

Article

Online/Offline VANETs with Lightweight Authentication Framework for Vehicular Communication

Pingyuan Zhang * and Limin Wang

School of Computer Science and Artificial Intelligence, Zhengzhou University of Light Industry,
Zhengzhou 450002, China; 2019838@zzuli.edu.cn

* Correspondence: jyzhang@zzuli.edu.cn

Abstract

Vehicular Ad Hoc Networks (VANETs) are mobile networks that offer new services and communication between moving vehicles, roadside infrastructure, and a trusted authority. With the development of autonomous and connected vehicles, the issue of authentication in VANETs has become increasingly prominent due to the lack of mutual trust among network entities. However, standard authentication models for VANETs must account for total computational and communication overhead, regardless of the timing of authentication message generation. To address this limitation, this work proposes an advanced authentication paradigm for VANETs called the online/offline VANET framework, and formalizes this novel framework to realize lightweight authentication by shifting heavy computational overhead to the offline phase. The proposed model is divided into an offline phase and an online phase. In the offline phase of the free time before the message becomes available, it allows more powerful trusted authority to pre-compute, and in the online phase, resource-constrained devices only execute a small set of residual operations. Based on this model and a new identity-based signature, we give an efficient instantiation and use a mobile platform to evaluate it. The experimental results demonstrate that our construction achieves low online computational and communication overhead.

Keywords: lightweight; Vehicular Ad Hoc Networks (VANETs); authentication protocol; security; online/offline

1. Introduction

Vehicular Ad Hoc Networks (VANETs) are networks formed by vehicles, trusted authority, and roadside units, enabling direct communication between vehicles (V2V) and between vehicles and roadside infrastructure (V2I) [1]. VANETs emerged alongside the proliferation of wireless communication technologies and the surging demand for Intelligent Transportation Systems (ITS), and they carry far-reaching practical value. As the fundamental communication layer of ITS, VANETs can mitigate traffic accidents by delivering real-time safety notifications, including collision avoidance reminders, speed alerts, and traffic hazard warnings. Beyond safety, VANETs also support intelligent traffic management, such as optimizing routes, reducing congestion, and contributing to the development of autonomous driving systems [2,3]. In short, VANETs help to improve road safety, optimize traffic flow, and offer crucial support for autonomous driving by exchanging data in real time.

The architecture of VANETs is illustrated in Figure 1. It consists of mobile vehicles, a trusted authority (TA), and roadside units (RSUs). In some models, the cloud server is



Academic Editor: Barbara M. Masini

Received: 20 May 2026

Revised: 25 June 2026

Accepted: 2 July 2026

Published: 7 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

also used to store some public data by interacting with RSUs to reduce the local VANET storage. The mobile vehicles communicate with one another and RSUs in that particular region by on-board units (OBUs) which are installed on vehicles. TA is a trusted authority with sufficient computation and storage capabilities than other entities in this work, and provides services to vehicles through RSUs. RSUs are fixed as the roadside infrastructures that mainly provide services and traffic information to vehicles.

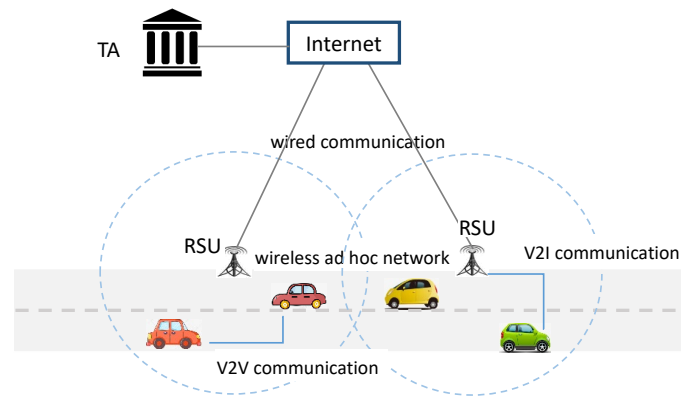


Figure 1. The architecture of VANETs.

In a VANET, secure authentication and high efficiency are the essential requirements for vehicular communication. First, the communication in VANETs has to be carried out in a secure way in order to identify and also isolate the attackers from the network. This contributes to providing services only to the legitimate users, and the false information provided by attackers can also be identified [4–6]. Thus authentication is a primary requirement in a VANET for secure communication, which only allows legitimate entities to join in this network, and ensures the traffic-related messages made by the entities are inerrant.

In addition to the security, a VANET with authentication should also be lightweight in terms of the low computation and communication overhead [7–9]. Lightweight authentication is highly significant due to the following arguments.

- The OBUs installed on vehicles are commonly low-power devices. Due to resource limitations, such as the computation capacity and bandwidth, the computation costs and the number of bits sent in wireless communication should be minimized.
- Considering a scenario where larger numbers of vehicles must be authenticated, the authentication must be rapid and it would be highly desirable if it only uses the minimum number of resources. Otherwise, the related devices and communication network may be overloaded because of the numerous accesses.
- Low computation and communication costs contribute to smaller overhead and lower delay, and thus bring the owners of the vehicles a better experience, which thus makes it easier for them to accept and also use the VANET network.

However, existing lightweight authentication protocols designed for standard VANETs, such as those proposed in [1,2,7–9], must account for the communication and computational overhead incurred throughout the entire authentication procedure. This includes the overhead generated during the offline phase prior to the availability of traffic messages, regardless of the timing at which these messages are ready for transmission. Such overall overhead introduces heavy computational and communication burdens for resource-limited OBUs deployed on vehicles. To mitigate this limitation, this paper proposes and formalizes a novel framework built on pre-computation and offline distributed mechanisms. Within this framework, all offline computational overhead is borne in advance by the TA; thus, vehicles are freed from handling any offline-side costs before traffic

messages become available. Accordingly, the proposed framework serves as an effective alternative to alleviate the excessive communication and computational overhead in conventional full-process VANET authentication schemes.

1.1. Our Result

In this work, a new variant of VANET model called the online/offline VANET is introduced, which provides a new and lightweight V2V authentication framework by shifting heavy computational overhead to the offline phase. The comparison of the standard model and our model is shown in Figure 2. Compared to standard VANETs with the authentication model, our model is divided into an offline phase and an online phase. The offline phase is performed by a powerful TA in free time before the traffic-related messages become available. Thus, our proposed model achieves online performance optimization at the cost of increased offline system overhead.

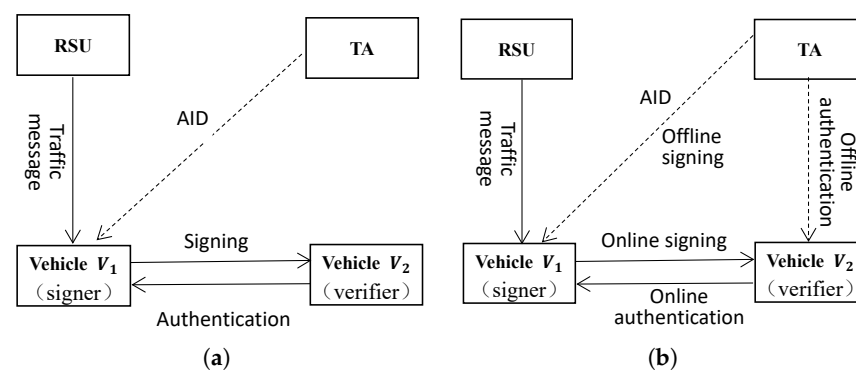


Figure 2. The comparison of different VANET authentication models, where the dashed arrows represent the offline authentication phase of any free time and the solid arrows represent the online authentication phase. (a) Standard model. (b) Proposed online/offline model.

More specifically, a scenario is considered where the OBUs in vehicles have to efficiently authenticate themselves in VANETs, but they are powerless because of the constrained computation and storage power. Meanwhile, our proposed online/offline model allows a more powerful TA to pre-compute before the traffic-related message is given. Thus, OBUs can quickly execute the online authentication based on the information generated in the offline phase. Thus our online/offline model has the perfect property for secure and lightweight V2V communication in VANETs. To sum up, we describe the contributions as follows.

- A New Framework for Achieving Efficient Authentication in VANET Systems. Our framework is different from directly applying an online/offline identity-based signature to standard VANETs. Firstly, due to the architecture of the standard VANET model, directly applying an online/offline signature to it is only applicable during the signing stage of the authentication scheme, whereas our online/offline VANET model is not their simple combination, which formally gives the definition of online/offline VANET and its security model. Some steps of both the signing and verifying stages can not only be performed offline, but can also be executed by more powerful TAs instead of vehicles. Secondly, the signing vehicle in our framework can extract the signature without inputting the private key in an online signing algorithm and only by using the pre-computed values from TAs. That is, the TA keeps the generated private keys secret by itself, which avoids the issue of secure key distribution in public communication channels. Finally, our framework also supports distributed computing in offline authentication stages besides the online/offline idea. In fact, the cloud server can also execute the offline verifying stage to provide data processing and storage together with the TA.

- **An Efficient Instantiation for Online/Offline VANET Authentication Model.** According to our online/offline approach, this work first designs a new identity-based online/offline signature and proposes a generic framework for the authentication model, then gives an efficient instantiation and its security proof. Our instantiation is far from the simple combination of the signature scheme and our online/offline framework. It adds the pre-processing and distributed computing executed by the TA, and avoids the issue of key distribution in the public communication channel. This is also the first attempt to achieve secure and lightweight authentication in the online/offline VANET model for V2V communication.
- **Lightweight Online Authentication for V2V Communication.** The high efficiency of our construction is considered to be a natural result. It stems from our model in which the OBUs in vehicles merely need to perform online costs because the offline phase has been performed in advance by the TA. Simple comparisons between the standard model and our model are presented in Table 1. Both the theoretical analysis and experimental results demonstrate that our scheme reduces the vehicle's online computation and communication costs, while introducing additional offline overhead such as TA processing and storage.

Table 1. Simple comparisons of the efficiency between standard VANET and our online/offline VANET with authentication, where the signature of the traffic-related message is $\sigma := \{\sigma_{\text{off}}, \sigma_{\text{on}}\}$, and is denoted by $T_{\text{off}}, T_{\text{on}}$ the time needed for extracting $\sigma_{\text{off}}, \sigma_{\text{on}}$, respectively.

