicles. Upon receiving Z', vehicles will check whether $\hat{e}(\sigma_{RSU}, P) = \hat{e}(H(D), PK_{RSU})$ holds or not. If holds, compute the new group key $K'_i = \hat{e}\left(D'_G, r_i^{-1}D'_i\right)$.

V. ILLEGAL SIGNATURES IDENTIFICATION WITH BT-BASED BATCH VERIFICATION SCHEME

In 2013, Atanasiu [27] proposed a BT-based batch verification scheme for identifying illegal signatures. When the verifier receives the messages $\langle M_1, \sigma_1\rangle$, $\langle M_2, \sigma_2\rangle$, $\cdots$, $\langle M_n, \sigma_n\rangle$ from the signer, the verifier will re-order these signatures by a total order relation and perform the following procedures to verify the illegal signature. The representative approach of Atanasiu's work is presented based on work in [13].

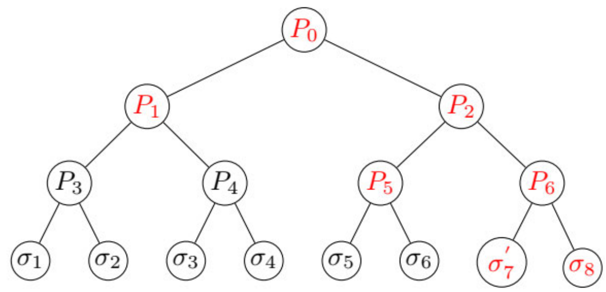


FIGURE 4. An example of an illegal signature σ'_7.

A. PRINCIPAL OF THE BT-BASED BATCH VERIFICATION SCHEME

For example, there are eight signatures in the batch, $\langle r_1, PID_1, \sigma_1, T_1\rangle$, $\langle r_2, PID_2, \sigma_2, T_2\rangle$, $\cdots$,$\langle r_8, PID_8, \sigma_8, T_8\rangle$ that come to the RSU. RSU will re-orders these signatures by a total order relation: $\langle r_1, PID_1, \sigma_1, T_1\rangle < \langle r_2, PID_2, \sigma_2, T_2\rangle < \cdots < \langle r_8, PID_8, \sigma_8, T_8\rangle$.

Assume there is one illegal signature σ'_7 appears in the batch (see Figure 4). The verifier performs one-time batch verification with all eight signatures in (3).

$$\hat{e}\left(\sum_{i=1}^{8} v_i\sigma_i, P\right) \stackrel{?}{=} \hat{e}\left(\sum_{i=1}^{8} v_iH(VID_i)(1 + b_ih(M_i)), P_{pub}\right) \tag{3}$$

Since there is one illegal signature σ'_7 in the batch, so (3) is not holds. The verifier divides these eight signatures into two parts: part 1 (left-side of the tree in Figure 4): $[\langle r_1, PID_1, \sigma_1, T_1\rangle, \langle r_2, PID_2, \sigma_2, T_2\rangle, \cdots, \langle r_4, PID_4, \sigma_4, T_4\rangle]$, and part 2 (right-side of the tree in Figure 4): $[\langle r_5, PID_5, \sigma_5, T_5\rangle, \langle r_6, PID_6, \sigma_6, T_6\rangle, \cdots, \langle r_8, PID_8, \sigma_8, T_8\rangle]$. The verifier performs one-time batch verification with all signatures in part 1 and part 2 (see (4) and (5)), respectively. Since there are no illegal signatures in part 1, so (4) holds. Meanwhile, because the illegal signature σ'_7 is located in part 2, so (5) is not holds. Then, the verifier divides those four signatures in part 2 into two sub-parts: part 3: $[\langle r_5, PID_5, \sigma_5, T_5\rangle, \langle r_6, PID_6, \sigma_6, T_6\rangle]$, and part 4: $[\langle r_7, PID_7, \sigma_7, T_7\rangle, \langle r_8, PID_8, \sigma_8, T_8\rangle]$.

$$\hat{e}\left(\sum_{i=1}^{4} v_i\sigma_i, P\right) \stackrel{?}{=} \hat{e}\left(\sum_{i=1}^{4} v_iH(VID_i)(1 + b_ih(M_i)), P_{pub}\right) \tag{4}$$

$$\hat{e}\left(\sum_{i=5}^{8} v_i\sigma_i, P\right) \stackrel{?}{=} \hat{e}\left(\sum_{i=5}^{8} v_iH(VID_i)(1 + b_ih(M_i)), P_{pub}\right) \tag{5}$$

The iteration of these steps will continue until the illegal signatures σ'_7 is detected. Once the verifier performs one-time batch verification with a signature in (6), it found one illegal signature σ'_7, therefore (6) is not holds, and the verifier immediately knows if $\langle r'_7, PID'_7, \sigma'_7, T'_7\rangle$ is illegal.

$$\hat{e}(\sigma_7, P) = \hat{e}(H(VID_7)(1 + b_7h(M_7)), P_{pub}) \tag{6}$$

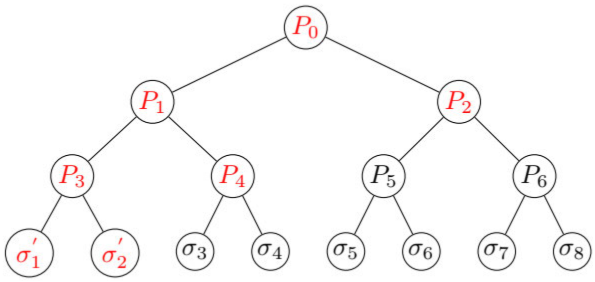


FIGURE 5. **The number of calculations for the best-case scenario with two illegal signatures σ_1' and σ_2'.**

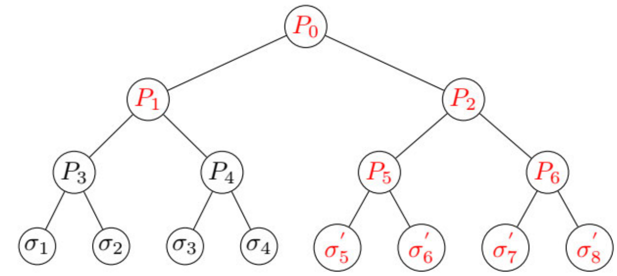


FIGURE 6. **The number of calculations for the best-case scenario with four illegal signatures σ_5', σ_6', σ_7', and σ_8'.**

From the above operation (see the red procedure in Figure 4 for $\{P_0, P_1, P_2, P_5, P_6, \sigma_7', \sigma_8\}$), we can see if a BT-based illegal signature identifying scheme can easily be applied to the batch system.

