ID-CAKE: Identity-based Cluster Authentication and Key Exchange Scheme for Message Broadcasting and Batch Verification in VANETs

Apurva K Vangujar[0000−0002−8194−4593],
Alia Umrani[0000−0003−1885−3629], and Paolo Palmieri[0000−0002−9819−4880]

University College Cork, Cork, Ireland
{a.vangujar, a.umrani, p.palmieri}@cs.ucc.ie

Abstract. Vehicle Ad Hoc Networks (VANETs) play a pivotal role in intelligent transportation systems, offering dynamic communication between vehicles, Road Side Units (RSUs), and the internet. Given the open-access nature of VANETs and the associated threats, such as impersonation and privacy violations, ensuring the security of these communications is of utmost importance. This paper presents the Identity-based Cluster Authentication and Key Exchange (ID-CAKE) scheme, a new approach to address security challenges in VANETs. The ID-CAKE scheme integrates the Cluster Consensus Identity-based Identification (CCIBI) with Zero-Knowledge (ZK) proofs and the Identity-based Multireceiver Key Exchange Mechanism (ID-mKEM) signature scheme. This integration provides robust *authorization* via CCIBI, while ID-mKEM signatures ensure message *integrity*, and guarantee both *non-repudiation* and *unforgeability* through mKEM for message broadcasting. The scheme employs a novel three-party ZK proof for batch verification using mKEM, which significantly reduces computational burdens. Our scheme also ensures *anonymity* and *unlinkability* by introducing pseudo-identities to all users in the cluster. The rigorous security proofs provided confirm the resilience of the ID-CAKE scheme against potential attacks, adhering to the different scenarios, against the hardness of the elliptic curve computational Diffie-Hellman under the random oracle model. The ID-CAKE scheme establishes a robust security framework for VANETs, and its introduction highlights potential pathways for future exploration in the realm of VANET security.

Keywords: Identity-based Identification · Key Exchange · Batch Verification · Zero-Knowledge · VANETs · Authentication Scheme · Signature Scheme.

1 Introduction

Vehicle Ad hoc Networks (VANETs), integral to the evolution of intelligent transportation systems, involve a dynamic network of vehicles, Road Side Units (RSUs), and internet servers. These networks facilitate crucial Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication, enhancing road safety through the broadcasting of safety messages [31]. Alarming statistics reveal that traffic accidents account for approximately 1.35 million deaths annually, positioning as the eighth primary cause of global fatalities. If no substantial measures are taken, road accidents are projected to become the seventh leading cause of death by 2030 [17].

Securing VANETs is essential to safeguard vehicular communication, prevent malicious interventions, and ensure road safety for all users. In VANETs, vehicles continuously communicate with each other using dedicated short-range communication protocols to update nearby vehicles about road conditions, traffic congestion, location and lane information, etc. Exchanging such information can improve traffic conditions, avoid collision and road accidents, and ensure safety. Yet, the open-access environment of VANETs network exposes them to myriad security threats, from impersonation to privacy violation [9,30]. For example, if a malicious vehicle appears on the network, it can generate a false emergency message to mislead vehicles into an unwanted condition. Security measures must ensure entity and message authentication, privacy preservation, *non-repudiation*, low overhead, traceability, and *unlinkability* to mitigate various attacks [20].

To navigate these security challenges, the VANETs domain predominantly leans on cryptographic techniques. Such methodologies, which encompass authentication schemes, Identity (ID)-based systems, key exchange, and Pseudo-Identity (PID) approaches, offer a secure framework to validate identities and messages, and ensuring the overall integrity of communication within VANETs. The concept of pseudonym-based systems was introduced by Chaum [7], allowing entities (such as individuals or groups) to communicate anonymously with multiple parties using different pseudonyms. Emerging research spotlights various authentication scheme in VANETs, providing secured communication. Most of these schemes use Key Encapsulation Mechanism (KEM) for signing the message and verifying of signature. Particularly, ID-based Cryptography (IBC) stands out by eliminating the need for certificate-based message authentication, thereby diminishing overhead and simplifying certificate management. The Batch Verification (BV) scheme is considered to be the best option to speed up the verification process. The batch signature verification based scheme authenticates multiple safety messages simultaneously using Elliptic Curve (EC) cryptography signatures.

In VANETs cluster, if a malicious vehicle appears on the network, it can broadcast a false emergency message, with the potential to mislead other vehicles into hazardous or undesired situations. This could result in traffic congestion, unnecessary detours, or even accidents. Therefore, it becomes critically essential to verify the authenticity of the message's origin, ensuring that the sending vehicle is authorized. Utilizing a cluster consensus approach for VANETs authentication and BV enhances scalability and reduces system overhead. By putting vehicles into clusters and leveraging consensus mechanisms, this proposed scheme offers streamlined authentication processes using Zero-Knowledge (ZK), boosts response times, and reinforces security measures. Furthermore, the adoption of PID within these clusters ensures *anonymity*, making it challenging for adversaries to track individual vehicles.

In this paper, we present a novel ID-based authentication scheme designed for VANETs of cluster setting, where messages are broadcasted among vehicles. Our approach uses ID-based Identification (IBI) scheme and Multi-receiver Key Exchange Mechanisms (mKEM) to ensure *authorization*, *non-repudiation*, and message *integrity*. To provide *anonymity*, each cluster member is assigned a PID. Moreover, we introduce a three-party ZK proof which ensures *authorization* and supports BV. By combining IBI, mKEM, and ZK schemes, we offer a comprehensive anonymous ID-based authentication and

BV solution, well-suited for clustered VANETs environments. This approach addresses the unique challenges posed by VANETs and provides an efficient and secure solution.

1.1 Related Work

ID-based Authentication Schemes IBC introduced by Shamir *et al.* [21], has since inspired various encryption and signature schemes, provides a certificate-free authentication framework crucial for VANETs, reducing overhead and enhancing efficiency and security. Sun *et al.* [24] proposed an IBC system using PID, ensuring vehicle privacy and traceability in VANETs. This method reduces storage and message overhead compared to EC cryptography-based Public Key Infrastructure (PKI) schemes. However, its scalability in dynamic environments needs further exploration and improvement. Bharadiya *et al.* [2] introduced an authentication with multiple levels of anonymity protocol which offers multi-level anonymity using an ID-based signature scheme along with PID and reduces message overhead compared to traditional PKI schemes. ID-based BV scheme by Tzeng *et al.* [25] ensures anonymous authentication, message integrity, privacy, and traceability. This scheme has computational cost of verification delay because the process of BV needs only a small constant number of Bilinear Pairing (BP) and point multiplication computations. The security proof seems addressing security requirements as per mentioned in the scheme.

Jenefa et al. [10] introduced a privacy-preserving scheme for vehicular communication using ID-based signature and ID-based online/offline Signature. This three-phase scheme includes registration, PID generation, and authentication, using IBS for V2R Side Unit and IBOOS for V2V communication. Although the scheme safeguards against impersonation and Sybil attacks, it requires vehicles to register within each RSU domain, and the scattered vehicular position complicates communication. In contrast, our approach focuses on forming well-defined clusters to enhance dynamic consensus communication in VANETs. Liu et al. [15] designed a privacy-preserving, ID-based auditable ring signature system for VANET communication involving four key participants and six phases. In comparison, our proposal significantly extends authentication functionalities within a cluster-based scenario. We also implement a KEM that not only fosters message *integrity* but ensures *non-repudiation* among cluster vehicles. Jiang *et al.* in [11] proposed an anonymous authentication scheme based on ID-based group signature in VANETs. The scheme uses ID-based group signature to provide authentication, DH key exchange to establish trust, and for anonymous communication, it uses advance encryption standard. Although, it achieves authentication, *anonymity*, forward secrecy, and *unlinkability*, it has a distributed architecture which is a trade-off between efficiency and delay. Kalmykov *et al.* [12] proposed ZK authentication protocol, reduce modular multiplicative operations time to minimize the disclosures of user authentication parameters while accessing the network. It reduces the authentication time and maximise security level.

Adopting IBC, we enhance authentication in Cluster Consensus IBI (CCIBI) scenarios to ensure *authorization* via ZK proof. Our method prioritizes forming distinct clusters to boost security and efficiency in VANETs.

Key Exchange Schemes Kim *et al.* [13] presented a scheme using group signatures for mutual identification and key exchange, with vehicles employing private keys for hashing and the group manager signing messages. This facilitates secure communication through ephemeral Diffie-Hellman (DH) exchanges. Palani *et al.* [18] propose V2V key exchange protocol enables vehicles to verify the time-bounded validity of certificates and *integrity* of keys. It also performs key exchange in the Random Oracle (RO) model and prove it secure using verification on Tamarin tool. S.A Chaudhary in [6] proposes a secure message exchange protocol for Internet of Vehicles (IoV) communicating with RSUs through wireless channels. The scheme uses symmetric encryption and hash functions to achieve mutual authentication, session key establishment, and message integrity among the IoV entities.

In [28], Wu *et al.* proposed a privacy-preserving mutual authentication and key exchange scheme in VANETs communication. The scheme consists of a Cloud Server (CS), RSU, and vehicles. The vehicles are authenticated using password and biological data while RSUs are authenticated using ID. The scheme utilizes EC Cryptography, XOR, and hash functions for secure communication, anonymity, unlinkability, and forward secrecy. However, the scheme does not include password update and the mutual authentication is only between CS-RSU and CS-Vehicle, whereas, RSU and vehicles are not mutually authenticated. Umrani *et al.* [26] introduced an anonymous multi-receiver signcryption scheme using mKEM and data encapsulation mechanism, applicable in VANETs. It provides secure communication with authentication, confidentiality, and *anonymity* based on EC Discrete Logarithm (ECDL) and EC Computational DH (EC-CDH) assumptions, while ensuring *unlinkability*, *non-repudiation*, and forward secrecy.

We propose Identity-based Cluster Authentication and Key Exchange (ID-CAKE) scheme where a mKEM that produces a symmetric key and a message signature guaranteeing both *non-repudiation, integrity*, and *unforgeability* among cluster vehicles and seamlessly integrates with ZK, adding an innovative touch to our approach.

Batch Verification Zhang et al. [29] proposed a scheme, which employed identity and batch verification (IBV) technique. In this scheme, RSU verifies large numbers of received messages at a time and hence reduces the verification time. It also reduces the communication overhead by employing an IBC and also ensures privacy by using different PID. However, it does not handle security attack. In [25], the improved IBV scheme has been proposed to solve the security and privacy issues in VANETs. It uses batch message verification that requires point multiplication as well as pairing operations and provides security under the RO model. However, this scheme involves a complex process of anonymous identity generation as well as message signing and verification. [27] proposed an identity-based privacy-preserving authentication scheme (SIPAR) has been proposed to support the efficient revocation of vehicles. This scheme ensures anonymity, non-repudiation, traceability, authentication, and provides resistance against modification attack, replay attack, impersonation attack. However, it does not give any information about the message or packet loss ratio.

Bayat et al. [1] proposed authentication scheme where key setup is constructed by eliminating all the repetitions of the parameters. Although there is additional factor in the verification computational cost that makes it inefficient. Similarly, Cahyadi et al.

[5] proposed CLAS scheme for VANETs that achieves mutual authentication, identity and location privacy, non-repudiation, unlinkability, and traceability and is secure against Type-I and Type-II adversary under CDHP assumption. Liu et al. [16] introduced a group key agreement and an anonymous batch authentication scheme for secure VANET communication. Cahyadi et al. [4] improved Liu et al.'s [16] scheme and proposed a symmetric key Binary Tree (BT)-based batch authentication scheme for VANETs. This scheme introduces a reputation system as an additional feature that assign a trust score to each vehicle and the signatures are verified according to the reputation score which significantly reduces the computation cost. [19] introduce Zero-Knowledge (ZK) authentication protocol for group in VANETs. It is comprised of three algorithms Key Management and OBU-groups Formation, Authentication Protocol, and Distributed Privilege Control Revoking Mechanism.

