

ALI: Anonymous Lightweight Inter-Vehicle Broadcast Authentication with Encryption

Author

Rezazadeh Baei, MA, Simpson, L, Boyen, X, Foo, E, Pieprzyk, J

Published

2022

Journal Title

IEEE Transactions on Dependable and Secure Computing

Version

Accepted Manuscript (AM)

DOI

[10.1109/TDSC.2022.3164436](https://doi.org/10.1109/TDSC.2022.3164436)

Rights statement

© 2022 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Downloaded from

<http://hdl.handle.net/10072/416009>

Griffith Research Online

<https://research-repository.griffith.edu.au>

ALI: Anonymous Lightweight Inter-Vehicle Broadcast Authentication with Encryption

Mir Ali Rezazadeh Bae , *Senior Member, IEEE*
 Leonie Simpson , Xavier Boyen, Ernest Foo , *Member, IEEE*, and Josef Pieprzyk 

Abstract—Wireless broadcast transmission enables Inter-vehicle or Vehicle-to-Vehicle (V2V) communication among nearby vehicles. This communication supports latency-critical applications for improved safety and maybe optimized traffic. However, V2V communication is vulnerable to cyber attacks involving message manipulation. Mechanisms are required to ensure both authenticity and integrity of broadcast data, while maintaining drivers privacy against surveillance. Considering the limited computational resources of vehicles and the possibility of high traffic density scenarios, authentication processes should have low computational overhead. Prior research has produced multiple authentication protocol proposals based on digital signatures, hash functions, or Message Authentication Codes (MACs). To date, there is no computationally efficient secure broadcast authentication scheme tolerable by the vehicles resource-constrained On-Board Units (OBUs) for latency-critical applications in heavy traffic conditions. This paper provides a new secure, efficient, and privacy-preserving scheme proposing Anonymous Lightweight Inter-vehicle (ALI) broadcast authentication with encryption. ALI provides a high level of anonymity by combining a message authentication scheme with beacon encryption. The cryptographic overhead for V2V communication in the ALI scheme is only 149 bytes, and can handle authentication of approximately 700 broadcast messages every 100 milliseconds (ms) on a 2.10 Gigahertz (GHz) Intel Core 2 Duo Processor. This demonstrates the suitability of the ALI scheme in heavy traffic scenarios. We show the security and efficiency of our proposal by conducting both a formal security proof and extensive performance analysis.

Index Terms—Cryptography, authentication, security, privacy, scalability.



1 INTRODUCTION

Cooperative Intelligent Transport System (C-ITS) is an emerging technology with the potential to improve road safety [1]. Car manufacturers embed devices such as IEEE 802.11p in vehicles to enable wireless communication with other vehicles and with nearby fixed equipment, referred to as Road Side Units (RSUs). This enables devices (vehicles and RSUs) within transmission range to establish a self-organizing network called a Vehicular Ad-hoc Network (VANET) [2]. The VANET exchanges beacon messages at a high update rate, carrying the identification and context information for a vehicle such as vehicle location, speed, braking status, traffic conditions, and traffic events. It is important that the beacon messages reach all the vehicles in a VANET. For this reason, beacons are broadcast

(entity authentication) and authentication of the message (assurance of data origin and data integrity). Further, for accountability, we need to ensure the sender of a message cannot deny the message transmission [3], [4].

Authentication requirements can often conflict with privacy requirements. A unique identifier, such as a Vehicle Identification Number (VIN), is provided to a vehicle for authentication purposes. This vehicle identifier can be associated with an identifiable individual (e.g., driver or vehicle owner). In this case, the data becomes personal information whose protection and confidentiality would fall under stringent requirements. An adversary can capture communications and link the identifiers to specific vehicles, and consequently to the drivers (ID disclosure), providing a means for surveillance. Moreover, the confidentiality of the beacon contents is another concern. The vehicle size attribute, acceleration, speed, steering angle, and position within a beacon message helps an observer to decide which beacon to link to a single vehicle. For example, in areas where a specific size vehicle is not frequent, two beacons can be linked to a single vehicle with high confidence [5].