Approaches	The Standard Model	Our Online/Offline Model
Extracted in offline phase	—	$\{\sigma_{\text{off}}\}$
Extracted in online phase	$\{\sigma_{\text{off}}, \sigma_{\text{on}}\}$	$\{\sigma_{\text{on}}\}$
Online authentication time	$T_{\text{off}} + T_{\text{on}}$	T_{on}

1.2. Related Works

Larger numbers of works surrounding the VANET authentication problem have been studied. In addition to the schemes based on asymmetric key management [10–12], early works also applied symmetric key cryptography for information authentication [13–15]. The main defect of these schemes is that the authenticate process requires a third trusted party. This is not very applicable for larger numbers of vehicle-to-vehicle communications. Hence, public-key-based VANET authentications are widely proposed, and they are also classified into public-key-infrastructure (PKI)-based schemes [16] and identity (ID)-based schemes. PKI-based authentication schemes for VANETs are still not feasible, because this paradigm requires numerous certificates, and suffers from space problems as well as heavy communication and computation overheads [17,18].

A larger number of ID-based VANET authentication protocols [19–24] have been constructed to reduce computation overhead. In ID-based protocols, certificate managements are simplified by digital signatures, and vehicle identities are used as the keys in ID-based cryptography for communication [25]. The work in Ref. [26] constructed an efficient ID-based authentication protocol using ID-based cryptography. Shim [27] also proposed an ID-based authentication protocol for vehicular communication based on bilinear pairing. However, this protocol has high computation costs because of several operations. On account of the heavy computation overhead, the work in Ref. [28] designed an efficient VANET authentication protocol without using bilinear pairing. In addition, when receiving larger numbers of authentication messages from multiple vehicles, batch verification is also a very efficient method to reduce the computation overhead [29]. In this work, we found that authentication for VANETs using online/offline ID-based cryptography is also feasible to reduce computation overhead. To this end, we formalize a new authentication

framework called the online/offline VANET by shifting heavy computational overhead to the offline phase, and give an efficient instantiation by a new signature algorithm.

1.3. Organization

After the preliminaries in Section 2, Section 3 proposes the proposed authentication framework and gives its security model. Section 4 first designs a new type signature for our construction, and then proposes a concrete online/offline VANET with authentication protocol. Section 5 proves the security of our scheme. Section 6 discussed the efficiency of our scheme in terms of the communication and computation costs. We finally concludes our work in Section 7.

2. Preliminaries

2.1. Notations

We show the frequently used notations in Table 2.

Table 2. The notations.

Notations	Descriptions
TA	A trusted authority
RSU	Roadside unit
OBU	On-board unit
V_i, V_j	Signer vehicle, verifier vehicle
ID	The original identity
AID	The anonymous identity
m	Traffic-related message
σ	The generated signature
T_i	Timestamp for V_i
$\mathbb{Z}_q$	Ring of integers modulo q
$x \leftarrow \mathbb{Z}_q$	x is uniformly sampled from $\mathbb{Z}_q$

2.2. Related Cryptographic Primitive

Definition 1 (Online/Offline Identity-Based Signature). *The definition of this signature scheme [30] includes the following five algorithms:*

- *Setup (1^λ): This algorithm inputs the security parameter λ , then outputs the public parameter PK and master secret key SK .*
- *Key extraction (PK, SK, ID): It inputs PK , SK and an identity ID , then outputs private key sk_{ID} .*
- *Offline sign (PK, sk_{ID}): Offline signing algorithm inputs PK and the private key sk_{ID} , then outputs the pre-computation value ψ before the message is available.*
- *Online sign (m, ψ): Given the message m and the pre-computation ψ , it outputs the final signature σ .*
- *Verify (PK, m, σ, ID): Verify algorithm inputs PK , (m, σ) and ID , then outputs ACCEPT or REJECT.*

The security of our construction is based on the discrete logarithm problem. We now recall its definition.

Definition 2. *Let G be a cyclic group with generator g . For a uniform element $h \in G$, the integer x satisfying $g^x = h$ is called the discrete logarithm. The discrete logarithm problem is to compute x .*

2.3. Security Requirements of Our VANET Framework

In this work, a VANET with authentication for V2V communication must have the following security features.

- **Authentication and Data Integrity:** Authentication is an essential requirement in VANETs, which only allows legitimate entities to join in this network. Thus, legitimate vehicles must be authenticated by other entities. In addition, the VANETs must also ensure the traffic-related messages sent by the entities are true and intact.
- **Unlinkability:** Any adversaries cannot link the data derived from the vehicle communication, such as a vehicle's driving information, to the genuinely related vehicle.
- **Anonymity:** In the authentication stage of the VANETs, anonymity means that the identity information of the user will not be leaked. Thus any attackers, including roadside infrastructure, cannot link a vehicle's information with the user's identity.
- **Traceability:** If there is a vehicle that was not certified or has transmitted a forged message, the TA is the only one that can trace this vehicle's original identity.

3. Our Online/Offline VANET Framework for V2V Communication

3.1. The Proposed Framework

Based on the identity-based signature primitive [30] and the idea of online/offline [31,32], our framework is described as follows.

Definition 3. Our online/offline VANET framework for V2V authentication mainly consists of seven algorithms:

- **Setup (1^λ):** This algorithm inputs security parameter λ . The TA outputs PK and SK for identity-based signature.
- **Vehicle AID Generation (ID):** The vehicle AID generation algorithm inputs the identity ID by a vehicle V_i , and outputs anonymous vehicle identity (AID). This algorithm is conducted by the TA, which allows only the TA to recover the original ID by a secret key or trapdoor in order to trace the illegal vehicles.
- **Vehicle Key Extraction (AID, PK, SK):** The vehicle key extraction algorithm inputs the AID, PK and SK by the TA, and extracts the private key sk_{ID} for online/offline identity-based signatures. The TA keeps the private key sk_{ID} secret by itself.
- **Offline Sign Pre-Computation (PK, AID, sk_{ID}):** It inputs PK and private key sk_{ID} by the TA, and outputs the pre-computation value ψ . Notice that the private key sk_{ID} is used in the offline phase by a more powerful TA before the traffic-related message is available.
- **Online Message Sign (m, ψ, T):** The online message sign algorithm inputs the traffic-related message m , the pre-computation ψ and the timestamp T . The signing vehicle V_i executes this algorithm to extract final signature σ without using the private key sk_{ID} , and sends it and T to the verifying vehicle V_j .
- **Offline TA Verify (ψ, PK):** The offline TA verify algorithm takes as input the pre-computed value ψ and PK. The TA outputs p , and then sends it to the verifying vehicle V_j . It also allows the TA to execute this process before the online message sign algorithm because it only takes as input ψ and PK.
- **Online Verify (p, σ, m):** The online verify algorithm inputs p and the message–signature pair (m, σ) from the vehicle V_i . The verifying vehicle V_j accepts or rejects the authentication request.

3.2. Security Model

According to their different capabilities of the adversaries, we now describe two types of security model by two games I and II between a challenger $\mathcal{C}$ and two adversaries $\mathcal{A}_1$ and $\mathcal{A}_2$ respectively.

Type-I Adversary: The adversary $\mathcal{A}_1$ may attempt to replace the legitimate partial pre-computation and public parameters with illegal ones and acts as a malicious user.

Type-II Adversary: Define the adversary $\mathcal{A}_2$ as a trusted but curious TA who can get the master secret key but can not replace the partial pre-computation of the system. $\mathcal{A}_2$ may attempt to create multiple forged identities by producing new public–private key pairs.

Game I: Game I is defined as a protocol between $\mathcal{C}$ and $\mathcal{A}_1$ and can be described as follows.

1. Setup Phase: $\mathcal{C}$ first generates system parameters by running the setup algorithm and sends them to $\mathcal{A}_1$.
2. Query Phase: In this phase, $\mathcal{A}_1$ executes the following queries:
 - Pre-processing queries: The adversary $\mathcal{A}_1$ queries on pre-computation of the AID . $\mathcal{C}$ generates the corresponding ψ and sends it to $\mathcal{A}_1$.
 - Private key queries: $\mathcal{A}_1$ queries on the private key of the AID . $\mathcal{C}$ generates the corresponding sk_{ID} and sends it to $\mathcal{A}_1$.
 - Partial pre-computation replacement queries: $\mathcal{A}_1$ sends the new pre-computation ψ' of the AID to the challenger $\mathcal{C}$. $\mathcal{C}$ replaces the original ψ with ψ' and sends it to $\mathcal{A}_1$.
 - Signing queries: $\mathcal{A}_1$ queries on the signature of the AID on adaptively chosen message m . $\mathcal{C}$ runs the related algorithm to generate the corresponding signature σ and sends it to $\mathcal{A}_1$.
3. Forgery Phase: The adversary $\mathcal{A}_1$ produces a forged triple (AID^*, m^*, σ^*) and wins the game if
 - σ^* is a valid signature.
 - $\mathcal{A}_1$ has never queried the signature on (AID^*, m^*) .
 - $\mathcal{A}_1$ has never queried the private key sk_{ID^*} of AID^* .

Game II: Game II is defined as a protocol between $\mathcal{C}$ and $\mathcal{A}_2$ and can be described as follows.

1. Setup Phase: $\mathcal{C}$ first generates the system parameter and the master secret key by running the setup algorithm and sends both of them to $\mathcal{A}_2$.
2. Query Phase: In this phase, $\mathcal{A}_2$ executes the following queries:
 - Private key queries: $\mathcal{A}_2$ queries on the private key of the AID . $\mathcal{C}$ generates the corresponding sk_{ID} and sends it to $\mathcal{A}_2$.
 - Pre-processing queries: The adversary $\mathcal{A}_2$ queries on pre-computation of the AID . $\mathcal{C}$ generates the corresponding ψ and sends it to $\mathcal{A}_2$.
 - Signing queries: $\mathcal{A}_2$ queries on the signature of the AID on adaptively chosen message m . $\mathcal{C}$ runs the related algorithm to generate the corresponding signature σ and sends it to $\mathcal{A}_2$.
3. Forgery Phase: The adversary $\mathcal{A}_2$ produces a forged triple (AID^*, m^*, σ^*) and wins the game if
 - σ^* is a valid signature.
 - $\mathcal{A}_2$ has never queried the signature on (AID^*, m^*) .
 - $\mathcal{A}_2$ has never queried the private key sk_{ID^*} of AID^* .