1.2 Contributions

We introduce an Identity based Cluster Authentication and Key Exchange (ID-CAKE) scheme for secure VANETs communication. Firstly, We construct CCIBI with ZK based on BLS scheme. For ID-CAKE scheme, we combine designed CCIBI scheme and ID-mKEM signature scheme. The security of the proposed scheme proves *euf-cma* security under RO model using ECCDH assumption. In the ID-CAKE scheme, we introduce a Pseudo-Identity (PID) generation algorithm that generates PID for each vehicle in the cluster thus providing identity privacy. Our scheme achieves user authentication and message *integrity* while simultaneously ensuring *traceability, unlinkability,* and *non-repudiation*. We construct the ID-CAKE scheme, a novel and compelling solution to secure VANET communication. The key contributions that make ID-CAKE an outstanding addition to the VANETs field are as follows:

1. **Cluster-Based Scenario Construction:** We pioneer a cluster-based ZK approach for VANETs authentication, an innovation that not only enhances consensus but also fortifies security within VANETs. We strategically combine the CCIBI scheme with ZK, based on the BLS scheme, and integrate the ID-mKEM signature scheme into ID-CAKE. This combination results in a versatile authentication and key exchange mechanism.
2. **Pseudo-Identity Generation:** We introduce a PID generation algorithm that fosters identity privacy within the cluster, a significant advancement in ensuring both *traceability* and *unlinkability* without compromising other security aspects. Our construction of anonymous cluster PID, coupled with the application of ZK proofs, represents a thoughtful balance between *anonymity* and the necessary transparency within the consensus process.
3. **Key Exchange and Batch Verification:** The ID-CAKE scheme uniquely allows senders to generate cluster-based signatures using mKEM. By encapsulating users' keys, it ensures stringent user *authentication* and *non-repudiation*, establishing a new benchmark in message *integrity*. Through the integration of BV using ZK proofs for identity authentication and message verification, the computational cost of the ID-CAKE scheme is significantly reduced.

4. **Robust Security Proofs:** The ID-CAKE scheme's security is rigorously proven to meet *euf-cma* standards under the RO model, utilizing the ECCDH assumption. This assures a high level of trust in the system's *integrity* and resilience against attacks. The ID-CAKE scheme's security considerations are designed for various possible scenarios in VANETs. This tailored approach ensures that the scheme is both practical and robust in real-world applications.
5. **Pathway to Future Exploration:** Alongside our concrete contributions, we also highlight potential avenues for future work, inviting further exploration and innovation in the field.

2 Preliminaries

2.1 Bilinear Pairing

Bilinear Pairing (BP) definition is adopted from [3] and possesses a set of cryptographic multiplicative groups $\mathbb{G}$ of the prime order q and pairing function e such that $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}$ must satisfy the following three properties:

1. Bilinearity satisfies $e(aP, bP) = e(P, P)^{ab}$ where, $P \in \mathbb{G}$, $a, b \in \mathbb{Z}_q^*$.
2. Non-degeneracy where pairing function for a generator P should not equal to one $e(P, P) \neq 1$.
3. BP is an efficient and computable.

2.2 Elliptic Curve Computational Diffie-Hellman

The security assumption of ECCDH is according to [8].

Definition 1. *The ECCDH assumption holds given $(P, aP, bP) \in \mathbb{G}$, where $a, b \in \mathbb{Z}_q^*$, it is computationally infeasible for any Probabilistic Polynomial-Time (PPT) algorithm to compute abP.*

2.3 IBI scheme

Definition 2. *The definition of IBI scheme is given by Kurosawa and Heng [14] has three PPT algorithms IBI =* $(\mathsf{KeyGen}, \mathsf{Extract}, \mathsf{Verification})$ *defined as follows:*

1. KeyGen. *On input 1^λ, it outputs public parameter* PP *and master secret key* msk.
2. $\mathsf{Extract}$. *It takes input as* $(\mathsf{msk}, \mathsf{ID})$ *and returns the private key* d.
3. $\mathsf{Verification}$. *In this phase, the prover* P *and the verifier* V *communicates with each other.* P *takes input as* $(\mathsf{PP}, \mathsf{ID}, \mathsf{d})$ *whereas the* V *takes input as* $(\mathsf{PP}, \mathsf{ID})$. P *and* V *communicates with each other with the help of* $(\mathsf{CMT}, \mathsf{CHA}, \mathsf{RES})$ *and gives output in boolean decision 0 (rejects) or 1 (accepts). The canonical protocol acts in four steps as: (i)* P *sends commitment* (CMT) *to* V. *(ii)* V *provides challenge* (CHA) *which is randomly chosen. (iii)* P *calculates the response* (RSP) *to* V *as per challenge. (iv)* V *verifies* $(\mathsf{param}, \mathsf{ID}, \mathsf{CMT}, \mathsf{CH}, \mathsf{RSP})$ *is Diffie Hellman (DH) tuple.*

2.4 The multi-recipient Key Exchange Mechanism (mKEM)

The notion of mKEM was first proposed by N.P Smart [22] and has a KEM like construction which takes multiple receivers.

Definition 3. *The mKEM consists of four algorithms* (Setup, KeyGen, mKEM.Encaps, mKEM.Decaps) *and given as below:*

1. Setup. *On input the security parameter* 1^{λ}*, it outputs* PP.
2. KeyGen. *Taking* PP *as input, it outputs each user's public key* pk *and private key* sk.
3. mKEM.Encaps. *On input* PP *and a set of receiver public keys* $\mathsf{pk}_{\mathsf{r_i}}$ *where* $1 \leq \mathsf{i} \leq \mathsf{t}$*, it outputs a symmetric session key* K *and an encapsulation* C *of* K*.*
4. mKEM.Decaps. *Taking* PP*, receiver's private key* $\mathsf{sk}_{\mathsf{r_i}}$*, and* C *as input, it outputs* K*. The correctness of mKEM holds if* $K = \mathsf{mKEM.Decaps}(\mathsf{PP}, \mathsf{sk}_{\mathsf{r_i}}, \mathsf{C})$.

3 Building Block for ID-CAKE Scheme

3.1 Cluster Consensus Identity-based Identification (CCIBI) Scheme

In this section, we introduce a transformation of the BLS signature scheme [3] into a BLS IBI scheme as proposed by Kurosawa and Heng's [14]. We construct the Cluster Consensus Identity-based Identification (CCIBI) scheme under the ECCDH assumption as building block for new ID-CAKE scheme. For clarity, we use $\mathsf{C_i}$ where $1 \leq \mathsf{i} \leq \mathsf{n}$ to represent the generic approach.

1. KeyGen. Trusted Authority (TA) sets keys for CM and cluster members in that cluster. The TA takes an input 1^{λ} where λ is a security parameter, choosing an elliptic curve E and a point P on E of large prime order q. $\mathbb{G}$ is the cyclic group and $H : \{0,1\}^{*} \times \mathbb{G} \rightarrow \mathbb{G}$ is the hashing function and e is BP function. TA outputs $\mathsf{PP} = (\mathbb{G}, q, E, P, \hat{e}, H)$. TA next takes the input as PP and selects a random integer $x \in \mathbb{Z}_q^{*}$ generates $\mathsf{mpk} = xP$ and $\mathsf{msk} = x$. Next, Cluster Manager CM_i of $\mathsf{C_i}$ selects a random integer $y_{\mathsf{i}} \in \mathbb{Z}_q^{*}$ and generates cluster public key $\mathsf{cpk_i} = y_{\mathsf{i}}P$ and cluster secret key $\mathsf{csk_i} = y_{\mathsf{i}}$.
2. Join. This algorithm allows new members to securely join the cluster. Assume $\mathsf{ID_{(i,j)}}$ wants to join $\mathsf{C_i}$ and selects a random integer $\hat{x}_{\mathsf{(i,j)}} \in \mathbb{Z}_q^{*}$ and generates user public key $\mathsf{upk_{(i,j)}} = \hat{x}_{\mathsf{(i,j)}}P$ and user secret key $\mathsf{usk_{(i,j)}} = \hat{x}_{\mathsf{(i,j)}}$. The cluster setting is $\mathsf{C_{IBI}} = (\mathsf{C_1}, \mathsf{C_2}, ..., \mathsf{C_i}, ..., \mathsf{C_m})$ where $1 \leq \mathsf{i} \leq \mathsf{m}$.
3. Extract. TA sets the user private key d for the cluster members. Consider $\mathsf{ID_{(i,j)}}$ from $\mathsf{C_i}$, TA takes an input $\mathsf{ID_{(i,j)}}$ and mpk. TA selects a random $\bar{x}_{\mathsf{(i,j)}} \in \mathbb{Z}_q^{*}$ and calculates $Q_{\mathsf{ID}_{(i,j)}} = H(\mathsf{ID_{(i,j)}})$. User secret key $\mathsf{d_{(i,j)}} = \bar{x}_{\mathsf{(i,j)}} Q_{\mathsf{ID_{(i,j)}}}$.[1]
4. Verify. This algorithm is the communication between a P as cluster member and a V as CM. The ZK offers batch verification of the cluster IDs as follows:

[1] The cluster representation of Join algorithm of CCIBI scheme is same as the Join algorithm of ID-CAKE scheme in Section 5. d of all the identities for all available CM can be calculated with the same technique described in Extract algorithm of CCIBI scheme.

(a) CMT. P selects a random number $r_{(\mathsf{i,j})} \in \mathbb{Z}_q^*$ for each $1 \leq \mathsf{j} \leq \mathsf{n}$ and calculates $R_{(\mathsf{i,j})} = r_{(\mathsf{i,j})} Q_{\mathsf{ID}_{(\mathsf{i,j})}}$. P sends all $R_{(\mathsf{i,j})}$ to V.
(b) CHA. The random challenge $c \in \mathbb{Z}_q^*$ is generated by V and passed it P.
(c) RES. For each $1 \leq \mathsf{j} \leq \mathsf{n}$, P calculates a response $U_{(\mathsf{i,j})} = d_{(\mathsf{i,j})}(r_{(\mathsf{i,j})} + c)$ and sends all $U_{(\mathsf{i,j})}$ to V.
(d) If the equation $e(U_{(\mathsf{i,j})}, P) = e(R_{(\mathsf{i,j})}, \mathsf{cpk_i}).e(Q_{\mathsf{ID}_{(\mathsf{i,j})}}, \mathsf{cpk_i})^c$ holds by ECCDH assumption for all j, V accepts cluster identities. If it does not hold for any j, V rejects cluster identities.

The CCIBI scheme's four PPT algorithms contribute to the establishment of a secure ID-CAKE scheme in VANETs.

3.2 Construction of ID-mKEM

This section introduces our ID-mKEM signature scheme, based on Def. 2.4. Adopting mKEM from [26], we upgraded the scheme by transforming from signcryption to signature. ID-mKEM signature scheme has four PPT algorithm described below and has n user where $\mathsf{n} = \{\mathsf{ID_s}, \{\mathsf{ID_1}, ..., \mathsf{ID_{r_i}}, ..., \mathsf{ID_{r_t}}\}\}$ where $1 \leq \mathsf{i} \leq \mathsf{t}$. Assume, a sender with anonymous $\mathsf{ID_s}$ sends an arbitrary length message m to t designated receivers denoted with anonymous $\mathsf{ID_{r_i}}$.

1. KeyGen. TA takes security parameter λ as input. It chooses cyclic group $\mathbb{G}$ of large prime order q, derived from an elliptic curve E. The TA selects a generator point $P \in \mathbb{G}$ and generates four hash functions. The first hash function is $H_0 : \{0,1\}^\ell \to \mathbb{G}$, where ℓ is a positive integer, the second hash function is $H_1 : \{0,1\}^\ell \times \mathbb{G} \times \mathbb{G} \to \mathbb{G}$, the third hash function is $H_2 : \mathbb{Z}_q^* \times \mathbb{Z}_q^* \times \mathbb{G} \times \mathbb{G} \to \{0,1\}^k$, where k denotes the plaintext box length, and the fourth hash function is $H_3 : \{0,1\}^* \times \{0,1\}^k \times \{0,1\}^* \times \{0,1\}^k \times \mathbb{G} \times \mathbb{G} \to \mathbb{Z}_q^*$. The TA outputs the public parameters $\mathsf{PP} = \{\mathbb{G}, q, E, P, H_0, H_1, H_2, H_3\}$. Next, it randomly selects $x \in \mathbb{Z}_q^*$ as the msk is kept secret, and calculates the $\mathsf{mpk} = xP$. Subsequently, the RA selects random $y \in \mathbb{Z}_q^*$ as its secret key $\mathsf{sk_{RA}}$ and calculates its public key $\mathsf{pk_{RA}} = yP$. RA computes PIDs for each user involved. Each user vehicle randomly chooses $\bar{x} \in \mathbb{Z}_q^*$ as secret key of vehicle $\mathsf{sk_v}$ and computes public key of vehicle $\mathsf{pk_v} = \bar{x}P$.
   (a) Users. Each user chooses random $\mathsf{RID} \in \{0,1\}^\ell$ and computes $R = \hat{x}P$ where $\hat{x} \in \mathbb{Z}_q^*$ is randomly chosen. Taking $(\mathsf{RID}, \hat{x})$ as input, it computes initial $\mathsf{PID_1} = \mathsf{RID} \oplus H_0(\hat{x}\mathsf{pk_{RA}})$ and sends $(\mathsf{PID_1}, R)$ to RA.
   (b) RA. Taking $(\mathsf{PID_1}, R)$ as input, the RA verifies the $\mathsf{RID} = \mathsf{PID_1} \oplus H_0(Ry)$. If $\mathsf{PID_1} = \mathsf{PID}$ holds, the RA accepts the registration request from users and sends $\mathsf{PID} = \mathsf{RID} \oplus H_0(\hat{x}\mathsf{pk_{RA}})$ to respective user.
2. $\mathsf{Extract}$. For each PID in set $\mathsf{n} = \{\mathsf{PID_s}, \{\mathsf{PID_1}, ..., \mathsf{PID_{r_i}}, ..., \mathsf{PID_{r_t}}\}\}$, the TA takes mpk as input and generates user private key $\mathsf{d} = xQ_{\mathsf{PID}}$ where $Q_{\mathsf{PID}} = H_1(\mathsf{PID} \| \mathsf{mpk})$.
3. Sign. The sender with $\mathsf{PID_s}$ and $\mathsf{sk_s}$ runs following steps to sign a message m and sends signature σ to receivers $\mathsf{PID_{r_i}}$ using $\mathsf{mKEM\text{-}Encaps}$:
   (a) Randomly chooses $r \in \mathbb{Z}_q^*$ and computes $U = rP$.
   (b) Taking $\mathsf{pk_{r_i}}$ and $Q_{\mathsf{PID_{r_i}}}$ as input, computes $Z_{1_i} = \mathsf{d_s} Q_{\mathsf{PID_{r_i}}}$ and $Z_{2_i} = \bar{x}_\mathsf{s}\mathsf{pk_{r_i}}$.