1.1 Research Motivation

This system is useful if all messages are legitimate. If broadcast messages can be altered by an attacker or bogus messages can be generated by an impersonator, this may have negative effects on the services provided. Within the automotive environment, such cyber attacks present significant safety risks for vehicle occupants and other road passengers. For example, suppose an attacker intends to change the route of a vehicle. To do this, they could generate a false safety message about traffic by braking within a short period of time, or capturing the message and broadcasting it over the network repeatedly. Legitimate vehicles receiving these messages interpret this as indicator of traffic jam and change their route to avoid it. Thus, mechanisms should be applied to ensure both identification of the data source

(entity authentication) and authentication of the message (assurance of data origin and data integrity). Further, for accountability, we need to ensure the sender of a message cannot deny the message transmission [3], [4].

Authentication requirements can often conflict with privacy requirements. A unique identifier, such as a Vehicle Identification Number (VIN), is provided to a vehicle for authentication purposes. This vehicle identifier can be associated with an identifiable individual (e.g., driver or vehicle owner). In this case, the data becomes personal information whose protection and confidentiality would fall under stringent requirements. An adversary can capture communications and link the identifiers to specific vehicles, and consequently to the drivers (ID disclosure), providing a means for surveillance. Moreover, the confidentiality of the beacon contents is another concern. The vehicle size attribute, acceleration, speed, steering angle, and position within a beacon message helps an observer to decide which beacon to link to a single vehicle. For example, in areas where a specific size vehicle is not frequent, two beacons can be linked to a single vehicle with high confidence [5].

To avoid privacy breaches, protection of the driver's identity during authentication must be guaranteed. A mechanism is required to provide message anonymity while also enabling identification by a trusted party [6]. For example, a government transportation authority or a vehicle manufacturer in liability-related cases may trace a vehicle that disrupts the network. In this case, a driver profile may be fetched to be used for legitimate reasons. However, other users should not be able to identify drivers. Designing an authentication mechanism that preserves driver privacy and

also tracks malicious or dishonest adversarial actions is a major challenge.

From a practical perspective, different cars may have different processing capacities to support C-ITS applications such as vehicular safety. For economic reasons, car manufacturers embed small-scale and low-cost hardware for vehicular communications, constraining the mechanisms applicable to secure communications. The limited in-car computational capabilities make complex cryptographic techniques infeasible [7][8]. For example, the RoadLINK SAF5400 [9] can relay up to 2000 safety message verifications per second on chip. Protocol designers must keep this constraint in mind. Hence, implementation efficiency is an important aspect of a privacy-preserving authentication scheme.

Using V2V technology for safety purposes requires the communication channel to be accessible for latency-critical applications, such as collision avoidance, with the highest priority and reliability [10]. Latency defines the time frame from when information is generated for transmission and when it is received. The required latency for most safety-critical applications is 100 Milliseconds (ms) in a communication range of 150 to 500 m [11].

1.2 Research Challenge

For V2V communication in VANETs, any possible solution for privacy-preserving authentication in latency-critical applications requires high scalability and low computational overhead. This is necessary due to the limited computing resources of vehicles when handling authentication of a high number of beacons in a short period of time.

Multiple authentication schemes have been proposed to secure V2V communication (for examples see [12–18]). The schemes in [12–17] only focus on authentication without considering beacon content encryption, and as a result, they fail to provide unlinkability. Beacon content confidentiality ensures that beacon messages do not help an observer to identify a particular vehicle.

Any cryptographic technique for authentication requires the use of a cryptographic key. These keys require protection. Thus, a key-management mechanism is necessary to control the distribution, use, update, and revocation of cryptographic keys [19]. Many of these authentication schemes (e.g., [12–14], [17], [18]) do not take a key-update protocol for long-term cryptographic keying material into consideration, or they fail to design such a protocol while satisfying the security requirements (e.g., [15]).

Unfortunately, many existing proposals on privacy-preserving authentication are not effective and/or efficient for use in V2V latency-critical applications. Most do not take the scalability, computation, and communication overhead into consideration, or they fail to satisfy the security and privacy requirements. We review these schemes and identify the weaknesses. We propose an alternative scheme that meets the constraints.

1.3 Research Brief

This paper proposes a novel scheme: Anonymous Lightweight Inter-vehicle (ALI) broadcast authentication

with encryption. This specifically addresses the development and evaluation of a secure and efficient authentication and key-management scheme for latency-critical applications in VANETs, resolving scalability issues without compromising the security and privacy requirement of C-ITS.