Definition 4. Our online/offline VANET with the authentication protocol is existentially unforgeable under the adaptively chosen message attacks (EU-CMA) if the advantages of winning the above games for $\mathcal{A}_1$ and $\mathcal{A}_2$ are negligible.

4. Our Construction

This section will give an instantiation in VANETs for V2V lightweight authentication based on the above framework.

4.1. A New Online/Offline Identity-Based Signature Scheme

Before the formal construction, we first need to design a new online/offline identity-based signature scheme. The proposed signature scheme is based on Schnorr signature [33] and Γ -signatures [34]. Suppose G is a cyclic group of prime order q , and g is a generator. $H_i : \{0, 1\}^* \rightarrow \mathbb{Z}_q, i = 1, 2, 3$ are three hash functions, and m is the message. Then our online/offline identity-based signature scheme is described in Table 3.

Table 3. The proposed online/offline identity-based signature scheme.

Setup:	Online sign (ψ, m):
01 $x \leftarrow \mathbb{Z}_q$	12 $c_3 := H_3(m)$
02 $X := g^x$	13 $z := y + uc_3 \mod q$
03 Return $PK = (G, q, g, X)$ and $SK = x$	14 Return signature $\sigma := (z, g^y, g^r)$
Extract (PK, SK, ID):	Verify (m, σ, ID, PK):
04 $r \leftarrow \mathbb{Z}_q$	15 $c_1 := H_1(g^r, ID)$
05 $c_1 := H_1(g^r, ID)$	16 $c_2 := H_2(g^y, ID)$
06 $s := r + c_1x \mod q$	17 $c_3 := H_3(m)$
07 Return private key $sk_{ID} = (s, g^r)$	18 ACCEPT if $g^z = g^y((g^r X^{c_1})^{c_2})^{c_3}$
Offline sign (sk_{ID}, PK):	19 Otherwise, return Reject
08 $y \leftarrow \mathbb{Z}_q$	
09 $c_2 := H_2(g^y, ID)$	
10 $u := sc_2 \mod q$	
11 Return $\psi := (y, g^y, u, g^r)$	

It is easy to prove that our scheme is correct because

$$\begin{aligned}
 g^z &= g^y((g^{r+c_1x})^{c_2})^{c_3} \\
 &= g^y((g^r g^{xc_1})^{c_2})^{c_3} \\
 &= g^y((g^r X^{c_1})^{c_2})^{c_3}
 \end{aligned}$$

where $c_1 = H_1(g^r, ID)$, $c_2 = H_2(g^y, ID)$ and $c_3 = H_3(m)$.

Our proposed scheme uses three hash functions for signature generation, which may increase the complexity of the whole authentication process. While most of the existing identity-based signatures based on the discrete logarithm only use two hash functions, the whole signing time of our scheme may be slightly longer than those of the existing schemes. However, our scheme has a shorter online signing time as it does not need to perform more time-consuming exponent operation g^y , and both the existing schemes and our scheme only perform one hash operation in the online authentication stage.

4.2. High Description

Our construction is more than the combination of our proposed framework and the above signature scheme. In fact, because our proposed signature scheme is an online/offline form, there will be some pre-computed values in this scheme when it is applied to our framework for VANETs. As a consequence, a few elements of the protocol, e.g., partial signature g^y , could be pre-computed and sent in advance before the initiation of the authentication. This contributes to reducing the online communication and computation costs for vehicles.

Beyond that, in standard VANETs, the TA will send the anonymous identity AID and the corresponding private key sk_{ID} to vehicle V_i by the secure channel, while in our construction, the TA will not transmit them to V_i as they will only be used offline by the

TA to generate the pre-computation $\psi = (y, g^y, u, g^r)$, and then sends (y, u) and g^y to the signing vehicle V_i and verifying vehicle V_j , respectively. The tuple (y, u) and the value g^y are used for online signing and online verifying, respectively. The value g^r is used for offline verifying. For multiple authentications, the TA can perform this offline algorithm for β times, and get the pre-computed values

$$\psi = \{(y_1, g^{y_1}, u_1, g^{r_1}), \dots, (y_\beta, g^{y_\beta}, u_\beta, g^{r_\beta})\}.$$

4.3. Formal Construction

We now describe the formal construction of our online/offline VANET with the authentication protocol for V2V communication, and present the sketch of the process in Figure 3.

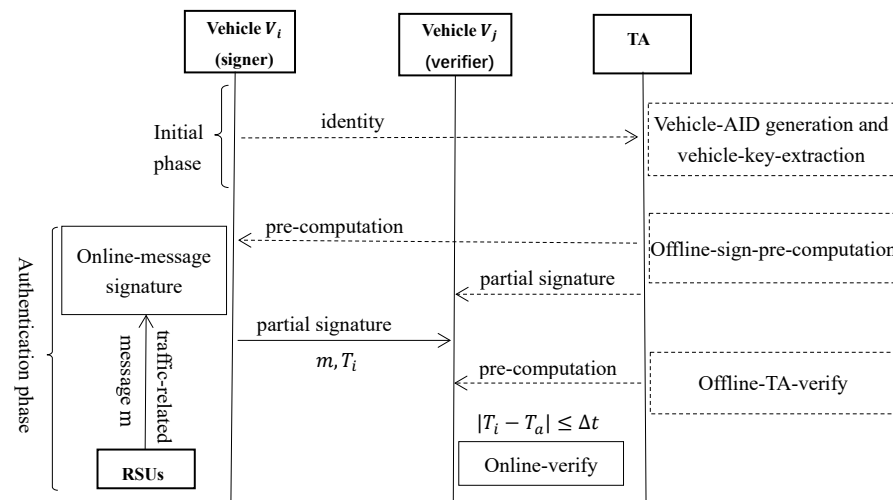


Figure 3. Illustration of our proposed VANET protocol. The dashed boxes and arrows represent the offline computation and communication. The solid boxes and arrows represent the online computation and communication respectively.

- Setup (1^λ): In this step, the TA initializes the VANETs to get SK and the system parameter.
 1. According to the security parameter λ , the TA chooses a cyclic group G that is of prime order q and g is a generator.
 2. The TA chooses $H_i : \{0, 1\}^* \rightarrow \mathbb{Z}_q, i = 1, 2, 3$.
 3. The TA randomly chooses $x \leftarrow \mathbb{Z}_q$ and takes $SK = x$ as the master secret key.
 4. Set $PK = (G, q, g, X, H_1, H_2, H_3)$ where $X = g^x$. Then the TA publishes it publicly across the network.
- Vehicle AID Generation and Vehicle Key Extraction: The vehicle V_i transmits its ID to the TA to join the network. The TA responds to an anonymous identity AID for V_i and generates a corresponding sk_{ID} for the subsequent steps.
 1. The TA chooses an encryption algorithm Enc and gets the public-secret key pair (pk, sk) .
 2. Given ID of the V_i , the TA computes $AID = Enc_{pk}(ID) \parallel ET$, where ET is the validity of time for the V_i .
 3. The TA chooses $r \leftarrow \mathbb{Z}_q$.
 4. The TA computes $c_1 = H_1(g^r, AID)$ and $s = r + c_1 x \mod q$.
 5. It then sets $sk_{ID} = (s, g^r)$ as the private key for the V_i .
 6. By executing γ times, the TA stores

$$\Lambda = \{(AID_1, sk_{ID,1}), \dots, (AID_\gamma, sk_{ID,\gamma})\}.$$

- **Offline Sign Pre-Computation** (PK, AID, sk_{ID}): Before the traffic-related message m is presented, this algorithm is allowed to be executed in any time by the TA instead of the vehicle V_i .
 1. The TA randomly chooses a pair (AID, sk_{ID}) from Λ .
 2. The TA computes $c_2 = H_2(g^y, AID)$ where $y \leftarrow \mathbb{Z}_q$.
 3. It then computes $u = sc_2 \bmod q$.
 4. The TA gets the pre-computed value $\psi = (y, g^y, u, g^r)$, and then sends (y, u, g^y) to the signing vehicle V_i .
- **Online Message Sign** (m, ψ, T): Once the traffic-related message m is presented by RSUs in a particular region, the vehicle V_i performs this phase to authenticate itself.
 1. V_i extracts the pre-computation (y, u, g^y) sent by the TA.
 2. For a valid timestamp T_i , V_i computes $c_3 = H_3(m, T_i)$ and $z = y + uc_3 \bmod q$.
 3. V_i outputs a part of the signature $\sigma = (z, g^y)$ and sends it to the vehicle V_j for a specific interval of time T_i .
- **Offline TA Verify** (ψ, PK): This phase can also be executed by a cloud server beyond the TA, which contributes to alleviating the computation and storage costs of the TA by this distributed processing.
 1. Given PK, AID and the value g^r in ψ , the TA pre-computes

$$p := (g^r X^{c_1})^{c_2}$$

where $c_1 = H_1(g^r, AID)$ and $c_2 = H_2(g^y, AID)$, and sends the value p to the vehicle V_j .

- **Online Verify** (p, σ, m): After receiving the pre-computation p and message/signature pair (m, σ) , the verifying vehicle V_j performs this verifying phase.
 1. V_j first checks the freshness of the message–signature tuple (m, σ, T_i) . Let T_a be the arrival time of the message–signature tuple, and Δt be the time interval that is already defined. If $|T_i - T_a| > \Delta t$, V_j rejects the authentication request. Otherwise, V_j continues this step.
 2. Given the partial signature (z, g^y) on the message m , the timestamp T_i and the value p in the offline TA verify algorithm, the vehicle V_j accepts the authentication request if $g^z = g^y p^{c_3}$ where $c_3 = H_3(m, T_i)$. Otherwise, V_j rejects the authentication request.

Remark 1. Notice that the private key sk_{ID} is only used in the offline phase by the more powerful TA before the traffic-related message is available. Instead of sk_{ID} , the signing vehicle V_i uses the pre-computed tuple (y, u, g^y) generated by sk_{ID} to extract the signature on a message. Thus in some sense, the vehicle V_i can execute the online message sign process without using the private key sk_{ID} .