(c) Computes $\psi = Z_{1_\mathsf{i}} Z_{2_\mathsf{i}}$, $K = H_2(\psi)$, $f = H_3(m, \psi, \mathsf{PID_s}, \mathsf{PID_{r_i}}, \mathsf{pk_s}, \mathsf{pk_{r_i}})$, and $S_\mathsf{i} = r^{-1}(f + w\mathsf{d_s}\bar{x}_\mathsf{s})$ where $w = \mathrm{x}_U \bmod q$ which is the x-coordinate of U.

(d) Sets $\mathsf{C_1} = (f, S_\mathsf{i})$ and outputs $\sigma = (\mathsf{C_1}, \mathsf{K}, m)$.

4. Verify. The designated receiver with $\mathsf{PID_{r_i}}$ takes $(\mathsf{sk_{r_i}}, \mathsf{pk_s})$ as input, and runs the following phases to verify the σ:

   Phase-1 (mKEM-Decaps).

   (a) Taking $\bar{x}_{\mathsf{r_i}}$ and $\mathsf{d_{r_i}}$ as input, computes $Z_{1_\mathsf{i}} = \mathsf{d_{r_i}} Q_{\mathsf{PID_s}}$ and $Z_{2_\mathsf{i}} = \mathsf{pk_s}\bar{x}_{\mathsf{r_i}}$.

   (b) Computes $\psi = Z_{1_\mathsf{i}} Z_{2_\mathsf{i}}$ and $\mathsf{K} = H_2(\psi)$. If $\mathsf{K} = \perp$, the receiver aborts otherwise verifies the S_i as follows:

   Phase-2 (Ver).

   (a) Taking $\mathsf{C_1}, m$, and $\mathsf{pk_s}$ as input, computes $f' = H_3(m', \psi, \mathsf{PID_s}, \mathsf{PID_{r_i}}, \mathsf{pk_s}, \mathsf{pk_{r_i}})$.

   (b) If $f' = f$, verifies S_i by checking if $U = rP$ and $w' = x_U \bmod q$.If $w' = w$, the receiver will accept the m else returns $\perp$ and aborts.

For the construction of our new ID-CAKE scheme, we will incorporate the Sign and Verify algorithms from the ID-mKEM scheme.

4 Our ID-CAKE VANETs scheme

4.1 VANETs Participants and Requirements

The ID-CAKE scheme is structured around three key entities: TA, Registration Authority (RA), and clusters $(\mathsf{C_1}, \mathsf{C_2}, \ldots, \mathsf{C_i}, \ldots, \mathsf{C_m})$. Within each cluster, there is a designated sender and several receivers. Assume $\mathsf{C_i} = (\mathsf{Sender} : \mathsf{ID_{s_i}}, \mathsf{Receivers} : \mathsf{ID_{r_{(i,1)}}}, \mathsf{ID_{r_{(i,2)}}}, \ldots, \mathsf{ID_{r_{(i,j)}}}, \ldots, \mathsf{ID_{r_{(i,n)}}})$ where $1 \leq \mathsf{i} \leq \mathsf{m}$ and $1 \leq \mathsf{j} \leq \mathsf{n}$. Below is an in-depth explanation of the responsibilities of each entity. Moreover, our ID-CAKE VANETs scheme meets the criteria outlined in Table 1.

- **TA**. The TA, linked to the RA via a wired channel, serves as an administrator with greater storage and computational capabilities than the RA and vehicles. It generates keys and updates system parameters in the cluster periodically.
- **RA**. RAs, positioned along roadsides or parking zones, have key duties in the cluster: (1) offering internet to vehicles, (2) amplifying VANETs' range by relaying messages, and (3) reporting traffic updates and malicious activities. As semi-trusted entities, generates their private key $\mathsf{sk_{RA}}$ and public key $\mathsf{pk_{RA}}$, and handle identity verification using ZK proofs and PID generation.
- **Sender Vehicle**. The sender with $\mathsf{PID_{si}}$ of $\mathsf{ID_{si}}$ (for $1 \leq \mathsf{i} \leq \mathsf{m}$) signs message m using mKEM-Encaps and sends signature σ to cluster receivers.
- **Receiver Vehicles**. Cluster receivers get the signed m from an RA-approved sender. They use mKEM-Decaps to extract m. If shared secret key K mismatched, they report the sender's PID to RA.

Table 1. Requirements in VANETs

Requirement	Description	Techniques
Authorization	Verifying validity of sender ID prior to communication. In the ID-CAKE scheme, the V (RA) verifies the ID of the P (sender) using ZK. If the ID is valid, then the V authorizes the P to communicate with receivers..	Verify- ZK
Anonymity	Communicating with the users without revealing the real-identity RID. In the ID-CAKE scheme, each user is assigned a PID by the RA. The user signs the m for multiple receivers using PID and keeps RID private.	Extract-PID
Integrity	Transmit information without modification during transmission. The ID-CAKE scheme uses a collision hash functions in Sign algorithm.	Sign- Hash
Non-repudiation	To prevent message ownership denial, we use mKEM to generate a symmetric key K by utilizing sender ad receiver's sk, pk. If K is valid for the entities then communication takes place hence ensuring *non-repudiation*.	Sign, Verify- mKEM
Unforgeability	To prevent signature forgery, our scheme signs m with the sender's private key using mKEM-Encaps and receivers verify σ using the sender's public key. Since only sender knows its private key, no adversary can forge the signature ensuring *unforgeability*.	Sign -mKEM-Encaps
Unlinkability	To hide the link between signatures/messages and user identities, we use randomness in PID with PID values changing every session, attackers can't associate m with original cluster users.	Extract-PID

4.2 System Description

We integrate CCIBI and ID-mKEM schemes to offer cluster authentication and batch verification in VANETs' anonymous broadcasts. Our VANETs system has three phases:

1. **System Initialization**. The setup includes infrastructure establishment, security configuration, trust building, key distribution, and creating system parameters for a secure VANETs. KeyGen algorithm initializes keys for TA and RA in the cluster.
2. **Vehicle Joining and Registration**. New vehicles register with the cluster, generate keys, and establish secure communication with the RA. The RA then creates PIDs for all the vehicles, forming a new anonymous cluster.
3. **Message Signing and Verification**. Vehicles broadcast message m within clusters, signing with ID-mKEM for authenticity. RA verifies the sender's PID, and if valid, sends signature σ to receivers. Vehicles then authenticate the source, ensure message *integrity*, evaluate trust, and make decisions using the Verify algorithm in our ID-CAKE scheme.

4.3 Definition of ID-CAKE Scheme

The ID-CAKE scheme is built on Def. 2, the CCIBI 3.1, and the ID-mKEM 3.2 signature scheme and consists of following five PPT algorithms:

1. KeyGen. With security parameter 1^λ, the TA produces a public parameter PP. Then, the TA outputs a pair of master public and secret keys $(\mathsf{mpk}, \mathsf{msk})$. While the RA generates a pair of registry public and secret keys $(\mathsf{rpk}, \mathsf{rsk})$.
2. Join. To add a new vehicle to the cluster, the RA executes the registration protocol and assigns a PID to the vehicle. *Phase-1.* Assuming a sender vehicle with identity $\mathsf{ID}_{\mathsf{s_i}}$, it performs key setup and generates a pair of keys $(\mathsf{pk}_{\mathsf{s_i}}, \mathsf{sk}_{\mathsf{s_i}})$. The same technique is used for setting up receiver vehicles, generating $(\mathsf{pk}_{\mathsf{r}_{(i,j)}}, \mathsf{sk}_{\mathsf{r}_{(i,j)}})$. The users send their pk to RA for registration along with their real ID. *Phase-2.* The user generates an initial PID in the cluster using their ID. *Phase-3.* The RA verifies the initial PID and generates PID for all users in the cluster.
3. $\mathsf{Extract}$. The TA generates user private keys d for all vehicles and for RA d_{RA} in the cluster, using inputs $(\mathsf{PID}, \mathsf{msk})$.
4. Sign. The sender, using inputs (PID_i, $\mathsf{sk}_{\mathsf{s_i}}$, m, d), runs the $\mathsf{mKEM\text{-}Encaps}$ algorithm and generates a signature σ to send to all receivers in the cluster.
5. Verify. This algorithm has two phases, facilitating communication between the sender, RA, and receivers via ZK and mKEM. *Phase-1.* The communication between the sender $\mathsf{PID}_{\mathsf{s_i}}$ (acting as P) and the RA (acting as V) performs a ZK using $(\mathsf{CMT}, \mathsf{CHA}, \mathsf{RES})$. If P accepts, the process proceeds to the next phase. *Phase-2.* Receiver vehicles compute an encapsulation key K using the $\mathsf{mKEM\text{-}Decaps}$ algorithm. If it holds, then it verifies the Signature component with the Ver algorithm. If it is valid, the receivers accepts the original message m.

4.4 Security Models

In our ID-CAKE scheme, we give security using the RO Model under ECCDH assumption. The ID-CAKE scheme views the RA as semi-honest and focuses on potential malicious behaviour of other communication entities.

Impersonator as a Sender The ID-CAKE scheme guarantees *authorization, non-repudiation*, *unforgibility*, and *integrity* under the following scenarios:

1. Malicious TA as a sender. A malicious TA as sender $\mathsf{ID}_{\mathsf{s_i}}$ creates a pair $(\mathsf{pk}_{\mathsf{s_i}}, \mathsf{sk}_{\mathsf{s_i}})$ and attempt impersonation using new PID. The RA, however, verifies the real ID for each PID. Detecting a mismatch, which signals TA impersonation, the RA rejects the PID registration and removes it, ensuring VANETs' privacy and security.
2. Malicious RA as a sender. Malicious RA as a sender where malicious RA will not hold the PID of sender and it randomly generates the pair of $(\mathsf{pk}_{\mathsf{s_i}}, \mathsf{sk}_{\mathsf{s_i}})$. Malicious RA as sender tries to generate PID via honest RA but it does not hold RA's rpk. Hence, algorithm aborts.

Malicious TA as a RA In this scenario, ID-CAKE scheme ensure *anonymity*. Malicious TA as RA where malicious TA run the KeyGen and generates its own set of $(\mathsf{rpk}, \mathsf{rsk})$ along with d_{RA}. It can generate PIDs for each user however, the initial PID_1 is verified using rsk which cannot be verified by the malicious TA since it does not know the original rsk of RA.

Malicious Sender In ID-CAKE scheme, it ensures *non-repudiation, authorization*, and *unforgeability* using following scenarios:

1. Malicious Sender as Inter-cluster Identity. A malicious sender can pose as an honest one within a cluster and send deceptive messages, potentially causing issues like traffic jams or accidents. Senders sign each message with their $\mathsf{sk}_{\mathsf{s}_i}$, ensuring *non-repudiation*. In case of malicious actions, the RA exposes the ID of the vehicle.
2. Malicious sender as an Intra-Cluster Identity. Malicious sender generates a bogus message in the cluster. To take part in cluster communication, the outsider sender registers itself with the RA, if the RID is not verified, the sender cannot take part in communication. Moreover, in the Ver phase, the RA throws a CHA to the sender using d. If the RES is not accepted by RA, then the sender *authorization* will fail and the outsider sender will not be able to send the message.
3. Malicious Sender as an Outsider Identity. It generates a fake message by his own. Malicious sender tries to register itself with RA by getting a PID, and self generate key pairs $(\mathsf{pk}_{\mathsf{s}_i}, \mathsf{sk}_{\mathsf{s}_i})$. However, during registration, RA checks if target $\mathsf{ID}^*_{\mathsf{s}_i}$ exists within cluster, if it does not, RA will not register the malicious identity. Moreover, TA will abort the game if $\mathsf{PID}_{\mathsf{s}_i} = \mathsf{PID}^*_{\mathsf{s}_i}$ and will not provide $\mathsf{d}_{\mathsf{s}_i}$.