B. ANALYSIS OF BT-BASED ILLEGAL SIGNATURES IDENTIFICATION MECHANISM

In this subsection, we analyze the effectiveness of the BT-based batch verification method in verifying illegal signatures. We divide the discussion into two scenarios, the best-case and the worst-case. In the best-case scenario, all illegal signatures' locations are located consecutively in the same tree. Figure 5 and Figure 6 are two examples of the number of calculations in the best-case scenario with two and four illegal signatures, respectively.

On the other hand, the worst-case scenario is that all illegal signatures' locations are in different trees and scattered everywhere. Figure 7 and Figure 8 are two examples of the number of calculations in the worst-case scenario with two and four illegal signatures, respectively.

1) THE BEST-CASE SCENARIO

If there are b illegal signatures in the n messages, the number of calculations T_{best} in the best-case scenario can be determined using (7)

$$T_{best} \leq 2[\lceil \lg n \rceil - \lg(\lceil \frac{b}{2} \rceil 2)] + 2^{\lceil \lg(\lceil \frac{b}{2} \rceil 2) \rceil + 1} - 1 \qquad (7)$$

Since we are using a ceiling function, the number of calculation T_{best} for one and two illegal signatures are the same. In the best scenario, if we have two illegal signatures (σ_1' and σ_2') in the eight messages, the number of calculations T_{best} is seven exponential operations (see the numbers of red operation $\{P_0, P_1, P_2, P_3, P_4, \sigma_1', \sigma_2'\}$ in Figure 5 and (8)):

$$\begin{aligned} T_{best} &\leq 2[\lceil \lg 8 \rceil - \lg(\lceil \frac{2}{2} \rceil 2)] + 2^{\lceil \lg(\lceil \frac{2}{2} \rceil 2) \rceil + 1} - 1 \\ &\leq 2(3-1) + 2^2 - 1 \\ &\leq 7 \end{aligned} \qquad (8)$$

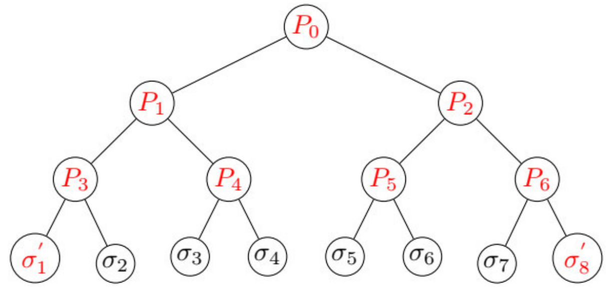


FIGURE 7. **The number of calculations for the worst-case scenario with two illegal signatures σ_1' and σ_8'.**

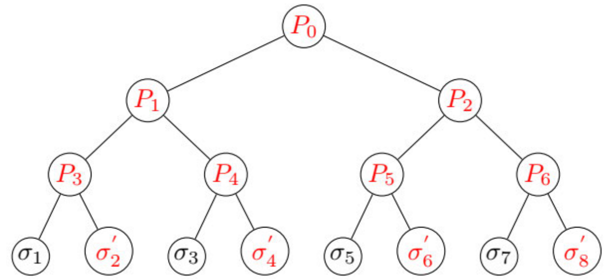


FIGURE 8. **The number of calculations for the worst-case scenario with four illegal signatures σ_2', σ_4', σ_6', and σ_8'.**

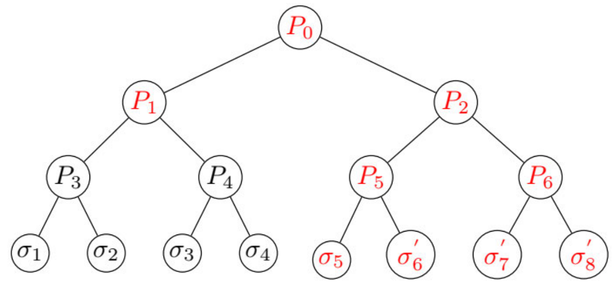


FIGURE 9. **The number of calculations for the best-case scenario with three illegal signatures σ_6', σ_7', and σ_8'.**

If there are three illegal signatures (σ_6', σ_7' and σ_8') in the eight messages (see Figure 9 and (9)), the number of calculations T_{best} is nine exponential operations $\{P_0, P_1, P_2, P_5, P_6, \sigma_5, \sigma_6', \sigma_7', \sigma_8'\}$. Those number of calculations is the same as if we have four illegal signatures in eight messages as seen in Figure 6. We still have to compute σ_5 even though it is not illegal.

$$\begin{aligned} T_{best} &\leq 2[\lceil \lg 8 \rceil - \lg(\lceil \frac{3}{2} \rceil 2)] + 2^{\lceil \lg(\lceil \frac{3}{2} \rceil 2) \rceil + 1} - 1 \\ &\leq 2(3-2) + 2^3 - 1 \\ &\leq 9 \end{aligned} \qquad (9)$$

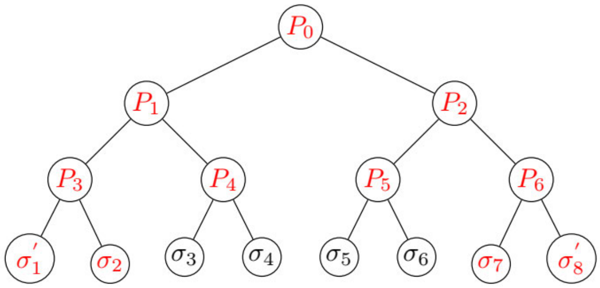


FIGURE 10. The number of calculations for the worst-case scenario with two illegal signatures σ_1' and σ_8'.

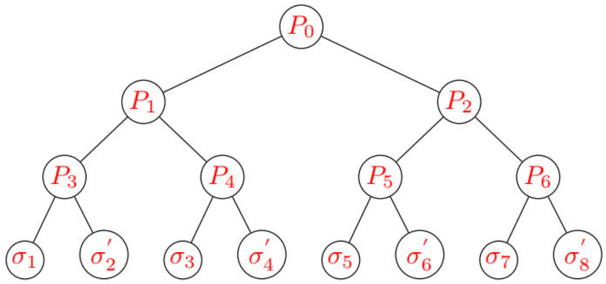


FIGURE 11. The number of calculations for the worst-case scenario with four illegal signatures σ_2', σ_4', σ_6', and σ_8' in the eight messages.