The ALI scheme consists of five phases: system initialization, system registration, communication, identity tracking and revocation, and vehicle key-update phases. For the key-management mechanism in the ALI scheme, all vehicles within a zone (region or area) need to perform daily handshaking with a Trusted Authority (TA) before being able to commence V2V communication. This passive revocation approach requires vehicles to regularly request new short-term credentials from the TA. These requests will be rejected if the corresponding long-term credential has been revoked. The received credentials from the TA enable vehicles to receive broadcast-keys from the zone RSUs. The received credentials from the TA and RSUs together enable vehicles to derive ephemeral keys for beacon encryption and privacy-preserving message authentication.

1.4 Research Contribution

The main contributions are summarized as follows:

1. For each new V2V broadcast message, we generate random ephemeral keys and combine message authentication with beacon encryption using Hash-based Message Authentication Code (HMAC) [20] and Advanced Encryption Standard (AES) [21], respectively. This produces different ciphertexts for similar beacon messages, providing anonymity and unlinkability.
2. We use public-key functionality via symmetric techniques provided by Elliptic Curve Integrated Encryption Scheme (ECIES) [22] and user inaccessible cryptographic keying material stored in vehicle Tamper-Proof Devices (TPDs). This enables authorized vehicles to anonymously exchange beacons, while permitting a TA to resolve disputes and track malicious vehicles.
3. We utilize the ECQV implicit certificate scheme to increase scalability and reduce computational overhead. To the best of our knowledge, this is the first study to apply ECQV implicit certificates in the construction of a VANET authentication scheme.
4. We perform an in-depth security analysis supported by formal security proof to demonstrate that ALI is indeed provably secure. We demonstrate the efficiency of ALI scheme by performing extensive performance analysis: ALI produces only 149 bytes of cryptographic overhead and can authenticate approximately 700 broadcast messages every 100 ms.

1.5 Organization of the Paper

The paper is organized as follows. Section 2 overviews the previous proposals. Section 3 introduces the system model, adversary's capabilities and assumptions, and security objectives. Section 4 explains the building blocks of our scheme. In Section 5, our scheme is presented in detail. Section 6 gives the security and privacy analyses. Section 7 presents the efficiency analyses (computation cost and communication cost for certain simulation scenarios). Finally, we present the concluding remarks in Section 8.

2 EXISTING WORK

This section reviews the most relevant state-of-the-art research into broadcast authentication schemes in V2V communication.

Raya and Hubaux [12] propose a Baseline Pseudonymous (BP) authentication scheme that requires a large number of short-lived pseudonym certificates that are pre-loaded in a vehicle OBU by TA. Each broadcast message needs to be signed using a certificate from the vehicle certificate pool. A vehicle should change its anonymous certificate and private key only after having used it for a certain number of messages. This pseudonym changing strategy alone is not sufficient to prevent an observer from linking the new pseudonym to the vehicle [23].

Group-oriented signature based schemes [24] provide non-traceability, unlinkability, and unforgeability. It is computationally infeasible for an adversary to learn whether two signatures on the message have been signed by the same group member, as only one public key is used for a group. The Certificate Revocation List (CRL) size of group signature is linear with the number of revoked vehicles. At least two cryptographic pairing operations are involved with the revocation checking [13]. However, the pairing calculations [25] used in group signature schemes are not computationally efficient for the VANETs latency-critical applications.

Hybrid schemes combine different authentication approaches, such as hash functions, MACs, Elliptic Curve Digital Signature Algorithm (ECDSA), group signatures, and pseudonymous certificates. Studer et al. [14] propose VAST: a hybrid authentication mechanism based on ECDSA and a modified version of the TESLA [26]. The TESLA protocol uses symmetric cryptography, reducing the computational overhead for authentication. However, TESLA cannot provide non-repudiation [14]. VAST is efficient, but it does not provide privacy preservation and conditional traceability.

Bae et al. [16] demonstrate that, when using the ECDSA for authentication of a high volume of messages and certificates, the verification causes a delay in driver notification and may result in insufficient driver reaction time. They recommend use of ECDSA and ECQV together while verifying certificates after every specific period of time. Note that the ECQV implicit certificate scheme has not yet been used in the construction of VANET authentication schemes.

Wang et al. [15] propose 2FLIP; all vehicle embedded TPDs are equipped with an identical copy of the system key to generate and verify MACs. However, this is viewed as a single point of failure [17]. If a single vehicle's system-key is compromised, all vehicles in the network will be affected, and all will need to be updated with a new system-key.