5 Proposed Identity-based Cluster Authentication and Key Exchange

The ID-CAKE scheme offers a unique authentication method for VANETs, integrating components from Sections 3.1 and 2.4. In ID-CAKE, the RA employs an efficient ZK proof of IBI and mKEM for verifying sender authenticity. This ensures only verified vehicles broadcast in VANET clusters, while preserving *anonymity* using a new PID generation algorithm. The scheme is visualized in Fig. 1. For security, ID-CAKE operates under the Random Oracle (RO) model and is grounded on ECCDH assumptions from the base schemes. It has three phases: system initialization, vehicle joining and registration, message broadcasting and batch verification, all supported by PPT algorithms.

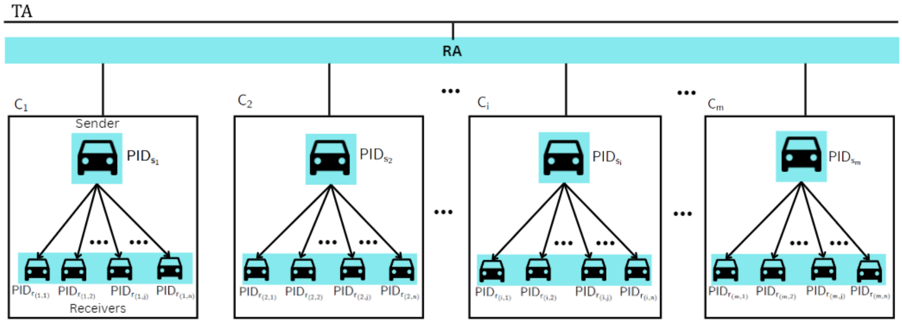
TA
RA
C1
C2
Ci
Cm
Sender
PIDs1
PIDs2
PIDsi
PIDsm
PIDr(1,1) PIDr(1,2) PIDr(1,j) PIDr(1,n)
Receivers
PIDr(2,1) PIDr(2,2) PIDr(2,j) PIDr(2,n)
PIDr(i,1) PIDr(i,2) PIDr(i,j) PIDr(i,n)
PIDr(m,1) PIDr(m,2) PIDr(m,j) PIDr(m,n)

Fig. 1. Cluster ID-based Identification and Signcryption Scheme

1. KeyGen. TA initializes the system by taking the security parameter λ as input. It selects two large prime number q and an elliptic curve E. It generates a cyclic additive $\mathbb{G}$ of a prime order q, selects a generator P of $\mathbb{G}$. TA defines four hash functions $H_0 : \{0,1\}^\ell \to \mathbb{G}$ where ℓ is a positive integer, $H_1 : \{0,1\}^\ell \times \mathbb{G} \times \mathbb{G} \to \mathbb{G}$, $H_2 : \{0,1\}^*$, and $H_3 : \{0,1\}^*$. TA chooses $x \in \mathbb{Z}_q^*$ randomly as master secret key msk and computes master public key $\mathsf{mpk} = xP$. TA generates $\mathsf{PP} = \{\mathbb{G}, q, E, P, \hat{e}, H_0, H_1, H_2, H_3\}$. Next, RA chooses $y \in \mathbb{Z}_q^*$ randomly as RA's secret key rsk and computes RA's public key $\mathsf{rpk} = yP$ and sends rpk to TA.
2. Join. The Join algorithm ensures a secure process for admitting new vehicles into the cluster. Each cluster from $\mathsf{C} = (\mathsf{C_1}, \mathsf{C_2}, ..., \mathsf{C_i}, ..., \mathsf{C_m})$ has a sender and a set of receivers given as follows:
   Phase-1. Each vehicle generates user secret key and public key pairs themeselves:
   (a) Sender. It chooses a random integer $\bar{x}_{\mathsf{s_i}} \in \mathbb{Z}_q^*$ as sender secret key $\mathsf{sk}_{\mathsf{s_i}}$ and computes sender public key $\mathsf{pk}_{\mathsf{s_i}} = \bar{x}_{\mathsf{s_i}} P$.
   (b) Receivers. Next, taking $(\mathsf{PID}_{\mathsf{r_{(i,j)}}}, \mathsf{rpk})$ as input, it chooses $\bar{x}_{\mathsf{r_{(i,j)}}} \in \mathbb{Z}_q^*$ as receiving vehicle secret key $\mathsf{sk}_{\mathsf{r_{(i,j)}}}$ and computes receiving vehicle public key $\mathsf{pk}_{\mathsf{r_{(i,j)}}} = \bar{x}_{\mathsf{r_{(i,j)}}} P$. Next, Each user send their pk along with real ID to RA.
   Phase-2. In cluster $\mathsf{C_i}$, each vehicle with ID generates the initial PID and ensures *anonymity* by following a generic approach as described below:
   (a) Sender. We consider $\mathsf{ID}_{\mathsf{s_i}}$ where $1 \leq \mathsf{i} \leq \mathsf{m}$ in the $\mathsf{C_i}$. $\mathsf{ID}_{\mathsf{s_i}}$ takes $(\mathsf{ID}_{\mathsf{s_i}}, \mathsf{rpk})$ as input and chooses a random integer $\hat{x}_{\mathsf{s_i}} \in \mathbb{Z}_q^*$ and calculates $R = \hat{x}_{\mathsf{s_i}} P$. The sender computes initial $\mathsf{PID}_{\mathsf{s_{i_1}}} = \mathsf{ID}_{\mathsf{s_i}} \oplus H_0(\hat{x}_{\mathsf{s_i}} \mathsf{rpk})$ and sends $(\mathsf{PID}_{\mathsf{s_{i_1}}}, R)$ to RA.
   (b) Receivers. We consider $\mathsf{ID}_{\mathsf{r_{(i,j)}}}$ where $1 \leq \mathsf{j} \leq \mathsf{n}$ in the $\mathsf{C_i}$. $\mathsf{ID}_{\mathsf{r_{(i,j)}}}$ takes $(\mathsf{ID}_{\mathsf{r_{(i,j)}}}, \mathsf{rpk})$ as input and chooses a random integer $\hat{x}_{\mathsf{r_{(i,j)}}} \in \mathbb{Z}_q^*$ and calculates $R = \hat{x}_{\mathsf{r_{(i,j)}}} P$. The sender computes initial $\mathsf{PID}_{\mathsf{r_{(i,j1)}}} = \mathsf{ID}_{\mathsf{r_{(i,j)}}} \oplus H_0(\hat{x}_{\mathsf{r_{(i,j)}}} \mathsf{rpk})$ and sends $(\mathsf{PID}_{\mathsf{r_{(i,j)}}}, R)$ to RA.

   Phase-3. The RA verifies the requests from all vehicles in the cluster, before issuing their PIDs.
   (a) Sender. RA takes an input $(\mathsf{PID}_{\mathsf{s_i}1}, R)$, the RA verifies $\mathsf{ID}_{\mathsf{s_i}}$ such that $\mathsf{ID}_{\mathsf{s_i}} = \mathsf{PID}_{\mathsf{s_i}1} \oplus H_0(Ry)$. It again calculates $\mathsf{PID}_{\mathsf{s_i}} = \mathsf{ID}_{\mathsf{s_i}} \oplus H_0(\hat{x}_{\mathsf{s_i}} \mathsf{rpk})$. If $\mathsf{ID}_{\mathsf{s_i}} = \mathsf{PID}_{\mathsf{s_i}1}$ holds then RA accepts the registration request and issues $\mathsf{PID}_{\mathsf{s_i}}$.
   (b) Receivers. Similarly, RA takes an input $(\mathsf{PID}_{\mathsf{r_{(i,j1)}}}, R)$, the RA verifies $\mathsf{ID}_{\mathsf{r_{(i,j)}}}$ such that $\mathsf{ID}_{\mathsf{r_{(i,j)}}} = \mathsf{PID}_{\mathsf{r_{(i,j1)}}} \oplus H_0(Ry)$. It again calculates $\mathsf{PID}_{\mathsf{r_{(i,j)}}} = \mathsf{ID}_{\mathsf{r_{(i,j)}}} \oplus H_0(\hat{x}_{\mathsf{r_{(i,j)}}} \mathsf{rpk})$. If $\mathsf{ID}_{\mathsf{r_{(i,j)}}} = \mathsf{PID}_{\mathsf{r_{(i,j1)}}}$ holds then RA accepts the registration request and issues $\mathsf{PID}_{\mathsf{r_{(i,j)}}}$. The new anonymous C is defined as:

$$
\begin{aligned}
\mathsf{C_1} &= \left(\mathsf{PID}_{\mathsf{s_1}}, \mathsf{PID}_{\mathsf{r_{(1,1)}}}, \mathsf{PID}_{\mathsf{r_{(1,2)}}}, \ldots, \mathsf{PID}_{\mathsf{r_{(1,j)}}}, \ldots, \mathsf{PID}_{\mathsf{r_{(1,n)}}}\right) \\
\mathsf{C_2} &= \left(\mathsf{PID}_{\mathsf{s_2}}, \mathsf{PID}_{\mathsf{r_{(2,1)}}}, \mathsf{PID}_{\mathsf{r_{(2,2)}}}, \ldots, \mathsf{PID}_{\mathsf{r_{(2,j)}}}, \ldots, \mathsf{PID}_{\mathsf{r_{(2,n)}}}\right) \\
&\ldots \\
\mathsf{C_i} &= \left(\mathsf{PID}_{\mathsf{s_i}}, \mathsf{PID}_{\mathsf{r_{(i,1)}}}, \mathsf{PID}_{\mathsf{r_{(i,2)}}}, \ldots, \mathsf{PID}_{\mathsf{r_{(i,j)}}}, \ldots, \mathsf{PID}_{\mathsf{r_{(i,n)}}}\right) \\
&\ldots \\
\mathsf{C_m} &= \left(\mathsf{PID}_{\mathsf{s_m}}, \mathsf{PID}_{\mathsf{r_{(m,1)}}}, \mathsf{PID}_{\mathsf{r_{(m,2)}}}, \ldots, \mathsf{PID}_{\mathsf{r_{(m,j)}}}, \ldots, \mathsf{PID}_{\mathsf{r_{(m,n)}}}\right)
\end{aligned}
$$

3. $\mathsf{Extract}$. This algorithm extracts the private key associated with a specific PID from the cluster and ensures the authenticity. The cluster member with PID computes the private key as follows:

   (a) Sender. The sender takes $(\mathsf{mpk}, \mathsf{PID}_{\mathsf{s_i}})$ as input and computes $Q_{\mathsf{PID}_{\mathsf{s_i}}} = H_1(\mathsf{PID}_{\mathsf{s_i}})$. The vehicle then computes private keys as $\mathsf{d}_{\mathsf{s_{i1}}} = xQ_{\mathsf{PID}_{\mathsf{s_i}}}$ and $\mathsf{d}_{\mathsf{s_{i2}}} = 1/x + Q_{\mathsf{PID}_{\mathsf{s_i}}}$. Sender private key is $\mathsf{d}_{\mathsf{s_i}} = (\mathsf{d}_{\mathsf{s_{i1}}}, \mathsf{d}_{\mathsf{s_{i2}}})$.

   (b) Receivers. The receiver vehicle takes $(\mathsf{mpk}, \mathsf{PID}_{\mathsf{r}_{(\mathsf{i,j})}})$ as input and computes $Q_{\mathsf{PID}_{\mathsf{r}_{(\mathsf{i,j})}}} = H_1(\mathsf{PID}_{\mathsf{r}_{(\mathsf{i,j})}})$. The vehicle then computes private keys as $\mathsf{d}_{\mathsf{r}_{(\mathsf{i,j1})}} = xQ_{\mathsf{PID}_{\mathsf{r}_{(\mathsf{i,j})}}}$ and $\mathsf{d}_{\mathsf{r}_{(\mathsf{i,j2})}} = 1/x{+}Q_{\mathsf{PID}_{\mathsf{r}_{(\mathsf{i,j})}}}$. Receivers private key $\mathsf{d}_{\mathsf{r}_{(\mathsf{i,j})}} = (\mathsf{d}_{\mathsf{r}_{(\mathsf{i,j1})}}, \mathsf{d}_{\mathsf{r}_{(\mathsf{i,j2})}})$.

   (c) RA. TA chooses $t \in \mathbb{Z}_q^*$ and calculate $\alpha = H_1(\mathsf{ID}_{\mathsf{RA}}, \mathsf{rpk}, \mathsf{mpk})$ and $\mathsf{d}_{\mathsf{RA}} = t + x\alpha$ where $\mathsf{ID}_{\mathsf{RA}}$ is an identity of RA.