2) THE WORST-CASE SCENARIO

If there are b illegal signatures in the n messages, the number of calculations T_{worst} in the worst-case is shown in (10).

$$T_{worst} \leq (2^{\lceil \lg b \rceil} - 1) + b[1 + 2(\lceil \lg n \rceil - \lceil \lg b \rceil)] \quad (10)$$

So, let two illegal signatures (σ_1' and σ_8') appear in the eight messages as depicted in Figure 7, the number of calculations T_{worst} is 11 exponential operations (see Figure 10). Even though it is just σ_1' and σ_8' that being illegal, (11) still need to compute σ_2 and σ_7, because they are located in the same tree.

$$\begin{aligned} T_{worst} &\leq (2^{\lceil \lg 2 \rceil} - 1) + 2[1 + 2(\lceil \lg 8 \rceil - \lceil \lg 2 \rceil)] \\ &\leq (2^1 - 1) + 2(1 + 4) \\ &\leq 11 \end{aligned} \quad (11)$$

So, if we have four illegal signatures σ_2', σ_4', σ_6', and σ_8' that located in the different tree, the number of calculations become 15 exponential operations (see Figure 11).

$$\begin{aligned} T_{worst} &\leq (2^{\lceil \lg 4 \rceil} - 1) + 4[1 + 2(\lceil \lg 8 \rceil - \lceil \lg 4 \rceil)] \\ &\leq (2^2 - 1) + 4(1 + 2) \\ &\leq 15 \end{aligned} \quad (12)$$

VI. IMPROVING THE EFFICIENCY OF BT-BASED BATCH VERIFICATION SCHEME

As discussed in Section V, to identify the illegal signatures that could appear in the batch, we have applied a BT-based scheme to address the forged signature's location. However, by such implementation, we still have a probability of having a worst-case scenario, in which the forged signatures could be scattered in the tree. By those conditions, we will suffer from a high computational cost.

To improve efficiency, we implement a reputation scoring mechanism for every vehicle in the network. The reputation algorithm used in this work aims to arrange all vehicles' reputation value in the table. By giving every vehicle a reputation score, we can arrange the signatures sequentially from the highest-reputable vehicle to the lowest. Therefore, with avowed assumptions in Section III.D, we try to make the probability of the best scenario appearing in the batch as frequent as possible. To implement those scenarios, we have to ensure the signatures from the low-reputation vehicles are arranged in the same branch of the tree. A message will be considered a trusted one if transmitted by a high-reputation vehicle and vice versa.

In [28], Hussain et al. proposed a hybrid (combined) trust model for vehicular social networks. To calculate trust, each node j calculates the trust value for its neighbor i based on two factors: a direct encounter between i and j, and endorsement by i's neighbors of message broadcasted by i. Relatively similar with [28], Dong et al. [29] also propose a reputation management scheme that involves the neighbors as the whole determinant of its scoring system. However, not like [28], Dong et al. propose their idea to work in a blockchain environment.

Meanwhile, a recent study in the data-centric trust model was proposed by Su et al. [30]. They offer a centralized reputation mechanism for detecting malicious information dissemination among vehicles in 5G networks. It will decide whether to trust a received message or not according to the reputation value of the sender. Meanwhile, the validation process of the collected information would be conducted later.

From all of those mentioned schemes [28], [29], [30], they have a similarity in how they use neighbor's validation and their trust value as part of the assessments. By slightly modifying their idea, we consider the neighboring vehicles as the partial contributor to every user's reputation value. We consider the current reputation value $rep_i^{(t)}$ is a mixed between vehicle V_i's previous reputation score $rep_i^{(t-1)}$ and the current neighbor's validation value. The scoring mechanism is done by fellow vehicles in a peer-to-peer manner, even though our authentication scheme is V2I-based.

$$rep_i^{(t)} = \frac{1}{2}\left[rep_i^{(t-1)} + \left(\frac{\sum_{j=1}^{n} p_{ij}^{(t)} rep_j^{(t)}}{\sum_{j=1}^{n} rep_j^{(t)}} \right) \right], \ with\ i \neq j \quad (13)$$

In above equation, $rep_i^{(t)}$ refers to vehicle's V_i reputation at time t, while $rep_i^{(t-1)}$ is V_i's previous reputation at time $t-1$. On the neighbor's side, $p_{ij}^{(t)}$ refers to current validation results of a message sent by V_i to V_j, at time t. Meanwhile, $rep_j^{(t)}$ refers to reputation of vehicle V_j (V2V) at time t. Since we consider a V2I communication in our approach, $p_{ij}^{(t)}$ refers to the current validation results of a message sent by V_i to

TABLE 3. Five-star reputation rating.

Rating	Value
⋆⋆⋆⋆⋆	1
⋆⋆⋆⋆	0.8
⋆⋆⋆	0.6
⋆⋆	0.4
⋆	0.2

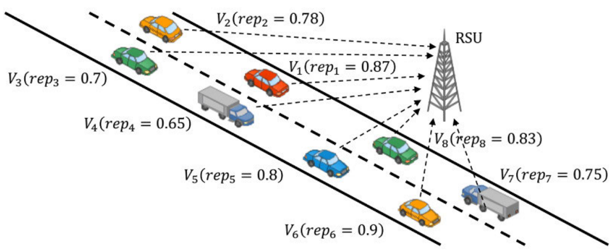


FIGURE 12. The vehicle-RSU interaction.

RSU. The operation of $\sum_{j=1}^{n} p_{ij}^{(t)} rep_j^{(t)}$ refers to accumulation of validation from all vehicles toward V_i, with n represents the total number of vehicles in the system.

By calculating (13), we can see if the current reputation $rep_i^{(t)}$ is composed by the average value of the V_i's previous reputation value $rep_i^{(t-1)}$, and added up by neighboring vehicle's current validations value $\sum_{j=1}^{n} p_{ij}^{(t)} rep_j^{(t)}$. However, by considering real-world applications, the majority of vehicles are honest; we assume if the assessment that comes from neighboring vehicles is fair.