Remark 2. The offline TA verify process could be executed in advance before the online message sign process as it only takes as input ψ and PK . That is, it allows the TA to execute the partial verifying process offline before the initiation of the real-time authentication. Notice that the elements used in this process will not leak any private information of our proposed protocol. Thus in order to alleviate the computation and storage costs of the TA, this offline verifying process can also be performed by the cloud server, which will provide data processing and storage for pre-computation $p := (g^r X^{c_1})^{c_2}$ where g^r is an element in ψ .

Remark 3. Assuming the TA is compromised and sends malicious messages, the vehicle's response can be divided into two situations. If the TA is only curious and attempts to create multiple forged

identities by producing new key pairs, acting as the type-II adversary in our security model, our scheme is still secure. The vehicle will verify and confirm that the message is invalid. If the TA is completely compromised and can make any attacks within its capacities, it can issue legitimate certificates and fake warnings or traffic messages with valid signatures. At this point, relying solely on all cryptography-based schemes is completely ineffective. In practical VANETs, the response mechanism of the vehicles will evaluate the credibility of the message source, and make judgments and responses relying on the trust management system of the whole network. The evaluation is mainly based on practical traffic conditions around vehicles, such as V2V and V2I cooperation verification. According to the evaluation results, the vehicles may mark and isolate the message, then use distributed consensus by multiple vehicles and RSUs to withdraw the legitimate identity of this TA. In short, from a purely cryptographic perspective, we have to assume that the TA cannot be completely captured. But in practical VANETs, vehicles make judgments and responses relying on the trust management system of the whole network besides cryptography.

Remark 4. In our VANET scheme, an RSU broadcasts the traffic message to the witness vehicle that acts as a sender or signer. The vehicle only sends the message–signature pair to other vehicles in the range of this RSU. Thus, the TA also only broadcasts the corresponding p to the vehicles by this RSU in its range. What is more, our scheme is also slightly lightweight compared with the standard schemes from this perspective, as the standard schemes need to broadcast the values g^r , c_1 , and c_2 to the verifier vehicles in the range of the RSU as well.

4.4. Discussions on Pre-Computed Values

We now discuss the following issues caused by pre-computed values in our scheme.

- **Distribution and Storage:** The TA generates pre-computed values in the offline stage, and distributes them to legitimate vehicles by the RSU within its range. The distribution of the pre-computed value will be bound to its corresponding timestamp and the AID to ensure the consistency of authentication. Furthermore, the hierarchical storage architecture stores the AID mapping list, private key, and ψ within the hardware security module (HSM), whereas the pre-computed values p are placed in regular flash memory managed by the TA or cloud, which eliminates excessive storage and distribution overhead concentrated on a single central node.
- **Freshness:** The freshness of pre-computed values generally follows the AIDs' refresh rules, which include fixed periodic refresh and event-driven emergency refresh. Because of our pre-computation mechanism, we may set the fixed refresh cycle of hundreds of seconds according to different traffic scenarios. Of course, for the event-driven emergency refresh such as TA handover and risk-triggered scenarios, the refresh cycle might be relatively shorter.
- **Reuse and Exhaustion:** To prevent pre-computed values from being reused infinitely, our scheme follows the AID replacement strategy. Vehicles are required to periodically change the AID. Upon expiration, this AID will be discarded unless the traceability mechanism is triggered during the period. When a dispute or maliciousness arises, the TA may extract the real identity of this vehicle from some related pre-computed values to enforce accountability. Meanwhile, each AID has the validity period. Once the TA completes the authentication with vehicles, it will compute and store new AIDs to replenish the exhausted resources, thus ensuring the continuity of the service.
- **Revocation:** An AID and pre-computation revocation list (APRL) maintained by the TA and RSUs are deployed in practical authentication. Once vehicles quit the network or violate network rules, all pre-computed values bound to the AID will be added to APRL synchronously. In addition, once an AID expires, the pre-computed values bound to it will also be revoked by the TA.

- Possible Compromise: The pre-computed *AID-ID* mapping list and the corresponding private keys are securely stored by the TA within HSM, while other pre-computed values do not disclose any information about the private key and vehicle *ID*. Suppose an adversary can query on or replace partial pre-computed values in advance, our scheme is still secure against this attack according to the type-I adversary defined in our security model.

5. Security Analysis

We first prove the security of the proposed protocol by our security model, and then discuss various attacks and security objective analyses.

5.1. Security Proof

Theorem 1. For any probabilistic polynomial time adversaries $\mathcal{A}_1$ and $\mathcal{A}_2$ defined in our security model, our proposed protocol has the EU-CMA security under the discrete logarithm assumption and random oracle model.

Proof. We first prove the security under the type-I adversary $\mathcal{A}_1$. Suppose there is an adversary $\mathcal{A}_1$ who can replace the legitimate partial pre-computation and public parameters, and then break our proposed protocol under the defined model. We build a simulator $\mathcal{B}$ to extract a solution to the discrete logarithm problem by interacting with $\mathcal{A}_1$. Given a problem instance (g, g^x) over the cyclic group G , in order to find a solution x , the simulator $\mathcal{B}$ controls random oracles and interacts with $\mathcal{A}_1$ as follows.

Setup: The simulator $\mathcal{B}$ chooses the public parameters $(G, q, g, X, H_1, H_2, H_3)$ and sends them to $\mathcal{A}_1$. The parameter X is obtained from the problem instance.

H_1 -Queries: $\mathcal{B}$ sets an initially empty hash list L_{H_1} which contains the queries and the corresponding responses. For a query on (AID_k, g^{r_k}) from $\mathcal{A}_1$ where $r_k \in \mathbb{Z}_q$, if (AID_k, g^{r_k}) is already in L_{H_1} , $\mathcal{B}$ responds to it by the corresponding c_{1k} . Otherwise, $\mathcal{B}$ sets $c_{1k} = c'_{1k}$ where $c'_{1k} \leftarrow \mathbb{Z}_q$, then adds (AID_k, g^{r_k}, c_{1k}) to L_{H_1} and sends c_{1k} to $\mathcal{A}_1$.

H_2 -Queries: $\mathcal{B}$ sets an initially empty hash list L_{H_2} . For a query on (AID_k, g^{y_k}) from $\mathcal{A}_1$ where $y_k \in \mathbb{Z}_q$, if (AID_k, g^{y_k}) is already in L_{H_2} , $\mathcal{B}$ responds to it by the corresponding c_{2k} . Otherwise, $\mathcal{B}$ sets $c_{2k} = c'_{2k}$ where $c'_{2k} \leftarrow \mathbb{Z}_q$, then adds (AID_k, g^{y_k}, c_{2k}) to the list L_{H_2} and sends c_{2k} to $\mathcal{A}_1$.

H_3 -Queries: $\mathcal{B}$ sets an initially empty hash list L_{H_3} . For a query on (m_k, T_{ik}) from $\mathcal{A}$, if (m_k, T_{ik}) is already in L_{H_3} , $\mathcal{B}$ responds to it by the corresponding c_{3k} . Otherwise, $\mathcal{B}$ sets $c_{3k} = c'_{3k}$ where $c'_{3k} \leftarrow \mathbb{Z}_q$, then adds (m_k, T_{ik}, c_{3k}) to the list L_{H_3} and sends c_{3k} to $\mathcal{A}_1$.

Pre-Processing Queries: $\mathcal{B}$ sets an initially empty hash list L_{pp} which contains the queries and the corresponding responses. For a pre-processing query on AID_k from $\mathcal{A}_1$, if it is already in L_{H_3} , $\mathcal{B}$ responds to it by the corresponding $(y_k, g^{y_k}, u_k, g^{r_k})$. Otherwise, $\mathcal{B}$ executes H_1 -queries and H_2 -queries to get (AID_k, g^{r_k}, c_{1k}) and (AID_k, g^{y_k}, c_{2k}) , and computes $u_k = c_{2k} s \mod q$, then adds $(AID_k, y_k, g^{y_k}, u_k, g^{r_k})$ to L_{pp} and sends the response to $\mathcal{A}_1$.

Private Key Queries: $\mathcal{B}$ sets an initially empty hash list L_{sk} . For a query on AID_k , $\mathcal{B}$ aborts and restarts to a new query for $AID_k = AID_k^*$. For $AID_k \neq AID_k^*$, $\mathcal{B}$ returns the array $(s_k, X^{-c_{1k}} g^{s_k})$ to $\mathcal{A}_1$ where $s_k \leftarrow \mathbb{Z}_q$ and $c_{1k} \in L_{H_1}$, and adds $(AID_k, s_k, X^{-c_{1k}} g^{s_k})$ to the list L_{sk} .

Partial Pre-Computation Replacement Queries: When $\mathcal{A}_1$ initiates a partial pre-computation replacement query on $(AID_k, g^{y'_k})$, $\mathcal{B}$ replaces the partial pre-computation tuple $(AID_k, y_k, g^{y_k}, u_k, g^{r_k})$ with a new $(AID_k, y_k, g^{y'_k}, u_k, g^{r'_k})$.

Signing Queries: For each query for (AID_k, m_k) made by $\mathcal{A}$ in these signing queries, if $AID_k \neq AID_k^*$, $\mathcal{B}$ gets the signature $\sigma_k = (z_k, g^{y_k}, g^{r'_k})$ by previous queries where

$z_k = y_k + u_k c_{3k} \bmod q$. Otherwise, $\mathcal{B}$ sets $R_k = X^{-c_{1k}} g^{s_k}$, $Y_k = Z_k (g^{r_k c_{2k}} X^{c_{1k}})^{-c_{3k}}$ where $s_k \leftarrow \mathbb{Z}_q$ and $Z_k \leftarrow G$. Finally it sends signature (Z_k, Y_k, R_k) to the adversary.

Forgery: The adversary $\mathcal{A}_1$ outputs a valid and forged signature σ^* where m^* and sk_{ID^*} have never been queried. According to the forking algorithm $\mathcal{F}_{PK}(\mathcal{A}_1)$ introduced in [35], $\mathcal{A}_1$ gets two valid signatures (z_1, g^y, g^r) and (z_2, g^y, g^r) with $z_1 \neq z_2$ for the distinct hash values. Thus it has

$$g^{z_1} = g^y ((g^r X^{c_1})^{c_2})^{c_3}$$

$$g^{z_2} = g^y ((g^r X^{c_1})^{c_2})^{c_3^*}$$

Let $p = (g^r X^{c_1})^{c_2}$; it has $z_1 - z_2 = (c_3 - c_3^*) \log p$. Thus $\mathcal{B}$ computes

$$(c_3 - c_3^*)^{-1} (z_1 - z_2) = \log p = c_2 (r + c_1 x)$$

and outputs $(c_1 c_2 (c_3 - c_3^*))^{-1} (z_1 - z_2) - c_1^{-1} r$ as the solution to the discrete logarithm problem instance.