4. Sign. We consider cluster $\mathsf{C_i}$, the sender with $\mathsf{PID}_{\mathsf{s_i}}$ and secret key $\mathsf{sk}_{\mathsf{s_i}}$ runs following steps to sign a message m and sends signature σ to receivers from the cluster $\mathsf{C_i}$ with $(\mathsf{PID}_{\mathsf{r}_{(\mathsf{i,j})}}, \mathsf{pk}_{\mathsf{r}_{(\mathsf{i,j})}})$ where $1 \leq \mathsf{i} \leq \mathsf{m}$ and $1 \leq \mathsf{j} \leq \mathsf{n}$ using $\mathsf{mKEM\text{-}Encaps}$:

   (a) $\mathsf{PID}_{\mathsf{s_i}}$ randomly chooses $r \in \mathbb{Z}_q^*$ and computes $U = rP$.

   (b) Taking $(\mathsf{pk}_{\mathsf{r}_{(\mathsf{i,j})}}, Q_{\mathsf{PID}_{\mathsf{r}_{(\mathsf{i,j})}}})$ as input, computes $Z_{1_\mathsf{i}} = \mathsf{d}_{\mathsf{s_{i1}}} Q_{\mathsf{PID}_{\mathsf{r}_{(\mathsf{i,j})}}}$ and $Z_{2_\mathsf{i}} = \bar{x}_{\mathsf{s_i}} \mathsf{pk}_{\mathsf{r}_{(\mathsf{i,j})}}$.

   (c) Computes $\psi = Z_{1_\mathsf{i}} Z_{2_\mathsf{i}}$, $K = H_2(\psi)$, $f = H_3(m, \psi, \mathsf{pk}_{\mathsf{s_i}}, \mathsf{pk}_{\mathsf{r}_{(\mathsf{i,j})}})$, and $S_\mathsf{i} = r^{-1}(f + w\mathsf{d}_{\mathsf{s_i}} \bar{x}_{\mathsf{s_i}})$ where $w = \mathrm{x}_U \mathrm{mod} q$ which is the x-coordinate of U.

   (d) Sets ciphertext $ct = (f, S_\mathsf{i})$ and outputs $\sigma = (ct, m)$. K will be separately send at the time of signature verification.

5. Verify. The algorithm has two phases: In the first, the sender vehicle's identity is authenticated using a ZK proof with RA. In the second, receivers use the $\mathsf{mKEM\text{-}Decaps}$ algorithm to verify signatures and retrieve the message. This protocol ensures secure communication between sender, RA, and receivers in the cluster.
   Correctness Proof. RA calculates and accepts if the following equation holds for each i:

$$
\begin{aligned}
e(\mathsf{X}_{\mathsf{s_i}}, P) &= e(\mathsf{V}_{\mathsf{s_i}}, \mathsf{pk}_{\mathsf{s_i}}).e(Q_{\mathsf{PID_{si}}}, \mathsf{pk}_{\mathsf{s_i}})^c \\
e((\hat{r} + c)\bar{x}_{\mathsf{s_i}} Q_{\mathsf{PID}_{\mathsf{s_i}}}, P) &= e(\hat{r} Q_{\mathsf{PID_{si}}}, \bar{x}_{\mathsf{s_i}} P).e(Q_{\mathsf{PID_{si}}}, \bar{x}_{\mathsf{s_i}} P)^c \\
e((\hat{r} + c)\bar{x}_{\mathsf{s_i}} Q_{\mathsf{PID}_{\mathsf{s_i}}}, P) &= e((\hat{r} + c)\bar{x}_{\mathsf{s_i}} Q_{\mathsf{PID}_{\mathsf{s_i}}}, P)
\end{aligned}
$$

   Receiver accepts the message after signature S_i verification by proving $U = rP$ if the correctness holds:
   Let $u_1 = f.P$ and $u_2 = w.\mathsf{pk}_{\mathsf{s_i}}.Z_{1_\mathsf{i}}.Q^{-1}_{\mathsf{PID}_{\mathsf{r}_{(\mathsf{i,j})}}}$ $U = S_\mathsf{i}^{-1}(u_1 + u_2) = S_\mathsf{i}^{-1}(f.P + w.\mathsf{pk}_{\mathsf{s_i}}.Z_{1_\mathsf{i}}.Q^{-1}_{\mathsf{PID}_{\mathsf{r}_{(\mathsf{i,j})}}}) = S_\mathsf{i}^{-1}(f.P + w.\mathsf{pk}_{\mathsf{s_i}}.\mathsf{d}_{\mathsf{s_i}}.Q_{\mathsf{PID}_{\mathsf{r}_{(\mathsf{i,j})}}} Q^{-1}_{\mathsf{PID}_{\mathsf{r}_{(\mathsf{i,j})}}}) = S_\mathsf{i}^{-1}(f.P + w.x_{\mathsf{s_i}}.P.\mathsf{d}_{\mathsf{s_i}}) = \frac{f.P + w.x_{\mathsf{s_i}}.P.\mathsf{d}_{\mathsf{s_i}}}{S_\mathsf{i}} = \frac{P(f + w.x_{\mathsf{s_i}}.\mathsf{d}_{\mathsf{s_i}})}{r^{-1}(f + w.x_{\mathsf{s_i}}.\mathsf{d}_{\mathsf{s_i}})} = \frac{P}{r^{-1}} = r.P$

Identity Identification and Signature Verification using ZK Proof for VANETs

Sender		**RA**		**Receivers**
$(\mathsf{pk}_{\mathsf{s_i}}, \mathsf{PID}_{\mathsf{s_i}}, \mathsf{d}_{\mathsf{s_i}}, \sigma)$				$(\bar{x}_{\mathsf{r}_{(i,j)}}, \mathsf{d}_{\mathsf{r}_{(i,j)}}, \mathsf{pk}_{\mathsf{s_i}})$
Select random $\hat{r} \in \mathbb{Z}_q^*$				
Calculates $\mathsf{V}_{\mathsf{s_i}} = \hat{r}.Q_{\mathsf{PID}_{\mathsf{s_i}}}$				
	$\xrightarrow{(\mathsf{V}_{\mathsf{s_i}}, \sigma)}$			
	$\xleftarrow{c}$	$c \in \mathbb{Z}_q^*$		
$\mathsf{X}_{\mathsf{s_i}} = \hat{r} + c.\mathsf{d}_{\mathsf{s_i}} \bmod q$	$\xrightarrow{\mathsf{X}_{\mathsf{s_i}}}$	Verify by 5	$\xrightarrow[\text{If accepts}]{(\mathsf{PID}_{\mathsf{s_i}}, \sigma)}$	Computes $Z_{1_i} = \mathsf{d}_{\mathsf{r}_{(i,j)}} Q_{\mathsf{PID}_{\mathsf{s_i}}}$
				$Z_{2_i} = \mathsf{pk}_{\mathsf{s_i}} \bar{x}_{\mathsf{r}_{(i,j)}}$
				$\psi = Z_{1_i} Z_{2_i}$, $K' = H_2(\psi)$.
				$K = K'$accepts[2]
				$f' = H_3(m, \psi, \mathsf{pk}_{\mathsf{s_i}}, \mathsf{pk}_{\mathsf{r}_{(i,j)}})$
				$w' = x_U \bmod q$
				$f = f', w = w'$
				Verifies S_i
				If valid it accepts m
			$\xleftarrow[\text{Report } (\mathsf{PID}_{\mathsf{s_i}})]{\sigma}$	If not valid $\perp$

2

6 Security Analysis

The security proof of the ID-CAKE schemes is described using ECCDH assumption using security model defined in Section 4.4.

Theorem 1. *The ID-CAKE scheme is secure against impersonation in the RO model if the ECCDH assumption holds. Impersonator I cannot distinguish the ECCDH assumption on a shared secret from a random element in $\mathbb{G}$ with a non-negligible advantage ϵ to ensure unforgeability, integrity, authorization, and non-repudiation.*

[2] If $K = K'$, then receivers verify S_i. To further verify S_i, the receivers compute f' and w'. If $f = f'$ and $w = w'$, then receiver accepts the m; otherwise, it aborts and reports the corresponding $\mathsf{PID}_{\mathsf{s_i}}$ along with σ to the RA.

6.1 Malicious TA as a Sender

Proof. According to the Def. 1, the challenger $\mathbb{C}$ interacts with the simulator S to ensure *unforgeability*, *integrity*, *authorization*, and *non-repudiation* by solving ECCDH as follows:

1. KeyGen. $\mathbb{C}$ generates $\mathsf{PP} = (\mathbb{G}, q, E, P, \hat{e}, H_0, H_1, H_2, H_3)$ by giving input 1^λ and passes PP. Again, it takes $\mathsf{mpk} = \theta P$ by choosing a random integer θ and passes mpk to I. I selects a $\mathsf{ID}^*_{\mathsf{s_i}}$ as a target sender identity.
2. **Training Phase.** In the training phase, I aims to learn from the sender's responses. $\mathbb{C}$ maintains sender's responses in list of hash queries oracle $\{L_0, ..., L_3\}$. $\mathbb{C}$ maintains the list L_{pk} to store public and secret parameters. I can issue a series of q queries which are polynomially bounded.
   Case 1. $\mathsf{ID_{si}} = \mathsf{ID}^*_{\mathsf{si}}$ where $\mathsf{ID}^*_{\mathsf{si}}$ is a targeted sender.
   (a) Join. I sends $\mathsf{ID_{s_i}}$ to $\mathbb{C}$ to get $(\mathsf{pk_{s_i}}, \mathsf{sk_{s_i}})$. $\mathbb{C}$ checks if $\mathsf{ID_s} = \mathsf{ID}^*_{\mathsf{s}}$. If yes, the $\mathbb{C}$ aborts. The system ensures both *unlinkability* and *anonymity*, as evident by the Join algorithm, which aborts the creation of a new $\mathsf{PID_{si}}^*$, thus maintaining the privacy of participants.
   (b) $\mathsf{Extract}$. Upon receiving the H_1 query, if $\mathsf{PID_{si}} = \mathsf{PID}^*_{\mathsf{si}}$, the $\mathbb{C}$ aborts. The $\mathbb{C}$ will still calculate $\mathsf{d_{RA}}$ for RA.
   (c) Sign. Upon receiving the H_2 query, $\mathsf{PID_{si}} = \mathsf{PID}^*_{\mathsf{si}}$, the $\mathbb{C}$ aborts.
   (d) Verify. When transcript will create even if not yet queried before as an $\mathsf{Extract}$ query. $\mathsf{PID_{s_i}}$ as P participates in transcript and adds in the set. RA will not be able to issue transcript for the already malicious sender. Hence, upon receiving H_3, receivers in the cluster will not get $(\mathsf{PID_{s_i}}, \sigma)$ and game aborts. $\mathsf{PID_{s_i}}$ is targeted ID and RA needs to verify it. $\mathsf{PID_{s_i}} = \mathsf{PID}^*_{\mathsf{s_i}}$, I acts as the cheater P, RA as the V, and $\mathbb{C}$ does not have user secret key of $\mathsf{PID}^*_{\mathsf{s_i}}$, however it needs to create it again to run ZK. When I tries to forge $\mathsf{PID}^*_{\mathsf{s_i}}$ then he should know $\mathsf{sk_{s_i}}$ and Verify aborts here We can perform transcript as many times as number of queries does not exceed.