Huang et al. [17] propose a new scheme to address the system-key problem in the scheme presented by Wang et al. [15]. However, the scheme is not scalable. For a new vehicle that joins into the network, the TA generates a list of pseudo identifiers along with a list of hashed values of the pseudo identifiers, and sends to the vehicle. Then, the TA updates all other vehicles with the hash of pseudo identifiers that are generated for the new vehicle. This means that all vehicles must store each other's hashed values of pseudo identifiers. In case of a revocation, all vehicles must be updated with a new list of pseudo identifiers and hashed values.

Recently, Camenisch et al. [18] introduce the novel concept of zone encryption as a practical means to authentically and confidentially transmit data in vehicular communications. The zone encryption idea relies on symmetric authenticated encryption using temporary keys that are exchanged among vehicles. However, the scheme does not provide non-repudiation in V2V communication, and as a result, traceability and revocation cannot be performed in the case of malicious activities.

3 PROBLEM STATEMENT

This section outlines the network model and assumptions, adversary's capabilities, and the design objectives.

3.1 Network Model and Assumptions

The vehicular network model used in this study consists of a Trusted Authority (TA), RSUs along the roads, OBUs embedded in vehicles, and communication between these entities. We discuss each component below.

TA: We assume the TA is always secure, trusted, and online. The TA can be implemented in a multi-layer structure, e.g., root TA and several sub-TAs. For the sake of simplicity, we show the TA as a single entity. The TA is a managing authority that acts as the root of trust to generate, update, and revoke credentials for other network entities, including vehicles. For example, the Department of Motor Vehicles or the Department of Transportation (DOT) can act as the root TA. Further, the TA is the only entity which can reveal a vehicle's real identifier in case of dispute.

RSU: The distributed RSUs are equipped with a higher computational capability and transmission power than OBUs. We assume that RSUs are not trusted parties. The TA and RSUs can connect to each other through wired connections or the Internet. The RSUs work as gateways to deliver data from the TA to roadside vehicles and vice versa. The range of an RSU-to-vehicle communication may be larger than that of the V2V and vehicle-to-RSU communications [27].

Vehicle: Smart vehicles equipped with OBU, sensors, and Global Positioning System (GPS) move along the roads, and communicate with other vehicles and RSUs according to a defined Intelligent Transportation Systems Radio Service (ITS-RS) standard, such as the Dedicated Short Range Communications (DSRC) protocol [27]. A TPD is embedded in each OBU to store the user inaccessible cryptographic keying materials involved in cryptographic operations. Moreover, each vehicle has a unique barcode/identifier that is known to the DOT.

3.2 Adversary's Capabilities and Motivations

Let \mathcal{A} be a Probabilistic Polynomial-Time (PPT) adversary. \mathcal{A} possess impressive communication abilities through powerful receivers, can control the communication channel, monitor on-the-fly data exchange, and delay their transmission, as well as tamper with messages and replace the original messages with modified messages. The adversary may intend to maximize its gains by cheating neighboring vehicles to make a clear path, or compromising legitimate users identities for impersonation, in order to take no responsibility for

its injurious behaviors. Furthermore, a malicious adversary may deliberately generate large amounts of legitimate or invalid messages in a relatively short period of time to disrupt the VANET safety applications, creating disorder.

Definition 1 (Adversary's Advantage). \mathcal{A} can guess b' for a bit b where b is a fair coin. The advantage $Adv_{\mathcal{A}}$ is the probability taken over all coin tosses made by \mathcal{A} , which is:

$$Adv_{\mathcal{A}} = 2|Pr[b = b'] - 1/2|.$$

The advantage $Adv_{\mathcal{A}}$ measures that how much \mathcal{A} is better than an adversary who simply guesses at random. An adversary who guesses at random has a success probability of $1/2$, and an advantage of $Adv = 0$. When the adversary always correctly finds the value of b' , we have $Pr[b = b'] = 1$ and thus an advantage of $Adv = 1$.

3.3 Security Objectives and Design Goals

In this section, we focus on security and privacy objectives of our proposal. The detailed description of each objective and goal is as follows.

Message Integrity and Authentication: Message authentication provides assurance of data origin, and also data integrity [19]. The authentication process must be secure against an adversary trying to fabricate (forge) an authentication tag for a message without having access to the respective legitimate sender's secret credential.

Anonymity and Unlinkability: It should be computationally infeasible for an adversary and the network entities (except the TA) to link a message to a vehicle/driver correctly. If an observer tried to guess which vehicle transmitted a particular message, there should be only a low probability of linking a vehicle's actions or identifying it among the set of all vehicles.