We then prove the security under the type-II adversary $\mathcal{A}_2$. The adversary $\mathcal{A}_2$ can get the master secret key but keep the private key secret and can not replace the partial pre-computation of the system. Given a problem instance, we now build a simulator $\mathcal{B}$ and describe its interaction process with the adversary $\mathcal{A}_2$ as follows.

Setup: The simulator $\mathcal{B}$ chooses the public parameters $(G, q, g, X, H_1, H_2, H_3)$, then sends them and the corresponding master secret key x to $\mathcal{A}_2$.

H_1 -Queries: $\mathcal{B}$ first sets an initially empty hash list L_{H_1} . For a query on (AID_k, g^{r_k}) from $\mathcal{A}_2$ where $r_k \in \mathbb{Z}_q$, if (AID_k, g^{r_k}) is already in L_{H_1} , $\mathcal{B}$ responds to it by the corresponding c_{1k} . Otherwise, $\mathcal{B}$ sets $c_{1k} = c'_{1k}$ where $c'_{1k} \leftarrow \mathbb{Z}_q$, then adds (AID_k, g^{r_k}, c_{1k}) to the list L_{H_1} and sends c_{1k} to $\mathcal{A}_2$.

H_2 -Queries: $\mathcal{B}$ sets an initially empty hash list L_{H_2} . For a query on (AID_k, g^{y_k}) from $\mathcal{A}_2$ where $y_k \in \mathbb{Z}_q$, if (AID_k, g^{y_k}) is already in L_{H_2} , $\mathcal{B}$ responds to it by the corresponding c_{2k} . Otherwise, $\mathcal{B}$ sets $c_{2k} = c'_{2k}$ where $c'_{2k} \leftarrow \mathbb{Z}_q$, then adds (AID_k, g^{y_k}, c_{2k}) to the list L_{H_2} and sends c_{2k} to $\mathcal{A}_2$.

H_3 -Queries: $\mathcal{B}$ sets an initially empty hash list L_{H_3} . For a query on (m_k, T_{ik}) from $\mathcal{A}_2$, if (m_k, T_{ik}) is already in L_{H_3} , $\mathcal{B}$ responds to it by the corresponding c_{3k} . Otherwise, $\mathcal{B}$ sets $c_{3k} = c'_{3k}$ where $c'_{3k} \leftarrow \mathbb{Z}_q$, then adds (m_k, T_{ik}, c_{3k}) to the list L_{H_3} and sends c_{3k} to $\mathcal{A}_2$.

Private Key Queries: $\mathcal{B}$ first sets an initially empty hash list L_{sk} . For a query on AID_k , $\mathcal{B}$ aborts and restarts to a new query for $AID_k = AID_k^*$. For $AID_k \neq AID_k^*$, $\mathcal{B}$ executes a H_1 -query and gets (g^{r_k}, AID_k, c_{1k}) , then computes $s'_k = r_k + c_{1k}x \bmod q$. Set $s_k = s'_k$, $\mathcal{B}$ returns the array (s_k, g^{r_k}) to $\mathcal{A}$ and adds (AID_k, s_k, g^{r_k}) to the list L_{sk} .

Pre-Processing Queries: $\mathcal{B}$ sets an initially empty hash list L_{pp} . For a pre-processing query on AID_k from $\mathcal{A}_2$, if it is already in L_{H_3} , $\mathcal{B}$ responds to it by the corresponding $(y_k, g^{y_k}, u_k, g^{r_k})$. Otherwise, $\mathcal{B}$ executes H_2 and private key queries to get (AID_k, g^{y_k}, c_{2k}) and (AID_k, s_k, g^{r_k}) , and computes $u_k = c_{2k}(r_k + c_{1k}x) \bmod q$, then adds $(AID_k, y_k, g^{y_k}, c_{2k}(r_k + c_{1k}x) \bmod q, g^{r_k})$ to L_{pp} and sends the response to $\mathcal{A}_2$.

Signing Queries: For each query for (AID_k, m_k) made by $\mathcal{A}_2$ in this signing queries, if $AID_k \neq AID_k^*$, $\mathcal{B}$ gets the signature $\sigma_k = (z_k, g^{y_k}, g^{r_k})$ by previous queries where $z_k = y_k + c_{2k}c_{3k}(r_k + c_{1k}x) \bmod q$. Otherwise, $\mathcal{B}$ sets $R_k = X^{-c_{1k}} g^{s_k}$, $Y_k = Z_k (g^{r_k c_{2k}} X^{c_{1k}})^{-c_{3k}}$ where $s_k \leftarrow \mathbb{Z}_q$ and $Z_k \leftarrow G$. Finally it sends signature (Z_k, Y_k, R_k) to $\mathcal{A}_2$.

Forgery: The adversary $\mathcal{A}_2$ outputs a valid and forged signature σ^* where m^* and sk_{ID^*} have never been queried. According to the forking algorithm $\mathcal{F}_{PK}(\mathcal{A}_2)$ introduced in [35], $\mathcal{A}_2$ gets two valid signatures (z_1, g^y, g^r) and (z_2, g^y, g^r) with $z_1 \neq z_2$ for the distinct hash values. Thus it has $g^{z_1} = g^y ((g^r g^{xc_1})^{c_2})^{c_3}$ and $g^{z_2} = g^y ((g^r g^{xc_1})^{c_2})^{c_3^*}$. $\mathcal{B}$ then computes

$$(c_2(c_3 - c_3^*))^{-1} (z_1 - z_2) - c_1 x = r$$

and outputs it as a solution to the discrete logarithm problem instance.

In conclusion, if the adversaries $\mathcal{A}_1$ and $\mathcal{A}_2$ can break our proposed protocol, then the discrete logarithm problem will become easy. Hence our proposed protocol is secure in the EU-CMA model under the discrete logarithm assumption. $\square$

5.2. Attack Categories

We now discuss some attacks below, and some of them such as the Sybil attack have already been incorporated in our defined security model.

- **Replay Attacks:** The hash calculation for each session of our scheme is forcibly bound to the unique random number of the current session. Thus, each signature has been embedded with particular timestamps and random nonces. Even if an attacker intercepts a valid signature, since it cannot obtain the real-time random number of the new session, this signature will be directly discarded in the next communication round. Furthermore, the type-I adversary defined in our security model possess the capacity of replay attacks, and the proposed scheme has been proven to be secure against this adversary. In conclusion, any expired or repeated message–signature pair will be rejected directly during verification, thus ensuring that our scheme is secure against replay attacks.
- **Stolen OBUs and Impersonation Attacks:** Even if the OBU device is physically stolen, the adversary cannot extract the original *ID* and private keys as our *AID* generation is based on cryptography, and the OBU stores pre-computed values rather than the corresponding private key for online signature generation. In addition, once the OBU offline anomaly access is detected, the TA will trigger emergency revocation of all keys bound to this OBU hardware, and will not refresh the *AID* and private key mapping of this vehicle.
- **Pre-Computation Reuse Attacks:** The pre-computation update mechanism follows the *AID* replacement strategy. Vehicles are required to periodically change the *AID*. Upon expiration, the *AID* and the corresponding pre-computations will be discarded unless the traceability mechanism is triggered during the period. The pre-computed values with incorrect timestamps will not be adopted by the system either.
- **TA/Cloud Compromise Attacks:** If the TA is only curious and attempts to create multiple forged identities by producing new public–private key pairs, acting as the type-II adversary in our security model, our scheme is still secure. In addition, replacing pre-verified values by the TA/cloud is also invalid as it has been defined in the proposed security model. If the TA is completely compromised and can make any attacks within its capacities, relying solely on the cryptography-based approaches is completely ineffective. Practical VANETs will shift from “verifying message” to “evaluating the credibility of the message source”, and make judgments and responses relying on the trust management system of the whole network.
- **Clock Desynchronization:** A tolerant clock synchronization mechanism is designed for VANET high-speed mobile vehicles. We can set a permissible time deviation window for vehicle–RSU clock difference, and adopt relative timestamp instead of absolute timestamp for signature verification. The clock desynchronization interval depends on different VANET scenarios, ensuring that it is able to eliminate verification failure and fake authentication risks caused by network clock offset.
- **Revocation Latency:** Local abnormal vehicle revocation around the RSU can be completed within single-hop vehicular communication and recording it in a revocation list. For cross-region revocation latency, it is necessary to jointly handle it through the TA and cloud server using a consensus protocol, and define the revocation delay

boundary. We do not consider it as a contribution of this paper at present and will consider it as a future work.

- **Sybil Attacks:** In our proposed scheme, assuming an adversary or curious TA attempts to create multiple forged identities by producing new keys, acting as the defined type-II adversary in our security model, our scheme has been proven to be secure against these Sybil attacks launched by this adversary. That is to say, our proposed scheme is capable of resisting Sybil attacks under the discrete logarithm assumptions.
- **Location and Route Privacy:** In order to ensure unlinkability across both space and time, instead of binding signatures to real vehicle *ID*, we adopt short-lived *AID* generated by the TA without carrying geographic coordinates. Each *AID* is only valid within a limited geographic segment and short time window. Importantly, *AID* refresh is triggered not only by time, but also by driving distance and randomized thresholds, and *AID*-to-vehicle mapping is maintained ephemerally at the TA and purged after a short retention period. Hence, only the TA holds the trapdoor for identity and route disclosure, and this is invoked only under strictly specific circumstances, such as in the event of a dispute or detected malicious behavior.

5.3. Security Objective Analysis

In this section, we will discuss the requirements for the security objectives of our proposed VANET for V2V communication. For convenience, we first briefly discuss the assumptions about the TA and RSUs, and the *AID* refresh rules.

The TA is a trusted authority that faithfully executes key distribution, *AID* mapping, malicious identity traceability algorithms, and will not actively leak vehicle real *ID* to any third-party node. It is also equipped with a hardware security module (HSM) to securely store the private key and *AID* mapping list, which supports judicial audit and malicious vehicle tracing officially. RSUs are fixed as the roadside infrastructures that mainly provide traffic information, local *AID* issuance and revocation services. However, curious RSUs will attempt to collect message data and link *AID* to track vehicle tracks, but cannot obtain and tamper the private key and *AID* mapping list.