   Case 2. $\mathsf{ID_{si}} \neq \mathsf{ID}^*_{\mathsf{si}}$ is a targeted sender.
   (a) Join. Given $\mathsf{ID_{si}} \neq \mathsf{ID}^*_{\mathsf{s_i}}$, the I aims to participate as a cluster member by generating a queries $(q_{\mathsf{pk}}, q_{\mathsf{sk}})$ and passes to $\mathbb{C}$. $\mathbb{C}$ randomly chooses γ as $\mathsf{sk}^*_{\mathsf{s_i}}$ and computes $\mathsf{pk}^*_{\mathsf{s}} = \gamma P$. $\mathbb{C}$ sends the $(\mathsf{pk}^*_{\mathsf{s_i}}, \mathsf{sk}^*_{\mathsf{s_i}})$ to I and updates L_{pk}. I then attempts to extract PID from RA. When the I sends a H_0 query, $\mathbb{C}$ checks if $(\mathsf{PID}^*_{\mathsf{s_{i1}}}, \mathsf{ID}^*_{\mathsf{s}}, R)$ is already listed in L_0. If found, $\mathbb{C}$ provides $\mathsf{PID}^*_{\mathsf{s_i}}$ to I. Otherwise, the $\mathbb{C}$ computes $\mathsf{PID}^*_{\mathsf{s_i}} = \mathsf{ID}^*_{\mathsf{s_i}} \oplus H_0(\hat{x}_{\mathsf{s_i}}\mathsf{rpk})$ and sends $\mathsf{PID}^*_{\mathsf{s_i}}$ to I and updates L_0.
   (b) $\mathsf{Extract}$. Upon receiving $(q_{\mathsf{d_{si1}}}, q_{\mathsf{d_{si2}}})$ queries, if it exists in L_{pk}, the $\mathbb{C}$ returns it to I. Otherwise, it computes $Q^*_{\mathsf{PID_{s_i}}} = H_1(\mathsf{PID}^*_{\mathsf{s_i}})$ and updates L_1. $\mathbb{C}$ selects a random integer β_1 and returns $\mathsf{d}^*_{\mathsf{s_{i1}}} = \beta_1 Q^*_{\mathsf{PID_{s_i}}}$ and using θ, $\mathsf{d}^*_{\mathsf{si2}} = \frac{1}{\theta + Q^*_{\mathsf{PID_{s_i}}}}$ and sends to I. Also $\mathbb{C}$ updates L_{pk} $(\mathsf{PID}^*_{\mathsf{s_i}}, \mathsf{d}^*_{\mathsf{si1}}, \mathsf{d}^*_{\mathsf{si2}}, \mathsf{pk}^*_{\mathsf{s_i}}, \mathsf{sk}^*_{\mathsf{s_i}})$.
   (c) Sign. Upon receiving $\mathsf{Sign.}$ query $q_{\mathsf{Sign.}}$, the $\mathbb{C}$ performs normal $\mathsf{Sign.}$ operation as defined in Section 5. It fetches the list L_2 to get ψ, L_3 to get f and L_{pk} to get values of $(U, Z_{1_\mathsf{i}}, Z_{2_\mathsf{i}}, S_\mathsf{i}, ct, \sigma)$ and passes the signature to I.

(d) Verify. If $\mathsf{PID_{s_i}} = \mathsf{PID^*_{s_i}}$, the $\mathbb{C}$ takes values $(\mathsf{pk_{s_i}}, \mathsf{PID_{s_i}}, \mathsf{d_{s_i}}, \sigma)$ from L_{pk} and performs regular Verify ZK proof between malicious sender and RA and then passes $(\mathsf{PID_{s_i}}, \sigma)$ to all the receivers in the clusters.

3. **Challenge.** The I takes targeted $\mathsf{PID^*_{si}}$ chooses target plaintext m^* and forged $ct^* = (f^*, S_{\mathsf{i}^*})$ along with $\sigma^* = (f^*, ct^*, m^*)$ which is the valid signature and is not the result of Sign oracle. I sends it to the $\mathbb{C}$. Moreover, the I can not ask for the $\mathsf{sk_{s_i}}$. Also, I generates $(\mathsf{V}^*_{s_i}, \mathsf{X^*_{s_i}})$, updates in L_3, and passes to $\mathbb{C}$ which is a RA. $\mathbb{C}$ selects β_2 and returns $(\mathsf{V^*_{s_i}}, \mathsf{X^*_{s_i}})$. $\mathbb{C}$ verifies $e(\mathsf{X^*_{s_i}}, P) = e(\mathsf{V^*_{s_i}}, \mathsf{pk^*_{s_i}}).e(Q^*_{\mathsf{PID_{s_i}}}, \mathsf{pk^*_{s_i}})^{c^*}$ and $\mathbb{C}$ aborts and σ will be sent further to receivers.
4. **Breaking Phase.** This phase where I acts as a cheating V and tries to convince $\mathbb{C}$ based on information gathered in the Training Phase. I wins the game if it is successful in convincing the $\mathbb{C}$ to accept with non-negligible probability. Taking the target sender $\mathsf{PID^*_{s_i}}$ and designated receiver's $\mathsf{PID_{r_{(i,j)}}}$ in the cluster, I outputs a forged $ct^* = (f^*, S_{\mathsf{i}^*})$ along with σ^* on m^* where $\sigma^* = (f^*, ct^*, m^*)$ which is the valid signature and is not the result of Sign oracle. Moreover, $\mathsf{PID^*_{s_i}}$ I outputs malicious values $(\mathsf{V^*_{s_i}}, \mathsf{X^*_{s_i}})$ which is the valid transcript for ZK and not the result of Verify oracle.

   The $\mathbb{C}$ extracts the list L_{pk} for the record $(\mathsf{PID^*_{s_i}}, \mathsf{d^*_{s_{i1}}}, \mathsf{d^*_{s_{i2}}}, \mathsf{pk^*_{s_i}}, \mathsf{sk^*_{s_i}})$ and L_3 for the record $(m^*, \psi^*, f^*, \mathsf{V^*_{s_i}}, \mathsf{X^*_{s_i}})$. If $\mathsf{PID_{si}} = \mathsf{PID^*_{si}}$, the $\mathbb{C}$ takes $\mathsf{mpk} = \theta P$, and fetches L_{pk} to extract $\mathsf{d^*_{s_{i1}}} = \beta_1 Q^*_{\mathsf{PID_{si}}}$. The $\mathbb{C}$ will win by obtaining $\theta\beta_1 P$ which is the solution to the ECCDH assumption by evaluating $\frac{\theta.Z_{1_i} - \mathsf{d_{r_{(i,j)}}}.r}{(\mathsf{d_{s_{i1}}} - U)} = \theta\beta_1 P$.

   the $\mathbb{C}$ takes $\mathsf{pk^*_{s_i}} = \gamma P$, and fetches L_{pk} to extract $\mathsf{d^*_{s_{i2}}} = 1/\theta + Q^*_{\mathsf{PID_{si}}}$. The $\mathbb{C}$ will win by obtaining $\theta\gamma\beta_2 P$ which is the solution to the ECCDH assumption. We consider $Q^*_{\mathsf{PID_{si}}} = bP$, using $\mathsf{d^*_{s_{i2}}}$ we can calculate $\mathsf{d^*_{s_{i2}}} \cdot (xP + bP) = P$. If and I can compute $\mathsf{d^*_{s_{i2}}} \cdot bP$, call this value as θ_1, then I can compute $\theta_1 \cdot \gamma = \mathsf{d^*_{s_{i2}}} \cdot bP \cdot \gamma = \mathsf{d^*_{s_{i2}}} \cdot \gamma bP$. Hence, $\mathbb{C}$ will win by obtaining $\mathsf{d^*_{s_{i2}}} \cdot \gamma bP$ which is a solution to ECCDH assumption.

   For probability distribution to prove *zero-knowledgeness* for $\mathbb{C}$, it is winning the game after solving the ECCDH assumption. Event A denotes the success of solving the ECCDH assumption a, while event B denotes not aborting the calculations. Joint probability $\mathrm{P}(A|B)$ joint probability represents the conditional probability of event A occurring given that event B has occurred.

   The $\mathbb{C}$ is able to find $\theta\beta_1 P$ and $\mathsf{d^*_{s_{i2}}} \cdot \gamma bP$ which is the solution to the ECCDH assumption. Next, we will analyse the advantage of the $\mathbb{C}$ in winning the game. The $\mathcal{C}$ advantage is based on the occurrence of the events in which the game aborts. The $\mathbb{C}$ aborts the game under the following conditions:

   - The secret key q_{sk} query where the game aborts for $\mathsf{ID_{s_i}} = \mathsf{ID^*_{s_i}}$. The probability is $\Pr(q_{\mathsf{sk}}) = 1/q_{\mathsf{sk}}$. The game aborts when if it guess right $\mathsf{ID^*_{s_i}}$, then probability of game stopping any random guess is 1 out of q_{sk}.
   - In $\mathsf{Extract}$, there is a query $\mathsf{d_{s_{i1}}}$ which query separately for $q_{\mathsf{d_{s_{i1}}}}$ and $q_{\mathsf{d_{s_{i2}}}}$. The probability of game abort is $\Pr(q_{\mathsf{d_{s_{i1}}}}) = 1/q_{\mathsf{d_{s_{i1}}}}$ and $\Pr(q_{\mathsf{d_{s_{i2}}}}) = 1/q_{\mathsf{d_{s_{i2}}}}$.
   - Sign query where game due to fake m. The probability of aborting game is q_{Sign} aborts $\Pr(1/2^k)$ where 2^k is message space.
   - $\mathbb{C}$ in the challenge phase aborts the game if I queries for $\mathsf{PID_{s_i}} \neq \mathsf{PID^*_{s_i}}$. The probability of aborting is $\Pr(q_{H_3}) = (1 - 1/q_{H_3})$.

Next, the $\mathbb{C}$ takes the L_2 to fetch ψ and L_3 to fetch $(\mathsf{V}_{\mathsf{s_i}}, \mathsf{X}_{\mathsf{s_i}}, f)$ and calculates $\theta\beta_1 P$ and $\mathsf{d}_{\mathsf{si2}} \cdot \gamma b P$ having independent probability $(1/q_{H_2}, 1/q_{H_3})$. $\mathbb{C}$ winning the game with calculated inverse of each abort probability advantage ϵ' as follows:

$$\epsilon' \geq \epsilon \left(\frac{1}{q_{H_2}}\right)\left(\frac{1}{q_{H_3}}\right)\left(\frac{1}{q_{\mathsf{d}_{\mathsf{s_i}}}}\right)\left(1-\frac{1}{q_{\mathsf{sk}}}\right)\left(1-\frac{q_{\mathsf{Sign}}}{2^k}\right) \tag{1}$$

6.2 Malicious TA as a RA

Theorem 2. *The ID-CAKE scheme is secure against an Impersonator I (malicious TA) under the RO based on the hardness of the ECCDH assumption. I cannot distinguish the ECCDH secret from a random element in $\mathbb{G}$ with a non-negligible advantage ϵ to ensure anonymity and unlinkability.*

Proof. According to the Def. 1, the challenger $\mathbb{C}$ interacts with the simulator S to ensure *anonymity* and *unlinkability* as follows:

1. KeyGen. $\mathbb{C}$ runs this algorithm to generate $\mathsf{PP} = \{\mathbb{G}, q, E, P, \hat{e}, H_0, H_1, H_2, H_3\}$. It takes x as msk randomly and computes $\mathsf{mpk} = xP$. The $\mathbb{C}$ sends mpk to I. I selects $\mathsf{ID}^*_{\mathrm{RA}}$ as a target identity. To generate $(\mathsf{rpk}, \mathsf{rsk})$, the $\mathbb{C}$ checks if $\mathsf{ID}_{\mathrm{RA}} = \mathsf{ID}^*_{\mathrm{RA}}$. If yes, it aborts otherwise $\mathbb{C}$ chooses θ_2 as rsk^*, computes $\mathsf{rpk}^* = \theta_2 P$ and sends rpk^* to I.
2. **Training Phase**. In the this stage, I seeks to understand the RA's response. $\mathbb{C}$ keeps a record of the RA's responses in the form of a list of hash query oracles, denoted as $\{L_0, ..., L_3\}$. Additionally, $\mathbb{C}$ has a list, L_{pk}, where it stores both public and secret parameters. I has the capability to make a sequence of queries, denoted by q, which are limited by a polynomial bound.
   Case 1. $\mathsf{ID}_{\mathrm{RA}} = \mathsf{ID}^*_{\mathrm{RA}}$ is a targeted registration authority.
   (a) Join. During *Phase-1*, every user within the cluster creates a pair $(\mathsf{pk}, \mathsf{sk})$. In *Phase-2*, each user generates their initial $\mathsf{PID}_{\mathsf{i_1}} = \mathsf{ID_i} \oplus H_0(\hat{x}_{\mathsf{i}}\,\mathsf{rpk})$ and forwards it to I for validation and the formation of PID. In *Phase-3*, if $\mathsf{ID}_{\mathrm{RA}} = \mathsf{ID}^*_{\mathrm{RA}}$, the game aborts since RA does not hold the original $\mathsf{rsk} = y$ required to validate the $\mathsf{PID}_{\mathsf{i_1}}$.
   (b) $\mathsf{Extract}$. Upon receiving $(q_{\mathsf{d_{si1}}}, q_{\mathsf{d_{si2}}}, q_{\mathsf{d_{RA}}})$ queries, if it exists in L_{pk}, the $\mathbb{C}$ returns it to I. Otherwise $\mathbb{C}$ will calculate $\mathsf{d}_{\mathsf{s_i}}$ for senders and $\mathsf{d}_{\mathsf{r}_{(\mathsf{i,j})}}$ for receivers in the cluster. For $\mathsf{d_{RA}}$, $\mathbb{C}$ checks if $\mathsf{ID}_{\mathrm{RA}} = \mathsf{ID}^*_{\mathrm{RA}}$, if yes, the $\mathbb{C}$ aborts due to the dependency involved with the original TA.
   (c) Sign. The $\mathbb{C}$ performs normal Sign algorithmic mentioned in Section 5. It takes the list $\{L_2, L_3, L_{\mathsf{pk}}\}$ to get values $(\phi, f, U, Z_{1_{\mathsf{i}}}, Z_{2_{\mathsf{i}}}, S_{\mathsf{i}}, ct, \sigma)$ and passes tuples to I.
   (d) Verify. The transcript will not create as it has been queried $(q_{\mathsf{d_{si1}}}, q_{\mathsf{d_{si2}}}, q_{\mathsf{d_{RA}}})$ before in $\mathsf{Extract}$. Thus, when the receivers in the cluster receive H_3, they won't obtain $(\mathsf{PID}_{\mathsf{s_i}}, \sigma)$, leading to the termination of the game.
   Case 2. $\mathsf{ID}_{\mathrm{RA}} \neq \mathsf{ID}^*_{\mathrm{RA}}$.