To make a substantive approach towards how the neighbor vehicles V_j validate the V_i, we use a five-star rating concept as the assessment method. This common practice will let users quickly rate other vehicles' information based on their real perception. The five-star reputation rating and its value are represented in Table 3.

Suppose we are given eight vehicles in the networks $\{V_1, V_2, V_3, V_4, V_5, V_6, V_7, V_8\}$, with RSU receiving messages from the entire neighborhood (see Figure 12). Every reputation score of each vehicle presented in Figure 12 is stated in time $t-1$.

To simplify the implementation of our reputation value in this context, we are setting several assumptions. First, we assume if every vehicle broadcasts the same accurate information that is equally correct to RSU. Second, every vehicle will give the same valuation $p_{ij}^{(t)}$ to any particular vehicle. Hence, by implementing (7), we now have,

$$\begin{aligned} rep_1^{(t)} &= \frac{1}{2}[0.87 + ((0.8 \times 0.78)+(0.8\times 0.7)+(0.8\times 0.65) \\ &\quad +(1 \times 0.8) + (1 \times 0.9) + (0.8 \times 0.75) \\ &\quad +(1 \times 0.83) + (1 \times 0.85))/6.26] \\ &= \frac{1}{2}\left[0.87 + \left(\frac{5.684}{6.26}\right)\right] \\ &= 0.889 \\ &\vdots \end{aligned}$$

TABLE 4. The current reputation score at time t.

Vehicle	$rep_i^{(t-1)}$	$rep_i^{(t)}$	Sorted vehicle	Sorted $rep_i^{(t)}$
V_1 (σ_1)	0.87	0.889	V_6 (σ_6)	0.9038
V_2 (σ_2)	0.78	0.8569	V_1 (σ_1)	0.889
V_3 (σ_3)	0.7	0.8161	V_8 (σ_8)	0.8693
V_4 (σ_4)	0.65	0.7906	V_2 (σ_2)	0.8569
V_5 (σ_5)	0.8	0.8545	V_5 (σ_5)	0.8545
V_6 (σ_6)	0.9	0.9038	V_7 (σ_7)	0.8416
V_7 (σ_7)	0.75	0.8416	V_3 (σ_3)	0.8161
V_8 (σ_8)	0.83	0.8693	V_4 (σ_4)	0.7906

$$\begin{aligned} rep_8^{(t)} &= \frac{1}{2}[0.83 + ((1 \times 0.87)+(0.8 \times 0.78) + (0.8 \times 0.7) \\ &\quad +(0.8 \times 0.65) + (1 \times 0.8) + (1 \times 0.9) \\ &\quad +(0.8 \times 0.75) + (1 \times 0.85))/6.3] \\ &= \frac{1}{2}\left[0.83 + \left(\frac{5.724}{6.3}\right)\right] \\ &= 0.8693 \end{aligned}$$

After getting all updated reputation values ($rep_1^{(t)}$ to $rep_8^{(t)}$) from each vehicle in the network, vehicle RSU as the receiver can sort each sender's reputation score from the highest to the lowest (see Table 4). Each reputation value $rep_i^{(t)}$ represents its corresponding signature σ_i. By using a common sort tree algorithm, we can arrange the signature from the highest reputation value or vice versa to maximize the best-case scenario probability.

VII. SECURITY AND PERFORMANCE ANALYSIS

In this section, we analyze the security and performance of the proposed scheme, which includes non-repudiation, identity privacy-preserving, message authentication, traceability, resistance to replay attacks, unlinkability, backward secrecy, and forward secrecy, as follows.

A. SECURITY ANALYSIS

1) MESSAGE AUTHENTICATION

Message authentication is the most fundamental security requirement to confirm the legitimacy of a message's source and its integrity in any communication [16]. Our proposed scheme employs a one-way hash function $h(\cdot)$ to protect message M_i in signature σ_i. Without knowing the shared secret value of a_i and b_i, that lead to c_i, it is inaccessible to forge a valid σ_i. Moreover, since we believe that the CDHP in G_1 is hard to solve, it is difficult to derive the c_i from s, a_i, and RID_i. Therefore, M_i that is sealed by $h(\cdot)$ is unforgeable, and the message authentication requirement is achieved.

2) NON-REPUDIATION

The vector v_i is used to avoid user swap of the M_i and σ_i [16]. If the adversary $\mathcal{A}$ wants to deny the signatures by swapping M_i and σ_i, his/her signatures will result in the batch message verification failing. We perform the small exponent test that previously conducted in [31] and [32]. Givenly P is a generator in G_1, we have $(\sigma_1, y_1), (\sigma_2, y_2), \cdots, (\sigma_n, y_n)$, with

$\sigma_i \in Z_p$ and $y_i \in G_1$, check if $\forall i \in \{1, 2, \cdots, n\} : \hat{e}(\sigma_i, P) = \hat{e}(y_i, Q)$, by doing the following steps:

- Selects random parameters $l_1, l_2, \cdots, l_n \in \{0, 1\}^l$
- Compute $A = \sum_{i=1}^{n} l_i y_i$ and $B = \sum_{i=1}^{n} l_i \sigma_i$
- If $\hat{e}(B, P) = \hat{e}(A, Q)$, then accept, otherwise reject.

The batch instance will be (σ_1, y_1), (σ_2, y_2),$\cdots$, (σ_n, y_n), with $y_i = (H(VID_i)(1 + b_i h(M_i)), P_{pub})$. The verification of the signature consists of checking operation that $\hat{e}(\sigma_i, P) = \hat{e}(y_i, Q)$. If $\mathcal{A}$ wants to make false multiple digital signatures σ_i valid, he/she must make those operation holds. Since $\mathcal{A}$ did not know the values of l that leads to the value of v_i, it is difficult for $\mathcal{A}$ to make $\hat{e}(\sigma_i, P) = \hat{e}(y_i, Q)$ holds.

3) IDENTITY PRIVACY-PRESERVING

To get a $PID_i = \{PID_{i,1}, PID_{i,2}\}$, user must input their RID and PWD, then verified by the TPD. Since $PID_{i,1} = r_i P$ and $PID_{i,2} = a_i \oplus RID_i \oplus H(b_i PID_{i,1})$, so $\mathcal{A}$ can try to retrieve RID_i by doing $RID_i = a_i \oplus VID_i = a_i \oplus PID_{i,2} \oplus H(b_i PID_{i,1})$. However, since we believe that computational Diffie-Hellman problem (CDHP) used in the bilinear pairing operation is hard, hence we argue that $\mathcal{A}$ cannot obtain any vehicle's V_i real identity RID_i easily [10], [33].