AID's refresh rules generally include fixed periodic refresh and event-driven emergency refresh. Because of the pre-computation mechanism of the proposed approach, we may set the fixed refresh cycle of hundreds of seconds according to different traffic scenarios. Of course, for the event-driven emergency refresh such as TA handover and risk-triggered scenarios, the refresh cycle might be relatively shorter.

We then give a more detailed discussion on authentication and data integrity, unlinkability, anonymity, and traceability below.

- **Authentication and Data Integrity:** In the above Theorem 1, we have proved our proposed scheme is secure under the discrete logarithm assumption. Thus the authentication and integrity are satisfied by our scheme since they can be reduced to the security proof of the underlying identity-based signature. As a result, a legal vehicle is able to authenticate its identity attached to the traffic-related message by the generated signatures. Any malicious adversaries without the private keys cannot forge the legal signatures and modify the integrity of the traffic-related messages. Any action of them will be detected by verifying the proposed message–signature pair. In a word, our proposed scheme satisfies authentication and data integrity, and ensures the normal operation of the VANETs.
- **Unlinkability:** It ensures that for any two messages sent by the same vehicle at different times or locations, an adversary cannot determine whether they originate from the same vehicle. In our scheme, each *AID* is used only within its validity period T_j . Upon expiration, the vehicle automatically switches to the next unused *AID* and

updates its private key. Different *AIDs* are cryptographically independent in our construction, ensuring no computable linkage between successive pseudonyms. After switching, all new messages are signed with a fresh key, and the corresponding certificate contains no long-term identity information. Suppose an adversary can observe multiple distinct *AIDs* within time window W , and $W \geq MT_j$, where M is the number of pseudonym switches during W . Ideally, the adversary's probability of correctly linking two messages from the same vehicle is $1/M$. If the time window W is relatively short and only one *AID* is observed, unlinkability will degenerate to anonymity as the adversary cannot link different messages from the same vehicle, and all messages share the same *AID*.

- **Anonymity:** Anonymity ensures that no adversaries (including RSUs or other vehicles) can associate transmitted information with the vehicle's real *ID* without accessing the TA's HSM. In our scheme, each vehicle V preloads a set of AID_j from the TA, each with a validity period T_j . For each session, V uses its currently active *AID* and corresponding private key to sign the message. The TA maintains an encrypted *AID* mapping strictly inside its HSM, inaccessible to RSUs or external entities, and any non-TA identity query is rejected. As a result, the adversary cannot recover the original *ID* from *AID* as it cannot access TA's HSM. In addition, it is assumed that the adversary can eavesdrop and capture all messages released by a target vehicle within a time window W . Given that $W \leq T_j$, the adversary only obtains messages signed by a single valid *AID*. Thus, the success probability for the adversary to map the captured *AID* to the target vehicle's real *ID* is bounded to $1/N$, where N denotes the total number of *AIDs* allocated in the whole system. This cracking probability is equivalent to blind random guessing. In conclusion, any adversarial entity, including compromised RSUs and eavesdropping neighboring vehicles, cannot link a specific vehicle to its historical communication behaviors or deduce the vehicle's permanent real *ID* in the proposed scheme.
- **Traceability:** Traceability ensures that in case of disputes, upon obtaining proper legal authorization, the TA can extract the real identity of the vehicle from any related message to enforce accountability. When a dispute arises in our scheme, RSUs send the tuple (m, σ, AID, T_i) of the disputed or malicious message to the TA in its range. Upon receiving the report, the TA queries its local HSM for the *AID*. If the vehicle has registered this *AID*, the TA then uses the corresponding secret key to extract the original *ID* of the vehicle. The TA generates a verifiable traceability proof and submits it to law enforcement authorities. Simultaneously, the TA revokes the registration of the malicious vehicle and broadcasts the revocation information across the entire VANET, ensuring that all RSUs and vehicles update their certificate revocation lists and reject future messages from this vehicle. In conclusion, our proposed scheme thus satisfies traceability for vehicular communication, enabling accountable dispute resolution while maintaining privacy during normal operations.

6. Performance Analysis

We give the performance analysis of this scheme through theoretical performance models and mobile platform experiments.

6.1. Performance Analysis Under VANET Mobility and Network Dynamics

We now give a theoretical performance evaluation based on the proposed scheme in realistic VANET environments, taking into account factors such as network delay, packet loss, mobility, and RSU availability.

- **Communication Delay and Packet:** The online signature size of our scheme is determined by $\sigma = (z, g^y)$. According to the hypotheses of 64 bytes of $|G|$, the total transmission payload is $|z| + |g^y| = 128$ bytes. This signature is significantly short in the online/offline performance. Let the total authentication delay be decomposed as $T = T_{auth} + T_{trans} + T_{queue}$. The transmission delay is given by $T_{trans} = |\sigma|/R$ where R is the data rate. Consequently, our short $|\sigma|$ directly minimizes T_{trans} . More importantly, in a lossy wireless environment with a random packet loss rate, the probability that a transmitted signature is successfully delivered without corruption is $P = (1 - BER)^L$, where BER is the bit error rate. Since our scheme minimizes $|\sigma|$, it achieves a significantly higher successful delivery probability under high packet loss scenarios.
- **Mobility:** We use Doppler shift to evaluate the vehicle mobility. Let the relative speed between the vehicle and the RSU be v , and the carrier frequency be f_c . The Doppler shift is $\Delta f = v \cdot f_c / c$, which directly increases BER at the physical layer. In our model, authentication failure is triggered either by equation rejection or packet corruption. Because our verification equation only requires two modular exponentiations and one hash operation, the computational complexity is independent of the vehicle's speed. However, the communication success rate degrades with speed. We model the effective authentication success rate as $P_{total}(v) = P_{equ} \cdot (1 - BER(v))^{|\sigma|}$. Given that $BER(v)$ increases with v , the minimal $|\sigma|$ ensures that $P_{total}(v)$ maintains a high confidence threshold even in the worst-case scenario.
- **Worst-Case RSU Unavailability:** For scenarios where the RSU is temporarily unavailable, vehicles cache the partially computed offline variables ψ along with the online signature σ . Because the size of the stored tuple is only 192 bytes plus the message payload, the storage overhead per vehicle is lightweight. Once connectivity is re-established, the RSU verifies the backlogged signatures using the same verification equation. The signature equation ensures that the temporal delay does not affect the correctness, proving our scheme's robustness against intermittent network disconnections.

In the following discussions, we will give the performance comparisons among our proposed online/offline model, the standard VANET model and state-of-the-art identity-based authentication schemes. The constructions of these schemes are all based on discrete logarithm assumptions, and the comparisons are configured with a uniform 128-bit security level. We use the elliptic curve $y^2 = x^3 + x$ and Java programming language with JPBC library-2.0.0. All the results are obtained by using a 5G mobile platform Qualcomm Snapdragon 778G with 8.0GB+2.0GB of RAM in Android12 to perform the experiment of the related schemes. Each operation was repeated 10 times with distinct random seeds to mitigate the impact of randomness, and their average values were recorded for final comparison. Each run was trained for a total of 200 iterations. We did not use a specific warm-up phase in our current setting because of the limited time and computing power. But for all quantitative results, we report the mean value together with the 95% confidence interval of each metric to ensure the reliability of the experiment.

6.2. Communication and Storage Costs

We first analyze the communication cost between the standard model and our online/offline model. The vehicle V_i first sends its ID to the TA to register the network. For the standard model of this initial phase, the TA has to send all the (s, g^r) generated by the AID to the vehicle V_i , while in our approach, s and g^r can be stored in private by the TA as they will only be used in offline signing and offline TA verify phases. It is easy to see that the communication cost of the authentication stage in the standard model is (m, z, g^y, g^r, T_i) , while in our proposed model it is segmented into offline and online communication costs.

In the offline authentication phase, the TA sends (y, u, g^y) to the vehicle V_i and p to V_j for online signing and verifying respectively. In the online authentication phase, V_i only needs to transmit (m, z, T_i) to V_j because V_j has received g^y in advance in the offline phase. That is to say, the communication cost in the online phase in our scheme is (m, z, T_i) .

According to the hypotheses of 64 bytes of $|G|$ and $|\mathbb{Z}_q|$ for elements in group G and ring $\mathbb{Z}_q$, 16 bytes of $|T|$ for timestamps, and 32 bytes of $|S|$ for random bit strings, the offline and online communication costs between the standard approach and our online/offline approach are showed in Table 4 and Figure 4. It is easy to see that our approach has a slightly larger total cost than the standard approach, but the online communication cost is smaller. Anyway, the offline communication costs cannot be ignored in practical VANETs.

We then compare our construction with the schemes [3,21,22] in terms of communication and storage costs, which are shown in Table 5 and Figure 4. In our scheme, the TA has to pre-process and store multiple values generated offline because of the proposed online/offline model. Thus it has to take slightly larger storage cost than other schemes. However, the communication cost of our scheme is smaller than that of other schemes as a few elements can be locally processed by the TA and do not need to be transmitted to vehicles in our scheme.

Table 4. Theoretical communication analysis of the standard approach and our online/offline approach.

	Standard Approach	Proposed Approach
Initial phase	$2 S + 2 \mathbb{Z}_q $	$2 S $
Offline authentication	0	$ S + 3 \mathbb{Z}_q $
Online authentication	$ S + 3 \mathbb{Z}_q + T $	$ S + \mathbb{Z}_q + T $
Total communication cost	$3 S + 5 \mathbb{Z}_q + T $	$4 S + 4 \mathbb{Z}_q + T $

Table 5. Communication and storage costs among different schemes.