(a) Join. The users participate as a cluster member by generating a queries $(q_{\mathsf{pk}}, q_{\mathsf{sk}})$ and passes to $\mathbb{C}$. $\mathbb{C}$ checks if $(q_{\mathsf{pk}}, q_{\mathsf{sk}})$ exists in L_{pk} otherwise, computes as in 5 and sends them to users. Users generate their initial PIDs by fetching the list L_0 and send them to RA for validation and PID generation. Now, I aims to participate as RA and tries to verify the initial PID for users by issuing the $(q_{\mathsf{rsk}}, q_{\mathsf{rpk}})$ to $\mathbb{C}$. $\mathbb{C}$ takes θ_2 as rsk^* and computes $\mathsf{rpk}^* = \theta_2 P$ and sends it to RA. RA verifies initial PID_1 and computes $\mathsf{PID}^* = \mathsf{ID}^*_{s_i} \oplus H_0(\theta_2 \mathsf{rpk}^*)$ for each user by taking the value from L_0 and sends them to each user in the cluster.

(b) $\mathsf{Extract}$. Upon receiving $(q_{\mathsf{d_{si1}}}, q_{\mathsf{d_{si2}}}, q_{\mathsf{d}_{\mathrm{RA}}})$ queries, if it exists in L_{pk}, it returns $(\mathsf{d}^*_{\mathsf{si1}}, \mathsf{d}^*_{\mathsf{si2}})$ for all the users in the clusters. Otherwise, $\mathbb{C}$ computes $\mathsf{d}^*_{\mathsf{r}_{(i,j)}} = \gamma_2 Q^*_{\mathsf{PID}_{\mathsf{r}_{(i,j)}}}$ where γ_2 is randomly chosen and $(\mathsf{d}^*_{\mathsf{si1}}, \mathsf{d}^*_{\mathsf{si2}})$ as normal operation. Next, $\mathbb{C}$ chooses β_2 randomly and computes $\mathsf{d}^*{}_{\mathrm{RA}} = \beta_2 + \theta_2.\alpha^*$ by choosing the value of α^* from L_1 as $(\alpha^*, \mathsf{ID}^*_{\mathsf{RA}}, \mathsf{rpk}^*, \mathsf{mpk}^*)$.

(c) Sign. The Sign algorithm as carried out same as outlined in Section 5.

(d) Verify The $\mathbb{C}$ takes values $(\mathsf{pk}^*_{\mathsf{s_i}}, \mathsf{PID}^*_{\mathsf{s_i}}, \mathsf{d}^*_{\mathsf{s_i}}, \sigma^*)$ from L_{pk} and chooses a random γ_1 and run $(\mathsf{CMT}, \mathsf{CHA}, \mathsf{RES})$ to calculate values of $\mathsf{V}^*_{\mathsf{s_i}} = \gamma_1 Q^*_{\mathsf{PID}_{\mathsf{s_i}}}$ and generates c^* and sender generates $\mathsf{X}^*_{\mathsf{s_i}} = \gamma_1 + c^*.\mathsf{d}^*_{\mathsf{s_i}} \bmod q$ and then $\mathbb{C}$ verifies the equation $e(\mathsf{X}^*_{\mathsf{s_i}}, P) = e(\mathsf{V}^*_{\mathsf{s_i}}, \mathsf{pk}^*_{\mathsf{s_i}}).e(Q^*_{\mathsf{PID}_{\mathsf{s_i}}}, \mathsf{pk}^*_{\mathsf{s_i}})^{c^*}$. Receivers will take values from L_{pk} and also from L_2 and L_3. If this equation holds then I sends $(\mathsf{PID}_{\mathsf{s_i}}, \sigma^*)$ to receivers from the cluster. Then, receivers will accept σ^* otherwise rejects σ^*.

(e) **Challenge.** The I takes forged $\mathsf{PID}^*_{\mathsf{si}}$ and σ^* and passes to receivers.

(f) **Breaking Phase.** $\mathbb{C}$ takes $(\mathsf{sk}^*_{\mathsf{r}_{(i,j)}}, \mathsf{pk}^*_{\mathsf{r}_{(i,j)}}, \mathsf{d}^*_{\mathsf{r}_{(i,j)}})$ and calculates $Z^*_{1_{\mathsf{i}}} = \mathsf{d}^*_{\mathsf{r}_{(i,j)}}.Q^*_{\mathsf{PID}_{\mathsf{s_i}}}$. It takes $\mathsf{rpk}^* = \theta_2 P$, msk, $Q^*_{\mathsf{PID}_{\mathsf{r}_{(i,j)}}}$, $\mathsf{d}^*_{\mathsf{s_i}}$ from L_{pk} and computes $\frac{Z^*_{1_{\mathsf{i}}}.\mathsf{msk}.\mathsf{rpk}^*}{\mathsf{d}^*_{\mathsf{s_i}}.Q^*_{\mathsf{PID}_{\mathsf{r}_{(i,j)}}}} = \theta_2 \gamma_2 P$ which is the solution to the ECCDH assumption.

We will analyse the advantage of the $\mathbb{C}$ in winning the game. The $\mathcal{C}$ advantage is based on the occurrence of the events in which the game aborts. The $\mathbb{C}$ aborts the game under the following conditions:

- The secret key q_{rsk} query where the game aborts for $\mathsf{ID}_{\mathrm{RA}} = \mathsf{ID}^*_{\mathsf{RA}}$. The probability is $\Pr(q_{\mathsf{rsk}}) = 1/q_{\mathsf{rsk}}$.
- In $\mathsf{Extract}$, there is a query $q_{\mathsf{d}_{\mathrm{RA}}}$, the probability of game abort is $\Pr(q_{\mathsf{d}_{\mathrm{RA}}}) = 1/q_{\mathsf{d}_{\mathrm{RA}}}$
- For the $\mathsf{Extract}$, $\mathbb{C}$ make a query q_{H_1}, the probability of aborting game is $\Pr(q_{H_1}) = 1/q_{H_1}$.
- In Verify, the receiver aborts if $Q_{\mathsf{PID}_{\mathsf{s_i}}} \neq Q^*_{\mathsf{PID}_{\mathsf{s_i}}}$ with the probability $\Pr(q_{H_0}) = 1/q_{H_0}$.

$\mathbb{C}$ winning the game with calculated inverse of each abort probability advantage ϵ' as follows:

$$\epsilon' \geq \epsilon \left(\frac{1}{q_{H_0}}\right)\left(\frac{1}{q_{H_1}}\right)\left(\frac{1}{q_{\mathsf{d}_{\mathrm{RA}}}}\right)\left(1 - \frac{1}{q_{\mathsf{rsk}}}\right) \tag{2}$$

7 Efficiency Analysis

In this Section, we look at how other similar VANETs authentication schemes work and compare them to the ID-CAKE scheme. We want to see how fast and efficient ID-CAKE is. Since there are not many schemes like ID-CAKE for VANETs, we also look at another common techniques. Table 2 contrasts computational and communication costs for Sign and Verify algorithms against current authentication and BV schemes. In the Anonymous Authentication Scheme (AAAS) by [11], vehicles sign a message $< f_v^i, Exp_{f_v^i}, TS_4, N_8 >$ for authentication and compute signature σ, using a random number. This message, along with signature is sent to RSU, which verifies it using three BP operations. The AAAS's communication cost is 304 bytes, with the complexity of both signing $\mathcal{O}(n) + \mathcal{O}(k^2) + \mathcal{O}(k^c) = \mathcal{O}(k^c)$ and verification being $\mathcal{O}(k^c)$, where k is the number of exponent bits and $c > 1$.

Wang et al. [27] discuss the computation and communication costs using the "MNT159" asymmetric group $\mathbb{G}_1$ with a 159-bit base field, emphasizing its efficiency in BV. We inferred that batch verifying n signatures scales as $\mathcal{O}(n \times k^{c-1})$ compared to a single signature verification of $\mathcal{O}(k^c)$, totaling 84 bytes. The byte cost in our table also accounts for the hash, which was overlooked in the original paper.

Paper	E	A	M	H	BP	Complexity	Security	Communication Cost	Byte
Jiang *et al.* [11]	6	0	6	2	3	$\mathcal{O}(k^c)$	SVO	$2\lvert\mathbb{G}_1\rvert + \lvert\mathbb{Z}_q^*\rvert + \lvert TS\rvert + \lvert Exp\rvert$	304
Wang *et al.* [27]	4	1	4	1	3	$\mathcal{O}(n \times k^{c-1})$	HSM	$3\lvert\mathbb{Z}_q^*\rvert + 2\lvert Exp\rvert + 1\lvert H\rvert$	90
Liu *et al.* [16]	0	1	2	1	6	$\mathcal{O}(k)$	Informal	$3\lvert\mathbb{Z}_q^*\rvert + 1\lvert\mathbb{G}\rvert + 1\lvert T\rvert$	92
Zhang *et al.* [29]	0	1	2	2	3	$\mathcal{O}(k)$	Informal	$4\lvert\mathbb{Z}_q^*\rvert + 2\lvert\mathbb{G}\rvert + 2\lvert H\rvert$	209
ID-CAKE	0	2	2	2	1	$\mathcal{O}(k)$	RO	$2\lvert\mathbb{Z}_q^*\rvert + 2\lvert H\rvert$	80

Table 2. Comparison of the Computation Cost, Security, and Communication Costs

Legends: k number of bits in exponent, c is greater than one, n is the number of signatures being batch verified. E is Exponentiation in $\mathbb{Z}_q^*$, A is Addition in $\mathbb{Z}_q^*$, M is a Multiplication $\mathbb{Z}_q^*$, P is BP operation, and H is hash operations.

Requirements	Jiang *et al.* [11]	Wang *et al.* [27]	Liu *et al.*[16]	Zhang *et al.* [29]	Our ID-CAKE
Authorization	✓	✓	×	×	✓
Anonymity	✓	✓	✓	✓	✓
Integrity	×	×	×	✓	✓
Non-repudiation	✓	✓	×	×	✓
Unforgeability	×	×	×	✓	✓
Unlinkability	✓	×	×	✓	✓

Table 3. Comparison of the Security Requirements

Liu et al. [16] exhibits a linear computing cost, $\mathcal{O}(k)$, which scales with group size, as shown in Table 2. While providing a generic security proof, the scheme does not clarify its assumption model. For communication, the scheme costs amount to 94 bytes, $\mathbb{G} = 256$ bits, $\mathbb{Z}_q^* = 160$ bits, and $T = 2.6$ seconds for each user. Therefore, the the size would be $2 \times 256 + 3 \times 160 + 2.6 = 94$ bytes. The message size is not standardized as it varies with shared user information. Zhang *et al.* [29] has a total transmission overhead of $21 + 125n$ for the BLS and $21 + 42n$ for ID-based BV. The overhead scales linearly with the number of receivers, leading to $\mathcal{O}(k)$ complexity. Our ID-CAKE scheme has the least computation costs and communication cost for n signature in Sign is $\mathcal{O}(k) + \mathcal{O}(k) + \mathcal{O}(k) = \mathcal{O}(k)$ and Verify is $\mathcal{O}(k)$, overall combined cost is $\mathcal{O}(k)$, the estimated size would be $2 \times 20 + 2 \times 20 = 80$ bytes. In Table 5, we present a comparative analysis of the security requirements between our scheme and existing authentication schemes. Our proposed scheme successfully achieves all security requirements, with higher efficiency with lower computational cost.

Wang *et al.* [27] gives computation cost and communication cost. To be specific, they adopt "MNT159" with degree 6 as the asymmetric group $\mathbb{G}_1$ which has a 159-bit base field size and they consider that "MNT159" has a shorter presentation for group elements and is more efficient in batch verification. We estimated if verifying one signature is $\mathcal{O}(k^c)$, then batch verifying m signatures efficient by $\mathcal{O}(m \times k^{c-1})$.