4) TRACEABILITY

Related to the previous elaboration where $RID_i = a_i \oplus VID_i = a_i \oplus PID_{i,2} \oplus H(b_i PID_{i,1})$, since only TA and the particular vehicle V_i who know the value of a_i, so in the case of dispute, TA can reveal the RID_i of all vehicles in the network.

5) RESISTANCE TO REPLAYING ATTACK

In the *vehicle signing* phase, we employ a timestamp T_i in $X_i = ENC_{PK_{RSU}}(r_i \parallel PID_i \parallel \sigma_i \parallel T_i)$ to ensure the freshness of the message. RSU will decrypt the message and receive the latest message from vehicles. Meanwhile, $\mathcal{A}$ cannot replay the message since it has been encrypted using RSU's public key, and only the RSU can decrypt it using its private key.

6) UNLINKABILITY

During the *vehicle signing* phase, a pseudo-identity $PID_i = \{PID_{i,1}, PID_{i,2}\}$ is utilized to generate the signature σ_i. To create $PID_{i,1} = r_i P$, we use a different random number $r_i \in Z_q^*$. Meanwhile, to generate $\sigma_i = c_i + b_i c_i h(M_i)$, we employ a timestamp T_i in $M_i = PID_i \parallel T_i$. Therefore, any $\mathcal{A}$ attempting to link two or more consecutive signatures may fail since the message's contents change each time the pseudo-identity and timestamp change.

7) BACKWARD SECRECY

Backward secrecy means any newly joining vehicles cannot obtain the previous group key, even if it has the current one. As a result, they are unable to read the group's previous conversations. When a new vehicle joining the network, RSU will generate a new random nonce $d'_{RSU} \in Z_q^*$, to compute $D'_i = d'_{RSU} PID_{i,1}$, $D_a = d'_{RSU} PID_{a,1}$, $D'_G = \sum_{i=1}^{n} D'_i + D_a$, and $\sigma'_{RSU} = SK_{RSU} H(D')$, where $D' = D'_G \parallel D'_1 \parallel D'_2 \parallel \cdots \parallel D'_n \parallel D_a$. RSU then broadcasts $Z' = \sigma'_{RSU} \parallel D'$ to vehicles in its area. After receiving Z' and validating σ'_{RSU}, all vehicles, including the new one, compute the new group key $K'_i = \hat{e}\left(D'_G, r_i^{-1} D'_i\right)$. Therefore, the newly joining vehicle don't have any opportunity to obtains the old group key K_i, and infiltrate any previous communication.

8) FORWARD SECRECY

Forward secrecy means any leaving vehicles cannot obtain the future group's key, even if it has the current one. As a result, they are unable to read the group's future conversations. When a vehicle leaving the network, RSU will generate a new random nonce $d'_{RSU} \in Z_q^*$, to compute $D'_i = d'_{RSU} PID_{i,1}$, $D'_G = \sum_{i=1}^{n} D'_i$, and $\sigma'_{RSU} = SK_{RSU} H(D')$, where $D' = D'_G \parallel D'_1 \parallel D'_2 \parallel \cdots \parallel D'_{n-1}$. RSU then broadcasts $Z' = \sigma'_{RSU} \parallel D'$ to vehicles in its area. After receiving Z' and validating σ'_{RSU}, all current vehicles compute the new group key $K'_i = \hat{e}\left(D'_G, r_i^{-1} D'_i\right)$. Therefore, the leaving vehicle don't have any opportunity to obtains the new group key K_i, and infiltrate any future communication.

B. PERFORMANCE ANALYSIS

This subsection mainly discusses the comparison of computation complexity between ours and the other related schemes, as presented in Table 5. Related to the rapid topology shift in VANETs, verification delay becomes the most critical process to address because it could affect information value.

Let PC is a pairing operation cost, SC is a scalar multiplication cost, HC is a map-to-point hash function cost, and EC is an exponentiation operation cost in G_1. We adopt an experiment in [34], which observes computation overhead in Python charm cryptographic library, on Intel Core i7-4765T 2.00 GHz and 8 GB RAM machine. The following results are obtained: PC is 1.34 ms, SC is 5.13 μs, HC is 0.0065 ms, EC is 2.03 ms. In Table 4, we only focus on comparing our scheme with the existing schemes proposed by Liu et al. [12], Tzeng et al. [15], Azees et al. [18], Gu et al. [19], Jiang et al. [20], Wang et al. [21], and Shim et al. [35], in batch signatures verification process, with and without $b \geq 1$ fake signatures.

In Table 5, we can see both of Liu et al.'s [12] and our improved scheme use the same constant $3PC + SC$ operation in the batch verification phase. In the n authentic signatures verification process, the number of pairing operation costs is stay constant for $3PC + SC$ (as well as Tzeng et al.'s [15] scheme for $2PC + SC$). Meanwhile, the computation cost of other schemes will linearly increase with the number of signatures. In Figure 13, we can see a substantial gap between Azees et al.'s [18] and Gu et al.'s [19] schemes, towards the other schemes. This happens because the pairing cost PC operation is affected by the increasing number of n received messages. Meanwhile, as seen in Figure 14, Tzeng et al.'s [15] scheme gives the best result in the n authentic signatures verification process among all compared schemes.

TABLE 5. Performance comparison of the batch signatures verification schemes.

Scheme	n authentic signatures	n signatures with $b \geq 1$ fake signatures
Liu *et al.* [12]	$3PC + SC$ = 4.02513 ms	-
Tzeng *et al.* [15]	$2PC + SC$ = 2.68513 ms	-
Azees *et al.* [18]	$(1+n)PC$ = 1.34(1+n) ms	-
Gu *et al.* [19]	$nPC + 3nEC$ = 7.43n ms	-
Jiang *et al.* [20]	$2PC + nSC$ = 2.68+0.00513n ms	$((b+1)\lg(n/b) + 4b - 2)PC + nSC$
Wang *et al.* [21]	$2PC + 3nSC$ = 2.68+0.01539n ms	$((b+1)\lg(n/b) + 4b - 2)PC + (2(n-1)b + n)SC$
Shim *et al.* [35]	$2PC + nSC$ = 2.68+0.00513n ms	-
Ours	$3PC + SC$ = 4.02513 ms	$(2[\lceil \lg n \rceil - \lg(\lceil \frac{b}{2} \rceil 2)] + 2^{\lceil \lg(\lceil \frac{b}{2} \rceil 2) \rceil + 1} - 1)PC + SC$

TABLE 6. Performance comparison with four fake signatures in 512 authentic ones.