Non-Repudiation and Revocation: The TA in liability-related cases has to trace an OBU that abuses the network, for example cheating neighboring vehicles to make a clear path. In addition, the authentication mechanism should be able to revoke dishonest vehicles from further commitment in the VANET communications. A dishonest vehicle cannot deny its action (non-repudiation), such as the time of generating and transmitting messages.

Perfect Forward Secrecy: We need to ensure that compromise of past session keys does not allow an adversary to compromise future session keys [19, §12.17]. A protocol is vulnerable to a known-key attack if perfect forward secrecy is not provided. The more frequently the keys are updated, the less data is processed with any given key, and as a result, the less impact the leak will have.

Low Computation and Communication Overhead: Considering the vehicles limited computation capability, the scheme should have low computational overhead to handle authentication of a high number of beacons in a short period of time. Moreover, the bandwidth needs to be effectively used by vehicles for an authentication request.

4 BUILDING BLOCKS

The ALI scheme uses well-known cryptographic primitives, including: the ECIES, ECDSA, and ECQV. This section briefly reviews their underlying security assumptions.

4.1 Elliptic Curve Integrated Encryption Scheme

In the ALI scheme, we use the ECIES presented by Abdalla et al. [22]: a hybrid public-key encryption scheme that employs the Diffie-Hellman protocol [28] over elliptic curves to establish a symmetric encryption key. This sub-section briefly reviews the well-known cryptographic primitives that we use with ECIES, outlining the underlying security assumptions.

4.1.1 Symmetric Encryption Scheme

Let (Enc, Dec) denote a symmetric encryption scheme which provides Indistinguishability under Chosen-Plaintext Attacks (IND-CPA). That is, given two messages of equal length, a challenger randomly selects one of the messages to encrypt, and then sends the ciphertext to the adversary. It should be hard for the adversary to distinguish which of the messages was encrypted. The maximum advantage of PPT adversary \mathcal{A} in breaking the IND-CPA property of (Enc, Dec) is denoted by $Adv_{\mathcal{A}(Enc, Dec)}^{ind-cpa}$. In our proposal, we use the Advanced Encryption Standard in Counter Mode (AES-CTR) for symmetric encryption, as specified in National Institute for Standards and Technology (NIST) Special Publication (SP) 800-38A [29].

4.1.2 Hash-based Message Authentication Code

Let $HMAC()$ denote a HMAC function which is believed to satisfy Strong Unforgeability under Chosen-Message Attacks (SUF-CMA). That is, it should be computationally infeasible for the adversary to find a valid pair of message and tag; whether the message is new and has never been signed by a legitimate signer, or the message is old and the new tag is not previously attached to this message by a legitimate signer [20]. The maximum success probability of PPT adversary \mathcal{A} in finding a valid pair (forgery) is denoted by $Succ_{\mathcal{A}(HMAC)}^{suf-cma}$, where \mathcal{A} is given access to the tagging/verification oracle. In our proposal, we use the HMAC for message authentication, as specified in the (U.S.) Federal Information Processing Standard (FIPS) 198-1 [30], as well a Secure Hash Algorithm $Hash$, as specified in the FIPS 180-4 [31].

4.1.3 Hash-based Key-Derivation Function

Let $HKDF()$ denote a Hash-based Key-Derivation Function (HKDF) based on the HMAC presented in Section 4.1.2. This extracts an input master secret key k , and expands it into several additional pseudorandom secret keys [32]. The security of HKDF as a HMAC-based KDF is based on the assumption that the compression function of the underlying hash is itself a pseudorandom function (PRF) [33]. The maximum advantage of PPT adversary \mathcal{A} in distinguishing the outputs of PRF better than by a random guess and breaking security of the PRF is denoted by $Adv_{\mathcal{A}(PRF)}^{ind-sec}$. In our proposal, we use the HKDF to convert a shared-secret into key material suitable for use in encryption, integrity checking or authentication, as specified in the Request For Comments (RFC) 2898 [34].

4.2 Digital Signature Scheme

Let (Sig, Ver) denote a digital signature scheme which provides Existential Unforgeability under Chosen-Message

Attacks (EUF-CMA). That is, it should be computationally infeasible for the adversary to create at least one pair of message/signature, where message has never been signed by the legitimate signer. The maximum success probability of PPT adversary \mathcal{A} in finding a forgery is denoted by $Succ_{\mathcal{A}(Sig, Ver)}^{euf-cma}$, where \mathcal{A} is given access to the signing oracle. In our proposal, we use the ECDSA [35] for messages authenticate with non-repudiation, as specified in [36].