The Scheme	Communication	Storage (TA)	Storage (Vehicle)
[3]	$3 G + \mathbb{Z}_q + 2 T $	$ S + \mathbb{Z}_q $	$ S + \mathbb{Z}_q $
[21]	$ G + 2 \mathbb{Z}_q + S $	$ S $	$ S + \mathbb{Z}_q $
[22]	$3 G + \mathbb{Z}_q + T + S $	$ S $	$ S + 2 \mathbb{Z}_q $
Ours	$ S + \mathbb{Z}_q + T $	$3 S + 2 \mathbb{Z}_q $	$ S + 2 \mathbb{Z}_q $

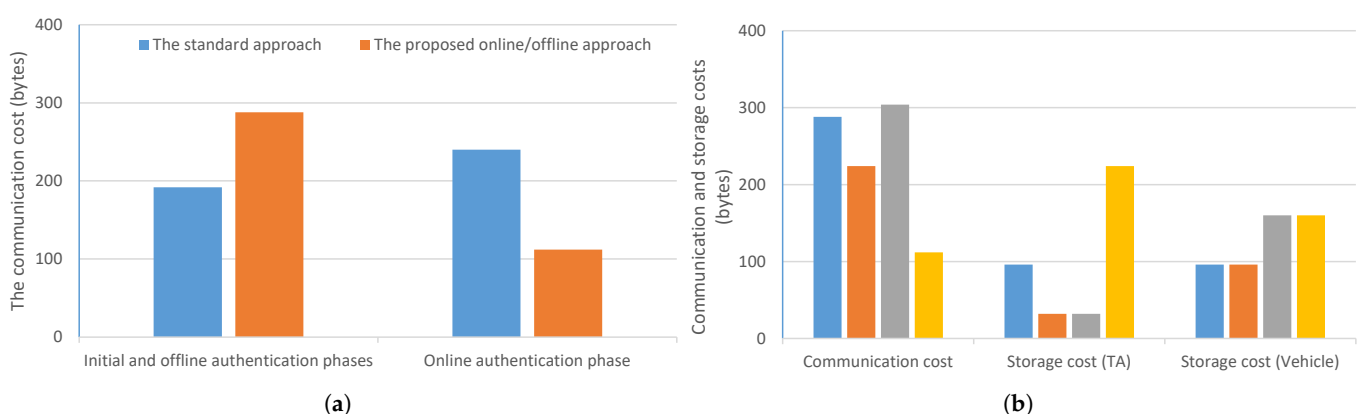


Figure 4. Comparisons of the communication and storage costs. (a) Standard and our online/offline approach. (b) The different schemes. In (b), Blue represents the scheme in [3], brown represents the scheme in [21], gray represents the scheme in [22], and yellow represents our scheme.

6.3. Computation Cost

Likewise, we first analyze the computation cost between the standard model and our online/offline model. Denoted by t_a , t_m , t_e , t_r , t_h , and t_{mm} are the times needed for performing one addition, multiplication, exponent, sampling, hashing and modular operation

respectively. Theoretical analyses of the computation cost for both approaches are shown in Table 6. In addition, our simulated experiment results of the two approaches are shown in Figure 5.

Table 6. Theoretical computation analysis of the standard approach and our online/offline approach. The online authentication is divided into online signing and online verifying.

	Standard Approach	Proposed Approach
Offline signing (TA)	0	$t_r + t_e + t_h + t_m + t_{mm}$
Online signing (V_i)	$t_h + t_a + t_m + t_{mm} + t_r + t_e$	$t_h + t_a + t_m + t_{mm}$
Offline verifying (TA or cloud)	0	$2(t_m + t_e + t_h)$
Online verifying (V_j)	$t_a + 3t_m + 3t_e + 2t_h$	$t_a + t_m + 2t_e + t_h$
Offline authentication	0	$3(t_e + t_h + t_m) + t_r + t_{mm}$
Online authentication	$4t_e + 3t_h + 4t_m + 2t_a + t_{mm} + t_r$	$2(t_e + t_h + t_m) + 2t_a + t_{mm}$
Total authentication	$4t_e + 3t_h + 4t_m + 2t_a + t_{mm} + t_r$	$5(t_e + t_h + t_m) + 2t_a + 2t_{mm} + t_r$

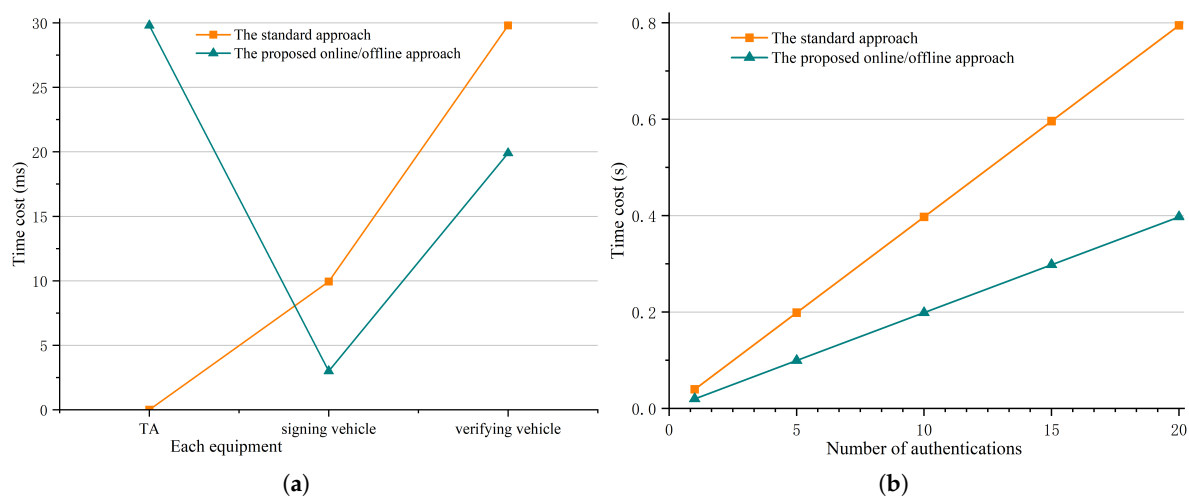


Figure 5. The performing time of the standard construction and our online/offline construction. (a) Online authentication. (b) Each equipment per session.

In our online/offline approach, V_i and V_j only take part in the online authentication phases. Besides these vehicles, the TA also plays an important role in preparing the pre-computed values in the offline phases. As Table 6 indicates, the time costs of the TA are $t_r + t_e + t_h + t_m + t_{mm}$ for offline signing and $2(t_m + t_e + t_h)$ for offline TA verifying. In addition, the offline authentication phase is performed by the TA as it includes the offline sign pre-computation and offline TA verification phases. Before the traffic-related message is given, this stage can be performed in any time by the TA. It can be seen in Table 6 that the performing time of the TA is $t_m + t_e + t_h + t_r + t_{mm} + 2(t_e + t_m + t_h)$ for offline authentication.

In the standard VANET authentication model, the signing and verifying phases are performed by the vehicles V_i and V_j respectively, which belong to the online authentication stage. Hence, the online performing time is $t_h + t_m + t_{mm} + t_r + t_a + t_e + t_a + 3t_e + 3t_m + 2t_h$, which includes the online signing cost for V_i and online verifying cost for V_j respectively. In short, compared to the standard approach, our proposed approach needs larger storage capacity for a more powerful TA, but has shorter online computation costs both for the signing vehicle V_i and verifying vehicle V_j .

We then analyze the computation time between our scheme and the related schemes. The related theoretical analyses and experimental results are shown in Table 7 and Figure 6 respectively. Our online/offline approach is performed in both the signing phase and the verifying phase. According to Figure 6, our verifying time is almost the same as the others.

This is because we choose the standard discrete logarithm problem to instantiate our model, as we would rather highlight the advantages of this framework itself. However, in the specific construction, our scheme needs to perform exponent operation in the verifying process, while other schemes are all elliptic curve-based discrete logarithms, which do not need to perform the time-consuming exponent operation. As a result, though our scheme has fewer operations in the verifying process, the verifying time is almost the same as these ECDLP-based schemes. However, the signing time of our approach is significantly shorter than those of the other schemes.

Table 7. Theoretical comparisons of computation cost among different schemes in the on-line/offline sense.

The Scheme	Signing Cost	Verifying Cost	Total Cost
[3]	$t_h + t_a + 2t_m + t_r + t_{mm}$	$t_h + 2t_a + 2t_m$	$2t_h + 3t_a + 4t_m + t_r + t_{mm}$
[21]	$4t_h + 3t_a + 4t_m + t_r$	$4t_h + 3t_a + 5t_m$	$8t_h + 6t_a + 9t_m + t_r$
[22]	$t_h + 2t_a + 2t_m + t_r$	$2t_h + 3t_a + 3t_m$	$3t_h + 2t_a + 5t_m + t_r$
Ours	$t_h + t_a + t_m + t_{mm}$	$t_h + t_a + t_m + 2t_e$	$2t_h + 2t_a + 2t_m + t_{mm} + 2t_e$

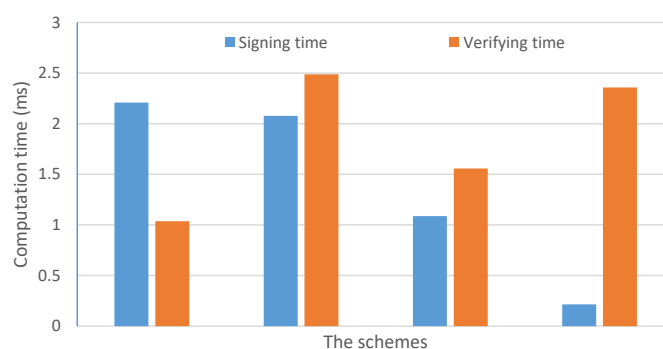


Figure 6. Computation time among different schemes. Where the first is the scheme in [3], the second is the scheme in [21], the third is the scheme in [22], and the last is our scheme.

For k signatures of the verification, the vehicle needs to perform k operations of $g^z = g^y p^{c_3}$, and this verifying equation time $T = k(2t_e + t_m) = 2.13k$ (ms). But when considering the batch verification for k signatures $\{(z_1, g^{y_1}, g^{r_1}), \dots, (z_k, g^{y_k}, g^{r_k})\}$ in valid times $T_1, \dots, T_k$, it allows a vehicle in our scheme to verify

$$g^{\sum_{i=1}^k z_i} = \prod_{i=1}^k g^{y_i} p_i^{c_{3i}}$$

where $c_{3i} = H_3(m_i, T_i)$ for $i = 1, \dots, k$. The verifying equation time $T_{batch} = (k+1)t_e + (k-1)t_m + kt_a = 1.08k + 1.05$ (ms). Hence $T_{batch} \leq T$ for $k \geq 1$. In conclusion, the batch verification is also efficient in the proposed scheme.