Scheme	$n = 512$	$b = 4, n = 512$
Jiang *et al.* [20]	$2PC + nSC$ = 5.30656 ms	$((b+1)\lg(n/b) + 4b - 2)PC + nSC$ = 68.28656 ms
Wang *et al.* [21]	$2PC + 3nSC$ = 10.55968 ms	$((b+1)\lg(n/b) + 4b - 2)PC + (2(n-1)b + n)SC$ = 89.258 ms
Ours	$3PC + SC$ = 4.02513 ms	$(2[\lceil \lg n \rceil - \lg(\lceil \frac{b}{2} \rceil 2)] + 2^{\lceil \lg(\lceil \frac{b}{2} \rceil 2) \rceil + 1} - 1)PC + SC$ = 29.48513 ms

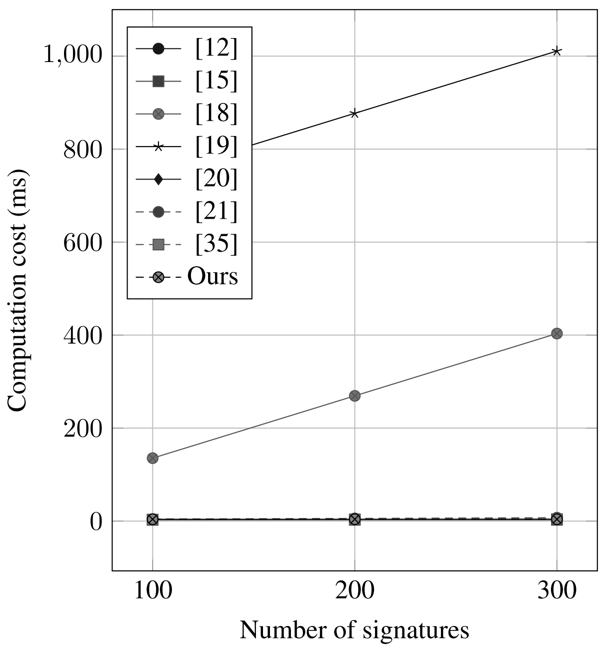


FIGURE 13. Verification cost of *n* authentic signatures.

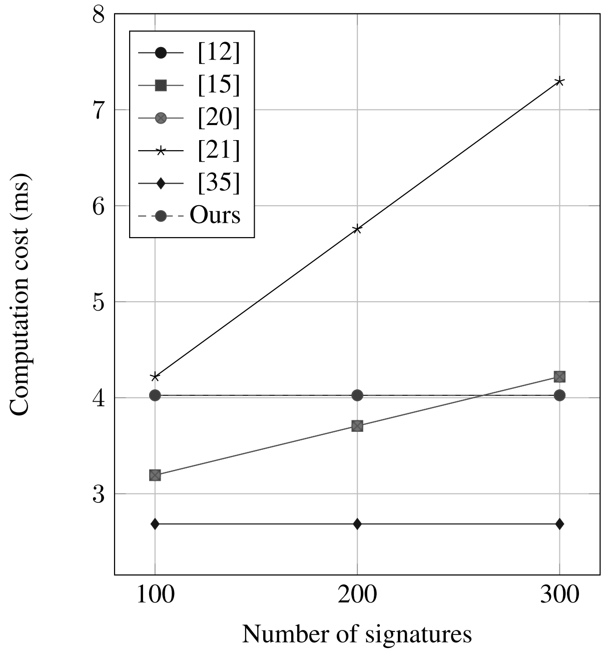


FIGURE 14. Verification cost of *n* authentic signatures without [18] and [19].

However, as seen in Table 5, Tzeng et al.'s [15] scheme does not have a mechanism for verifying n signatures with $b \geq 1$ fake signatures appearing in the batch. Therefore, their scheme is not supposedly suitable to encounter a situation, that possibly happens in the real world, where the adversary broadcasts forged messages to the network. At this stage, when such a condition happens, from the above-compared schemes, only Jiang et al.'s [20], Wang et al.'s [21], and our schemes, that have an illegal signatures identification property. Based on the discussion in Section I, a verifier (RSU or vehicle) has to verify around 600 messages per second. To simplify the calculation, we assume there are 512 messages (n) that come to an RSU with four messages (b) presumably forged. In Jiang et al.'s scheme, it takes 5.30656 ms to verify 512 authentic signatures. Meanwhile, when there are four fake signatures appear in 512 messages, their scheme takes $49PC + 512SC = 68.28656$ ms. For the same case in Wang et al.'s scheme, it takes 10.55968 ms to verify 512 authentic signatures, and $49PC + 4600SC = 89.258$ ms for four fake-included signatures verification. Finally, our scheme only needs 4.02513 ms and 29.48513 ms for without and with four fake signatures from 512, respectively. This result indicates that our proposed scheme can endure the fake signature attacks and provide light computation. This thing is guaranteed by our sorting reputation mechanism that allows our BT-based scheme to be in the best-case scenario state for most of the time. Compared to Jiang et al.'s scheme, which is counted in an average evaluation between best-case and worst-case boundaries. The

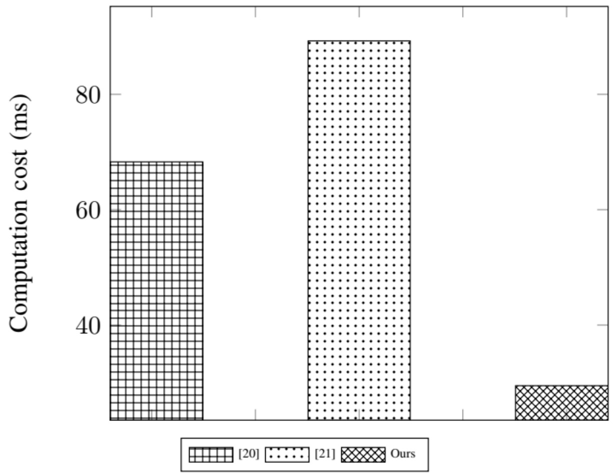


FIGURE 15. Verification cost of $n = 512$ authentic signatures with $b = 4$ illegal signatures.

performance comparison with $b = 4$ and $n = 512$, between Jiang et al.'s, Wang et al.'s, and our schemes are shown in Table 6 and Figure 15.