Liu *et al.* [16] requires computation cost that requires total of six pairing, two multiplication, one hash and addition operations for signature generation and verification. The scheme has linear computing cost which linearly increases with the group size shown as $\mathcal{O}(k)$ in Table 4. Moreover, the scheme defines a generic security proof and does not specify any assumption model. Zhang *et al.* in [29] provides the computation cost of the scheme which include three bilinear pairing, two hash and two multiplication and an addition operation for signature and verification of a single user. Similar to the Liu *et al.*, this scheme provides an informal and generic security proof.

In this table, we are comparing computational cost with existing authentication and batch verification schemes for signature and verification phase. Legends: k number of bits in exponent, c is greater than one, m is the number of signatures being batch verified. E is Exponentiation in $\mathbb{Z}_q^*$, A is Addition in $\mathbb{Z}_q^*$, M is a Multiplication $\mathbb{Z}_q^*$, P is pairing operation, and H is hash operations.

Paper	E	A	M	H	BP	Complexity	Security	Communication Cost	Byte
Jiang *et al.* [11]	6	0	6	2	3	$\mathcal{O}(k^c)$	SVO logic	$2\lvert\mathbb{G}_1\rvert + \lvert\mathbb{Z}_q^*\rvert + \lvert TS\rvert + \lvert Exp\rvert$	304
Wang *et al.* [27]	4	1	4	1	3	$\mathcal{O}(m \times k^{c-1})$	HSM	$3\lvert\mathbb{Z}_q^*\rvert + 2\lvert Exp\rvert + 1\lvert H\rvert$	90
Liu *et al.* [16]	0	1	2	1	6	$\mathcal{O}(k)$	Informal	$3\lvert\mathbb{Z}_q^*\rvert + 1\lvert\mathbb{G}\rvert + 1\lvert T\rvert$	92
Zhang *et al.* [29]	0	1	2	2	3	$\mathcal{O}(k)$	Informal	$1\lvert ID\rvert + 1\lvert\sigma\rvert\ 2\lvert\mathbb{Z}_q^*\rvert + \lvert\mathbb{G}\rvert$	63
ID-CAKE									

Table 4. Comparison of the Computation Cost, Security Proof, and Communication Costs

In Table 5, we present a comparative analysis of the security requirements between our scheme and existing authentication schemes. The comparison parameters include *authorization*, *anonymity*, *integrity*, *non-repudiation*, *unforgeability*, and *unlinkability*.

Requirements	Jiang *et al.* [11]	Wang *et al.* [27]	Liu *et al.*[16]	Zhang *et al.* [29]	ID-CAKE
Authorization	✓	✓	×	×	✓
Anonymity	✓	✓	✓	✓	✓
Integrity	×	×	×	✓	✓
Non-repudiation	✓	✓	×	×	✓
Unforgeability	×	×	×	✓	✓
Unlinkability	✓	×	×	✓	✓

Table 5. Comparison of the Security Requirements

Our proposed scheme successfully achieves all security requirements as shown in Table 4, offering superior efficiency with lower computational costs, setting it apart from the others.

Communication cost refers to the total size of message transmitted. According to [23], for type A pairing with respect to 80 bit security level, the size of p is equal to 64 bytes, A point on the group of points $E(F_q)$ consists of x and y coordinates. This means that the size of each element in $\mathbb{G}_1$ is $64 \times 2 = 128$ bytes whilst that of each element in $\mathbb{G}_2$ is $20 \times 2 = 40$ bytes. In addition, the size for a general hash H function in $\mathbb{Z}_q^*$, a expiration, and a timestamp are considered to be 20 bytes, 4 bytes, and 4 bytes, respectively. As the basic configuration information is the same for above schemes, we ignore the size of message and only take into account the size of the signature on the message with the corresponding $\mathsf{PID}_{\mathsf{s_i}}$. The communication cost of AAAS scheme is $2|\mathbb{G}_1|+|\mathbb{Z}_q^*|+|TS|+|Exp|$ which is 304 bytes and complexity for sign algorithm is $O(n)+O(k^2)+O(k^c) = O(k^c)$ and verify also has same complexity so overall complexity is $O(k^c)$ where k is number of bits in exponent and c is greater than 1 based on BP operation.
[27] give communication cost septerly and we combine it for our comparison $(2+1)\mathbb{Z}_q^* + (1+1)|Exp| + 1H$

Liu *et al.*'s [16] scheme has $3|\mathbb{Z}_q^*|+1|\mathbb{G}|+1|T|$ as communication cost with $\mathbb{G} = 256$ bits, $\mathbb{Z}_q^* = 160$ bits, and $T = 2.6$ seconds for each user. Therefore, the the size would be $2 \times 256 + 3 \times 160 + 2.6 = 94$ bytes. Further, we do not take the size of the message as it could vary according to the amount of information shared among the users. Zhang *et al.*'s [29] scheme has sends a signature and an identity to the receiver as a message which provides the $|ID| + |\sigma|$ as an overhead. The scheme specifies 42 bytes for ID and 21 bytes for the original message as a signature therefore $21 + 42 = 63$ bytes is the message overhead where the overhead increases linearly with the number of receivers resulting in the $\mathcal{O}(k)$ complexity.

8 Conclusion and Future Work

In the face of rising security concerns in VANETs, this paper introduced the innovative ID-CAKE scheme under ECCDH assumption. Using CCIBI with ZK proof and ID-mKEM, ID-CAKE prove a robust mechanism for VANETs authentication, ensuring identity *authorization*, *anonymity*, *integrity*, *unforgeability*, *non-repudiation*, and *unlinkability*. Additionally, its novelty in generating cluster-based signatures via the mKEM approach, coupled with efficient batch verification through ZK proofs, underscores its potential in reducing computational burdens. ID-CAKE scheme is proven secure under RO model for different senarios in VANETS. The ID-CAKE scheme bolsters VANETs' security, blending *anonymity* with consensus transparency. Its integration with ITS can further elevate security. Future research should focus on optimizing real-time performance and exploring lattice-based post-quantum cryptography.

References

1. Bayat, M., Barmshoory, M., Rahimi, M., Aref, M.R.: A secure authentication scheme for vanets with batch verification. Wireless networks **21**, 1733–1743 (2015)
2. Bhavesh, N.B., Maity, S., Hansdah, R.C.: A protocol for authentication with multiple levels of anonymity (amla) in vanets. In: 2013 27th international conference on advanced information networking and applications workshops. pp. 462–469. IEEE (2013)
3. Boneh, D., Franklin, M.: Identity-based encryption from the weil pairing. In: 21st Intl. Cryptology Conf. (CRYPTO 2001), Proceedings. pp. 213–229. LNCS, Springer (2001)
4. Cahyadi, E.F., Hwang, M.: A lightweight bt-based authentication scheme for illegal signatures identification in vanets. IEEE Access **10**, 133869–133882 (2022). https://doi.org/10.1109/ACCESS.2022.3232301, `https://doi.org/10.1109/ACCESS.2022.3232301`
5. Cahyadi, E.F., Su, T., Yang, C.C., Hwang, M.: A certificateless aggregate signature scheme for security and privacy protection in VANET. Int. J. Distributed Sens. Networks **18**(5), 155013292210806 (2022). https://doi.org/10.1177/15501329221080658, `https://doi.org/10.1177/15501329221080658`
6. Chaudhry, S.A.: Designing an efficient and secure message exchange protocol for internet of vehicles. Security and Communication Networks **2021**, 1–9 (2021)
7. Chaum, D.: Security without identification: Transaction systems to make big brother obsolete. Communications of the ACM **28**(10), 1030–1044 (1985)
8. Cohen, H., Frey, G., Avanzi, R., Doche, C., Lange, T., Nguyen, K., Vercauteren, F.: Handbook of elliptic and hyperelliptic curve cryptography. CRC press (2005)
9. Gong, Z., Gao, T., Guo, N.: PCAS: cryptanalysis and improvement of pairing-free certificateless aggregate signature scheme with conditional privacy-preserving for vanets. Ad Hoc Networks **144**, 103134 (2023)
10. Jenefa, J., Anita, E.A.M.: Secure vehicular communication using ID based signature scheme. Wirel. Pers. Commun. **98**(1), 1383–1411 (2018). https://doi.org/10.1007/s11277-017-4923-7, `https://doi.org/10.1007/s11277-017-4923-7`
11. Jiang, Y., Ge, S., Shen, X.: Aaas: An anonymous authentication scheme based on group signature in vanets. IEEE Access **8**, 98986–98998 (2020)
12. Kalmykov, I.A., Olenev, A.A., Kalmykova, N.I., Dukhovnyj, D.V.: Using adaptive zero-knowledge authentication protocol in vanet automotive network. Information **14**(1), 27 (2022)

13. Kim, D., Choi, J., Jung, S.: Mutual identification and key exchange scheme in secure vanets based on group signature. In: 2010 7th IEEE Consumer Communications and Networking Conference. pp. 1–2. IEEE (2010)
14. Kurosawa, K., Heng, S.H.: From digital signature to id-based identification/signature. In: International Workshop on Public Key Cryptography. pp. 248–261. Springer (2004)
15. Liu, F., Wang, Q.: IBRS: an efficient identity-based batch verification scheme for vanets based on ring signature. In: 2019 IEEE Vehicular Networking Conference, VNC 2019, Los Angeles, CA, USA, December 4-6, 2019. pp. 1–8. IEEE (2019). https://doi.org/10.1109/VNC48660.2019.9062800, https://doi.org/10.1109/VNC48660.2019.9062800
16. Liu, L., Wang, Y., Zhang, J., Yang, Q.: A secure and efficient group key agreement scheme for vanet. Sensors **19**(3), 482 (2019)
17. Organization, W.H.: Global status report on road safety: Time for action. Retrieved from https://www.afro.who.int/publications/global-status-report-road-safety-time-action (2021)
18. Palaniswamy, B., Camtepe, S., Foo, E., Simpson, L., Baee, M.A.R., Pieprzyk, J.: Continuous authentication for vanet. Vehicular Communications **25**, 100255 (2020)
19. Rasheed, A.A., Mahapatra, R.N., Hamza-Lup, F.G.: Adaptive group-based zero knowledge proof-authentication protocol in vehicular ad hoc networks. IEEE Trans. on Intelligent Transportation Systems **21**(2), 867–881 (2019)
20. Raya, M., Hubaux, J.P.: The security of vehicular ad hoc networks. In: Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks. pp. 11–21 (2005)
21. Shamir, A.: Identity-based cryptosystems and signature schemes. In: 4th Intl. Cryptology Conf.e (CRYPTO 1984), Proceedings. LNCS, vol. 196, pp. 47–53. Springer (1984)
22. Smart, N.P.: Efficient key encapsulation to multiple parties. In: 4th Security in Communication Networks Conf. (SCN 2004), Proceedings. pp. 208–219. LNCS, Springer (2004)
23. Standard, I.B.C.: 1: Supersingular curve implementations of the bf and bb1 cryptosystems
24. Sun, J., Zhang, C., Zhang, Y., Fang, Y.: An identity-based security system for user privacy in vehicular ad hoc networks. IEEE Trans. on Parallel and Distributed Systems **21**(9), 1227–1239 (2010)
25. Tzeng, S.F., Horng, S.J., Li, T., Wang, X., Huang, P.H., Khan, M.K.: Enhancing security and privacy for identity-based batch verification scheme in vanets. IEEE Trans. on Vehicular Technology **66**(4), 3235–3248 (2015)
26. Umrani, A., Vangujar, A.K., Palmieri, P.: An anonymous multi-receiver certificateless hybrid signcryption (amclhs) using mkem-dem for broadcast communication. Cryptology ePrint Archive, Paper 2023/780 (2023)
27. Wang, Y., Zhong, H., Xu, Y., Cui, J., Wu, G.: Enhanced security identity-based privacy-preserving authentication scheme supporting revocation for vanets. IEEE Systems Journal **14**(4), 5373–5383 (2020)
28. Wu, T.Y., Lee, Z., Yang, L., Chen, C.M.: A provably secure authentication and key exchange protocol in vehicular ad hoc networks. Security and Communication Networks **2021**, 1–17 (2021)
29. Zhang, C., Lu, R., Lin, X., Ho, P.H., Shen, X.: An efficient identity-based batch verification scheme for vehicular sensor networks. In: IEEE INFOCOM 2008-The 27th Conference on Computer Communications. pp. 246–250. IEEE (2008)
30. Zhou, Y., Wang, Z., Qiao, Z., Yang, B., Zhang, M.: An efficient and provably secure identity authentication scheme for vanet. IEEE Internet of Things Journal (2023)
31. Zhu, F., Yi, X., Abuadbba, A., Khalil, I., Huang, X., Xu, F.: A security-enhanced certificateless conditional privacy-preserving authentication scheme for vehicular ad hoc networks. IEEE Trans. on Intelligent Transportation Systems (2023)