4.3 Elliptic Curve Qu-Vanstone

An ECQV implicit certificate [37] is a variation of the digital certificates that must be reconstructed from public data. The scheme is particularly well suited for application environments where resources such as computing power is limited [16].

Algorithm 1 ECQV

procedure P1: Certificate Generation by TA

- 1: Receive a certificate request $T_v = \hat{R}_v$ from a vehicle v , where $R_v = k_v.P$, k_v is a random integer $\in \mathbb{R}\mathbb{Z}_q^*$, and id_v is a vehicle identifier.
- 2: Select a random integer $k_{ta} \in \mathbb{R}\mathbb{Z}_q^*$.
- 3: Compute public key reconstruction data:
 $P_v = R_v + k_{ta}.P$.
- 4: Compute $imCert_v = Encode(P_v, id_v)$, where $Encode$ is an encoding algorithm (e.g., Base64).
- 5: Compute $e = Hash(imCert_v)$.
- 6: Compute private key reconstruction data:
 $r = ((e \times k_{ta}) + d_{ta}) \pmod{q}$.
- 7: Return $(r, imCert_v)$.

procedure P2: Public key Computation by Any Party

- 1: Compute $e = Hash(imCert_v)$.
- 2: Compute $Q_v = (e.P_v + Q_{ta})$.
- 3: Return public key Q_v .

procedure P3: Private key Computation by Vehicle v

- 1: Compute $e = Hash(imCert_v)$.
 - 2: Compute $d_v = ((e \times k_v) + r) \pmod{q}$.
 - 3: Return private key d_v .
-

Using ECQV, a vehicle TPD generates a certificate request (using order q of the base point generator P on elliptic curve E) and sends that to the TA for certificate generation. Algorithm 1 shows the pseudocode for certificate generation, public key computation, and private key computation in three separate procedures: **P1**, **P2**, and **P3**. A vehicle public key Q_v can be computed by any party that knows the vehicle's certificate $imCert_v$ and TA's public key Q_{ta} (Algorithm 1 §P2). Unlike the traditional explicit public-key certificates, in ECQV certificate generation the TA has no knowledge about the private key. It is computationally hard for all entities (except the owner of key) to reconstruct the ECQV private key d_v without knowledge of random k_v (Algorithm 1 §P3). The security of ECQV relies on the standard security assumption of hardness of the Discrete Logarithm Problem in Elliptic Curves (ECDLP) [38].

5 THE ALI AUTHENTICATION SCHEME

This section presents the ALI scheme for VANETs. The ALI can be used for both V2TA and V2V communications. The ALI scheme consists of five phases: system initialization, system registration, communication, identity tracking and revocation, and vehicle key-update phases.

5.1 System Initialization Phase

In this phase, the car manufacturer and the TA perform the following steps to generate the cryptographic parameters.

5.1.1 Car Manufacturer

In this phase, the Car Manufacturer (CM) generates the system parameters in accordance with the IEEE 1609.2 security standard, and pre-loads them in the TPDs. The following steps are executed by the CM in this phase:

1. The CM generates a unique identifier id_{tpd} for the vehicle TPD, and generates a unique barcode/identifier id_v for the vehicle (e.g., 17-digits vehicle identifier number) that reveals the vehicle information in detail (e.g., vehicle TPD's identifier id_{tpd}).
2. The CM generates the system parameters $CM_{par} = \{id_{tpd}, id_v, crypt\}$, where $crypt$ is a cryptographic library containing different algorithms (e.g., Secure Hash Algorithms SHA-256, SHA-512, and AES-CTR), and different elliptic curves and their fundamental parameters such as finite fields in accordance with the IEEE 1609.2 security standard, and stores CM_{par} in the vehicle TPD.