To sum up, this paper's idea is to enhance the features of our batch verification scheme. Our scheme can efficiently detect a modest amount of illegal signatures that appear in the batch. By giving b fraudulent signatures, the number of pairing operations is becoming high if they are uniformly distributed throughout the leaf nodes. The number of pairing procedures is reduced when they are distributed in the batch. Combined with the proposed reputation management, a particular user can batch verifying the received signature that comes to them. After assessing the sending vehicles' trustworthiness, the subsequent sorting operation can be used to keep the computation low. By such an improvement, when the receiver has all-legal signatures, then the message authentication protocol can handle it well by default. Meanwhile, if the receiver has illegal signatures in the batch, the proposed BT-based batch verification scheme with a reputation management method can eminently complement it.

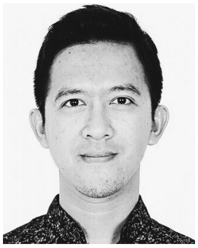
VIII. CONCLUSION

In this paper, we have proposed a lightweight, robust, and practical authentication scheme for V2I (that also could be applied in V2V) communications in VANETs. The security analysis shows that our scheme could withstand non-repudiation, identity privacy-preserving, message authentication, traceability, resistance to replaying attacks, unlinkability, and backward-forward secrecy. To significantly improve the system performance and prevent it from losing its efficiency, we include an extension in our BT-based batch verification scheme as our main point. Our reputation mechanism can guarantee the best-case scenario will appear as much as possible, which keeps the number of computations in finding the illegal signature low. This mechanism can be beneficial for applied in VANETs' environment, particularly for a modest amount of illegal signatures. Because in the real world, we argue if there are more honest people than dishonest ones.

REFERENCES

[1] Z. Jianhong, X. Min, and L. Liying, "On the security of a secure batch verification with group testing for VANET," *Int. J. Netw. Secur.*, vol. 16, no. 4, pp. 313–320, 2014.
[2] X. Shen, X. Cheng, L. Yang, R. Zhang, and B. Jiao, "Data dissemination in VANETs: A scheduling approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 2213–2223, Oct. 2014.
[3] C.-C. Chang, J.-H. Yang, and Y.-C. Wu, "An efficient and practical authenticated communication scheme for vehicular ad hoc networks," *Int. J. Netw. Secur.*, vol. 17, no. 6, pp. 702–707, 2015.
[4] E. F. Cahyadi and M.-S. Hwang, "An improved efficient anonymous authentication with conditional privacy-preserving scheme for VANETs," *PLoS ONE*, vol. 16, no. 9, Sep. 2021, Art. no. e0257044.
[5] Z. Lu, W. Liu, Q. Wang, G. Qu, and Z. Liu, "A privacy-preserving trust model based on blockchain for VANETs," *IEEE Access*, vol. 6, pp. 45655–45664, 2018.
[6] M. Raya and J.-P. Hubaux, "Securing vehicular ad hoc networks," *J. Comput. Secur.*, vol. 15, no. 1, pp. 39–68, 2007.
[7] X. Zhu, S. Jiang, L. Wang, and H. Li, "Efficient privacy-preserving authentication for vehicular ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 63, no. 2, pp. 907–919, Feb. 2014.
[8] M. S. Bouassida, "Authentication vs. privacy within vehicular ad hoc networks," *Int. J. Netw. Secur.*, vol. 13, no. 3, pp. 121–134, 2011.
[9] E. F. Cahyadi and M.-S. Hwang, "A comprehensive survey on certificateless aggregate signature in vehicular ad hoc networks," *IETE Tech. Rev.*, vol. 39, no. 6, pp. 1265–1276, Nov. 2022, doi: 10.1080/02564602.2021.2017800.
[10] C. Zhang, R. Lu, X. Lin, P.-H. Ho, and X. Shen, "An efficient identity-based batch verification scheme for vehicular sensor networks," in *Proc. IEEE 27th Conf. Comput. Commun. (INFOCOM)*, Phoenix, AZ, USA, Apr. 2008, pp. 816–824.
[11] K. Hakuta, Y. Katoh, H. Sato, and T. Takagi, "Batch verification suitable for efficiently verifying a limited number of signatures," in *Proc. Inf. Secur. Cryptol. (ISC)*, Seoul, South Korea, 2012, pp. 425–440.
[12] L. Liu, Y. Wang, J. Zhang, and Q. Yang, "A secure and efficient group key agreement scheme for VANET," *Sensors*, vol. 19, no. 3, p. 482, Jan. 2019.
[13] H.-T. Pan, E. F. Cahyadi, S.-F. Chiou, and M.-S. Hwang, "Research on batch verification schemes for identifying illegal signatures," *Int. J. Netw. Secur.*, vol. 21, no. 6, pp. 1062–1070, 2019.
[14] P. Vijayakumar, M. Azees, A. Kannan, and L. J. Deborah, "Dual authentication and key management techniques for secure data transmission in vehicular ad hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 1015–1028, Apr. 2016.
[15] S.-F. Tzeng, S.-J. Horng, T. Li, X. Wang, P.-H. Huang, and M. K. Khan, "Enhancing security and privacy for identity-based batch verification scheme in VANETs," *IEEE Trans. Veh. Technol.*, vol. 66, no. 4, pp. 3235–3248, Apr. 2017.
[16] C.-C. Lee and Y.-M. Lai, "Toward a secure batch verification with group testing for VANET," *Wireless Netw.*, vol. 19, no. 6, pp. 1441–1449, 2013.
[17] E. F. Cahyadi, C. Damarjati, and M.-S. Hwang, "Research on identity-based batch verification schemes for security and privacy in VANETs," *J. Electron. Sci. Technol.*, vol. 20, no. 3, pp. 1–19, 2022.
[18] M. Azees, P. Vijayakumar, and L. J. Deboarh, "EAAP: Efficient anonymous authentication with conditional privacy-preserving scheme for vehicular ad hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 9, pp. 2467–2476, Sep. 2017.
[19] T. Gu, B. Yuan, Y. Liu, P. Wang, L. Li, and L. Chang, "An improved EAAP scheme for vehicular ad hoc networks," *Int. J. Commun. Syst.*, vol. 33, no. 6, p. e4283, Apr. 2020.
[20] Y. Jiang, M. Shi, X. Shen, and C. Lin, "BAT: A robust signature scheme for vehicular networks using binary authentication tree," *IEEE Trans. Wireless Commun.*, vol. 8, no. 4, pp. 1974–1983, Apr. 2009.
[21] H. Wang, B. Qin, and J. Domingo-Ferrer, "An improved binary authentication tree algorithm for vehicular networks," in *Proc. 4th Int. Conf. Intell. Netw. Collaborative Syst.*, Bucharest, Romania, Sep. 2012, pp. 206–213.