7. Conclusions

The standard authentication model for VANETs must account for total computational and communication overhead, regardless of the timing of authentication message generation. To mitigate this limitation, this paper proposes and formalizes a novel VANET authentication framework by shifting heavy computational overhead to the offline phase. Furthermore, this work also proposes a lightweight instantiation by our generic framework. The resulting scheme cuts vehicle-side online computation and communication overhead at the expense of increased offline overhead such as TA processing and offline data storage. As a result, offline overhead optimization might serve as a future research direction. In addition, this work discusses the elementary security of lightweight authentication for V2V

communication. In the future work, we will also analyze more security properties such as cross-domain authentication issues [36] in practical VANETs.

Author Contributions: Conceptualization, P.Z.; methodology, P.Z.; validation, P.Z.; formal analysis, P.Z. and L.W.; investigation, L.W.; writing—original draft preparation, P.Z.; writing—review and editing, L.W.; visualization, P.Z.; supervision, P.Z.; project administration, P.Z.; funding acquisition, P.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Foundation of Henan Province (262300421819).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zeadally, S.; Hunt, R.; Chen, Y.-S.; Irwin, A.; Hassan, A. Vehicular ad hoc networks (VANETs): Status, results, and challenges. *Telecommun. Syst.* **2012**, *50*, 217–241.
2. Zhou, M.; Lin, C.; Xu, S.; Wu, W. Sphinx: Certificateless conditional privacy-preserving authentication with secure transmission for VANETs. *IEEE Trans. Depend. Secur. Comput.* **2025**, *22*, 7189–7202. [[CrossRef](#)]
3. Shen, X.; Lai, J.; Han, J.; Chen, L. Lightweight conditional privacy-preserving scheme for VANET communications. *IEEE Trans. Cloud Comput.* **2025**, *13*, 1487–1497. [[CrossRef](#)]
4. Chen, S.; Guan, Y. Efficient certificateless conditional privacy-preserving authentication scheme with dynamic pseudonyms for VANETs. *IEEE Sens. J.* **2025**, *25*, 27611–27627.
5. Annabi, M.; Zeroual, A.; Messai, N. Towards zero trust security in connected vehicles: A comprehensive survey. *Comput. Secur.* **2024**, *145*, 104018. [[CrossRef](#)]
6. Ke, C.; Xiao, F.; Cao, Y.; Huang, Z. A group-vehicles oriented reputation assessment scheme for edge VANETs. *IEEE Trans. Cloud Comput.* **2024**, *12*, 859–875. [[CrossRef](#)]
7. Wei, L.; Cui, J.; Zhong, H.; Bolodurina, I.; Liu, L. A lightweight and conditional privacy-preserving authenticated key agreement scheme with multi-tamodell for fog-based VANETs. *IEEE Trans. Depend. Secur. Comput.* **2023**, *20*, 422–436. [[CrossRef](#)]
8. Li, J.; Lu, H.; Guizani, M. ACPN: A novel authentication framework with conditional privacy-preservation and non-repudiation for VANETs. *IEEE Trans. Parallel Distrib. Syst.* **2015**, *26*, 938–948. [[CrossRef](#)]
9. Tahir, H.; Mahmood, K.; Ayub, M.F.; Saleem, M.A.; Ferzund, J.; Kumar, N. Lightweight and secure multi-factor authentication scheme in VANETs. *IEEE Trans. Veh. Technol.* **2023**, *72*, 14978–14986. [[CrossRef](#)]
10. Alaya, B.; Sellami, L. Clustering method and symmetric/asymmetric cryptography scheme adapted to securing urban VANET networks. *J. Inf. Secur. Appl.* **2021**, *58*, 102779. [[CrossRef](#)]
11. Ibrahim, S.; Hamdy, M. A comparison on VANET authentication schemes: Public key vs. symmetric key. In Proceedings of the Tenth International Conference on Computer Engineering & Systems (ICCES), Cairo, Egypt, 23–24 December 2015; pp. 341–345.
12. Cao, Y.; Xu, S.; Chen, X.; He, Y.; Jiang, S. A forward-secure and efficient authentication protocol through lattice-based group signature in VANETs scenarios. *Comput. Netw.* **2022**, *214*, 109149.
13. Lin, X.; Sun, X.; Wang, X.; Zhang, C.; Ho, P.-H.; Shen, X. Tsvc: Timed efficient and secure vehicular communications with privacy preserving. *IEEE Trans. Wirel. Comm.* **2008**, *7*, 4987–4998. [[CrossRef](#)]
14. Studer, A.; Bai, F.; Bellur, B.; Perrig, A. Flexible, extensible, and efficient VANET authentication. *J. Commun. Netw.* **2009**, *11*, 574–588. [[CrossRef](#)]
15. Wang, M.; Liu, D.; Zhu, L.; Xu, Y.; Wang, F. LESPP: Lightweight and efficient strong privacy preserving authentication scheme for secure VANET communication. *Computing* **2016**, *98*, 685–708.
16. Asghar, M.; Doss, R.R.M.; Pan, L. A scalable and efficient pki based authentication protocol for VANETs. In Proceedings of the 28th International Telecommunication Networks and Applications Conference (ITNAC), Sydney, NSW, Australia, 21–23 November 2018; pp. 1–3.
17. Jiang, S.; Chen, X.; Cao, Y.; Xu, T.; He, J.; Cui, Y. APKI: An anonymous authentication scheme based on pki for VANET. In Proceedings of the 7th International Conference on Computer and Communication Systems (ICCCS), Wuhan, China, 22–25 April 2022; pp. 530–536.

18. Manivannan, D.; Moni, S.S.; Zeadally, S. Secure authentication and privacy-preserving techniques in vehicular ad-hoc networks (VANETs). *Veh. Commun.* **2020**, *25*, 100247. [\[CrossRef\]](#)
19. Lu, H.; Li, J.; Guizani, M. A novel id-based authentication framework with adaptive privacy preservation for VANETs. In Proceedings of the Computing, Communications and Applications Conference, Hong Kong, China, 11–13 January 2012; pp. 345–350.
20. Wang, Y.; Zhong, H.; Xu, Y.; Cui, J.; Wu, G. Enhanced security identity-based privacy-preserving authentication scheme supporting revocation for VANETs. *IEEE Syst. J.* **2020**, *14*, 5373–5383. [\[CrossRef\]](#)
21. Qiao, Z.; Ma, K.; Zhou, Y.; Yang, Q.; Xia, Z.; Yang, B.; Zhang, M. An anonymous and efficient certificate-based identity authentication protocol for VANET. *IEEE Internet Things J.* **2024**, *11*, 11232–11245. [\[CrossRef\]](#)
22. Zhu, D.; Guan, Y. Secure and lightweight conditional privacy preserving identity authentication scheme for VANET. *IEEE Sens. J.* **2024**, *24*, 35743–35756. [\[CrossRef\]](#)
23. Li, H.; Pei, L.; Liao, D.; Sun, G.; Xu, D. Blockchain meets vanet: An architecture for identity and location privacy protection in VANET. *Peer. Peer. Netw. Appl.* **2019**, *12*, 1178–1193. [\[CrossRef\]](#)
24. Zhou, Y.; Cao, L.; Qiao, Z.; Xia, Z.; Yang, B.; Zhang, M. An efficient identity authentication scheme with dynamic anonymity for VANETs. *IEEE Internet Things J.* **2023**, *10*, 10052–10065. [\[CrossRef\]](#)
25. Manasrah, A.; Yaseen, Q.; Al-Aqrabi, H.; Liu, L. Identity-based authentication in VANETs: A review. *IEEE Trans. Intell. Transp. Syst.* **2025**, *26*, 4260–4282. [\[CrossRef\]](#)
26. Sun, Y.; Lu, R.; Lin, X.; Shen, X.; Su, J. An efficient pseudonymous authentication scheme with strong privacy preservation for vehicular communications. *IEEE Trans. Veh. Technol.* **2010**, *59*, 3589–3603. [\[CrossRef\]](#)
27. Shim, K.-A. Cpas: An efficient conditional privacy-preserving authentication scheme for vehicular sensor networks. *IEEE Trans. Veh. Technol.* **2012**, *61*, 1874–1883. [\[CrossRef\]](#)
28. Ali, I.; Lawrence, T.; Li, F. An efficient identity-based signature scheme without bilinear pairing for vehicle-to-vehicle communication in VANETs. *J. Syst. Archit.* **2020**, *103*, 101692. [\[CrossRef\]](#)
29. Prajapat, S.; Kumar, D.; Hossain, M.S. Secure quantum aggregate signature scheme for vehicular ad-hoc networks. *Veh. Commun.* **2025**, *56*, 100974. [\[CrossRef\]](#)
30. Lai, J.; Huang, X.; He, D.; Wu, W. Provably secure online/offline identity-based signature scheme based on SM9. *Comput. J.* **2022**, *65*, 1692–1701.
31. Even, S.; Goldreich, O.; Micali, S. On-line/off-line digital signatures. *J. Cryptol.* **1996**, *9*, 35–67.
32. Albrecht, M.R.; Gama, N.; Howe, J.; Narayanan, A.K. Post-quantum online/offline signatures. In Proceedings of the Cryptographers' Track at the RSA Conference (CT-RSA), San Francisco, CA, USA, 28 April–1 May 2025; pp. 273–300.
33. Schnorr, C.P. Efficient signature generation by smart cards. *J. Cryptol.* **1991**, *4*, 161–174. [\[CrossRef\]](#)
34. Yao, A.C.-C.; Zhao, Y. Online/offline signatures for low-power devices. *IEEE Trans. Inf. Forensics Secur.* **2013**, *8*, 283–294. [\[CrossRef\]](#)
35. Boldyreva, A.; Palacio, A.; Warinschi, B. Secure proxy signature schemes for delegation of signing rights. *J. Cryptol.* **2012**, *25*, 57–115.
36. Yoshizawa, T.; Singelée, D.; Muehlberg, J.T.; Delbruel, S.; Taherkordi, A.; Hughes, D.; Preneel, B. A survey of security and privacy issues in v2x communication systems. *ACM Comput. Surv.* **2023**, *55*, 1–36. [\[CrossRef\]](#)

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.