5.1.2 Trusted Authority

In this phase, the TA generates system parameters which are necessary for the network entities (vehicles and RSUs) to establish trust with the TA and receive the related credentials to commence network communications. The following steps are executed by the TA in this phase:

1. The TA chooses a large prime p , a random point P of order q , and an elliptic curve $E(\mathbb{F}_q)$ over finite field \mathbb{F} to satisfy the *Short Weierstrass form* of equation: $y^2 = x^3 + ax + b \pmod{q}$ [39], where $q = p^m$ for $m \geq 1$ is the smallest positive integer such that $q.P = \mathcal{O}$, $a, b \in \mathbb{F}_q$ are the constant coefficients of the curve, and $\langle P \rangle$ is the cyclic subgroup of points generated by P .
2. The TA chooses a random private key $d_{ta} \in \mathbb{R}\mathbb{Z}_q^*$, and computes public key $Q_{ta} = d_{ta}.P$.
3. The TA generates system parameters $TA_{par} = \{Q_{ta}, p, q, a, b, P, csc\}$, where csc is a list of cipher suite components applicable in the network (e.g., NIST P-256 curve and SHA-256).

5.2 System Registration Phase

In this phase, the RSUs and vehicles are registered in the network by the TA.

5.2.1 Roadside Unit Registration

Each RSU is configured at the time of installation at the roadside, or before that. An officer who is provided a handheld device (e.g., a mini-computer) which acts as a trusted gateway to securely connect the RSU to the TA through the Transport Layer Security (TLS) protocol [40]. The TA authenticates the device and the officer using their credentials (e.g., password-based authentication), and verifies their authorization to perform the task. Once verified, the below steps are executed by an RSU and the TA in this phase:

RSU \leftarrow **TA**

1. The TA sends TA_{par} to the RSU.

RSU \rightarrow **TA**

1. The RSU extracts the system parameters TA_{par} .
2. The RSU chooses a random integer $k_{rsu} \in \mathbb{R}\mathbb{Z}_q^*$ and generates an ECQV implicit certificate request $T_{rsu} = \{id_{rsu}, R_{rsu}\}$, where $R_{rsu} = k_{rsu} \cdot P$ (Algorithm 1 §P1). Finally, the RSU sends T_{rsu} to the TA.

RSU \leftarrow TA

1. The TA generates a response $T'_{rsu} = \{imCert_{rsu}, r_{rsu}\}$ for the RSU, where $imCert_{rsu}$ is an ECQV implicit certificate, and r_{rsu} is an ECQV private key reconstruction data. Then, sends the T'_{rsu} to the RSU.

RSU \downarrow

1. The RSU extracts T'_{rsu} .
2. The RSU computes public key Q_{rsu} and private key d_{rsu} from $imCert_{rsu}$ and r_{rsu} , respectively (Algorithm 1 §P2, P3).

At the end of this phase the officer's task is completed and the RSU can communicate securely with the TA directly (e.g., via the TLS).

5.2.2 Vehicle Registration

The vehicle registration phase consists of two parts: firstly, the vehicle owners register their vehicles on the TA's on-line web-based platform over the Internet, and secondly, the vehicle TPDs communicate with the TA to receive the relevant credentials to enable them to further participate in the network.

User Registration: Before a vehicle is able to commence network communications, the vehicle owner (as a main user) needs to link the vehicle information to his/her identifier. All vehicles are known to the DOT through id_v that reveals the vehicle information in detail (e.g., vehicle TPD's identifier id_{tpd}). Once the vehicle information and its ownership is verified, the TA generates a ticket value for the vehicle TPD. The vehicle user enters this ticket value into the TPD through the vehicle OBU computing interface. Then, the TPD sends a registration request to the TA through an RSU within communication range of the vehicle. Finally, the TA generates a public/private key pair and replies to the vehicle through an RSU. This enables the vehicle to commence network communications. The following steps are executed by the TA and a vehicle user in this phase:

User \Rightarrow TA

1. The user creates an account on the TA's Internet-based on-line vehicle registration server.
2. The user provides a valid e-mail address and chooses a strong password pwd_u to complete the registration process.
3. The TA computes $PWD_u = HKDF(salt_u, pwd_u)$, where $salt_u$ is a long unique value generated truly at random. Then, the server securely stores the hash value PWD_u in its database.
4. The user provides further identity information (e.g., national identity/security number), a valid mobile phone number, or alternatively requests a hand-held password generator token. The token generates a password which is synchronized with the server, and is valid for a short period of time (e.g., one minute).
5. The TA generates a unique identifier id_u for the user, and verifies the vehicle ownership through the DOT. Then, the TA links the vehicle identifiers id_{tpd} and id_v to id_u , and keeps the record in its database.