[22] E. F. Cahyadi and M.-S. Hwang, "An improved efficient authentication scheme for vehicular ad hoc networks with batch verification using bilinear pairings," *Int. J. Embedded Syst.*, vol. 15, no. 2, pp. 139–148, 2022.
[23] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Proc. Annu. Int. Cryptol. Conf.*, vol. 2139, 2001, pp. 213–229.
[24] A. Miyaji, M. Nakabayashi, and S. Takano, "New explicit conditions of elliptic curve traces for FR-reduction," *IEICE Trans. Fundamentals*, vol. E84-A, no. 5, pp. 1234–1243, 2001.
[25] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, vol. 2248, 2001, pp. 514–532.
[26] J. Zhang, "A survey on trust management for VANETs," in *Proc. IEEE Int. Conf. Adv. Inf. Netw. Appl.*, Mar. 2011, pp. 105–112.
[27] A. Atanasiu, "A new batch verifying scheme for identifying illegal signatures," *J. Comput. Sci. Technol.*, vol. 28, no. 1, pp. 144–151, Jan. 2013.
[28] R. Hussain, W. Nawaz, J. Lee, J. Son, and J. T. Seo, "A hybrid trust management framework for vehicular social networks," in *Proc. Int. Conf. Comput. Social Netw.*, vol. 9795, 2016, pp. 214–225.
[29] W. Dong, Y. Li, R. Hou, X. Lv, H. Li, and B. Sun, "A blockchain-based hierarchical reputation management scheme in vehicular network," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Waikoloa, HI, USA, Dec. 2019, pp. 1–6.
[30] S. Su, Z. Tian, S. Liang, S. Li, S. Du, and N. Guizani, "A reputation management scheme for efficient malicious vehicle identification over 5G networks," *IEEE Wireless Commun.*, vol. 27, no. 3, pp. 46–52, Jun. 2020.
[31] M. Bellare, J. A. Garay, and T. Rabin, "Fast batch verification for modular exponentiation and digital signatures," in *Advances in Cryptology—EUROCRYPT* (Lecture Notes in Computer Science), vol. 1403. Berlin, Germany: Springer, 1998, pp. 236–250.
[32] S.-J. Horng, S.-F. Tzeng, Y. Pan, P. Fan, X. Wang, T. Li, and M. K. Khan, "B-SPECS+: Batch verification for secure pseudonymous authentication in VANET," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 11, pp. 1860–1875, Nov. 2013.
[33] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing," *J. Cryptol.*, vol. 17, no. 4, pp. 297–319, 2004.
[34] M. Nabil, M. Bima, A. Alsharif, W. Johnson, S. Gunukula, M. Mahmoud, and M. Abdallah, "Priority-based and privacy-preserving electric vehicle dynamic charging system with divisible e-payment," in *Smart Cities Cybersecurity and Privacy*. New York, NY, USA: Elsevier, 2019, pp. 165–186.
[35] K.-A. Shim, "Reconstruction of a secure authentication scheme for vehicular ad hoc networks using a binary authentication tree," *IEEE Trans. Wireless Commun.*, vol. 12, no. 11, pp. 5386–5393, Nov. 2013.
[36] I. A. Kamil and S. O. Ogundoyin, "An improved certificateless aggregate signature scheme without bilinear pairings for vehicular ad hoc networks," *J. Inf. Secur. Appl.*, vol. 44, pp. 184–200, Feb. 2019.
[37] Y. Ming and X. Shen, "PCPA: A practical certificateless conditional privacy preserving authentication scheme for vehicular ad hoc networks," *Sensors*, vol. 18, no. 5, p. 1573, May 2018.
[38] I. Ali, M. Gervais, E. Ahene, and F. Li, "A blockchain-based certificateless public key signature scheme for vehicle-to-infrastructure communication in VANETs," *J. Syst. Archit.*, vol. 99, Oct. 2019, Art. no. 101636.



EKO FAJAR CAHYADI (Member, IEEE) received the B.Eng. degree in electrical engineering from the Institut Sains dan Teknologi AKPRIND Yogyakarta, in 2009, the M.Sc. degree in electrical engineering from the Institut Teknologi Bandung, Bandung, in 2013, and the Ph.D. degree in computer science and information engineering from Asia University, Taichung, in 2022. He is a Lecturer with the Faculty of Telecommunication and Electrical Engineering, Institut Teknologi Telkom Purwokerto, Indonesia. His current research interests include network security protocol and mobile communications.

MIN-SHIANG HWANG (Member, IEEE) received the B.S. degree in electronic engineering from the National Taipei Institute of Technology, Taipei, Taiwan, in 1980, the M.S. degree in industrial engineering from National Tsing Hua University, Taiwan, in 1988, and the Ph.D. degree in computer and information science from National Chiao Tung University, Taiwan, in 1995. He studied applied mathematics at National Cheng Kung University, Taiwan, from 1984 to 1986. He passed the National Higher Examination in the field of electronic engineer, in 1988. He also passed the National Telecommunication Special Examination in the field of information engineering, qualified as an Advanced Technician (Hons.), in 1990. From 1988 to 1991, he was the Computer Center Leader at the Telecommunication Laboratories (TL), Ministry of Transportation and Communications, where he was the Project Leader of research in computer security, in July 1990. His current research interests include database and data security, cryptography, image compression, and mobile communications. He is a member of ACM and the Chinese Information Security Association. He was a recipient of the 1997, 1998, and 1999 Distinguished Research Awards from the National Science Council, Taiwan.