6. The TA generates a long unique random value ticket tic (e.g., 16 bytes) using a pseudo-random number generator.
7. The TA derives a secret key k_3 such that $k_3 = HKDF(3, tic)$.
8. The TA generates a HMAC tag $ref = k_3 \delta(0)$ as a reference number.
9. The TA links tic and ref to id_v in its database for a short period of time (e.g., 24 hours).
10. The TA sends tic to the user through the TA's on-line secure web service. Note: the user needs to keep tic secret for the vehicle registration phase.

The Vehicle TPD Registration: Figure 1 shows the message exchange in this phase. Once a vehicle owner received the ticket value tic , he/she needs to register the vehicle TPD. The vehicle user configures the TPD settings related to the cipher suite components used in the TPD registration phase, such as the elliptic curve and the hash function used for key derivation. The related instructions must be publicly available on the TA's website or the DOT's website. The configuration can be set through the OBU's computing interface. Note that in this phase the TPD is not aware of the system parameters and the cipher suite components used in a region. The following steps are executed by the vehicle TPD and the TA in this phase:

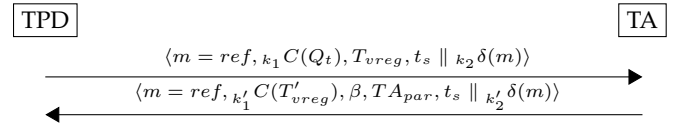


Fig. 1. Vehicle registration phase.

TPD \rightarrow TA

1. The vehicle user enters the ticket value tic through the OBU's computing interface.
2. The TPD derives three secret keys k_1 , k_2 , and k_3 such that $k_1 = HKDF(1, tic)$, $k_2 = HKDF(2, tic)$, and $k_3 = HKDF(3, tic)$.
3. The TPD generates a HMAC tag $ref = k_3 \delta(0)$ as a reference number (similar to the TA's generated ref).
4. The TPD chooses a random temporary private key $d_t \in \mathbb{R}\mathbb{Z}_q^*$ and computes a temporary public key $Q_t = d_t \cdot P$.
5. The TPD chooses a random integer $k_v \in \mathbb{R}\mathbb{Z}_q^*$ and generates a certificate request $T_{vreg} = \{R_v\}$ such that $R_v = k_v \cdot P$ (Algorithm 1 §P1).
6. The TPD computes ciphertext $k_1 C(Q_t)$. Note that the key Q_t must be encrypted because it is used to establish a Diffie-Hellman shared secret key by the TA when it responds to this request. Otherwise, the adversary can capture this information and forge an authentication tag on any malicious response.
7. The TPD creates message $m = \{ref, k_1 C(Q_t), T_{vreg}, t_s\}$.
8. The TPD generates a HMAC tag $k_2 \delta(m)$ on message m .
9. The TPD sends tuple $\{m \parallel k_2 \delta(m)\}$ to the TA. This tuple is only sent once, which does not harm the privacy.

TPD \leftarrow TA

1. The TA determines the freshness, and accepts the message if $0 \leq t_r - t_s \leq \Delta t$, otherwise drops the message.
2. The TA reads reference number ref in message m , and searches for the corresponding id_v linked to ref in its database (the TA drops the message if nothing is found).

3. The TA loads the corresponding tic value linked to id_v and derives two secret keys k_1 and k_2 such that $k_1 = HKDF(1, tic)$ and $k_2 = HKDF(2, tic)$.
4. The TA generates a new HMAC tag $k_2\delta'(m)$ and accepts the message if, and only if, $k_2\delta'(m) = k_2\delta(m)$. This is to ensure that message m has not been altered.
5. The TA decrypts ciphertext $k_1C(Q_t)$ and recovers the vehicle TPD's temporary public key Q_t .
6. The TA generates an ECQV implicit certificate $imCert_v$ and a private key reconstruction data r_v for the TPD (Algorithm 1 §P1).
7. The TA links $imCert_v$ to id_v and id_u , and keeps the record in its database. This should save communication bandwidth in future communications, as a vehicle only sends its encrypted id_v to the TA instead of sending its encrypted certificate.
8. The TA chooses a random integer $b \in \mathbb{R}\mathbb{Z}_q^*$, and computes $\beta = b.P$ and $k' = b.Q_t$.
9. The TA derives two secret keys k'_1 and k'_2 such that $k'_1 = HKDF(1, k')$ and $k'_2 = HKDF(2, k')$.
10. The TA generates a response $T'_{vreg} = \{imCert_v, r_v, salt_u\}$ and computes ciphertext $k'_1C(T'_{vreg})$.
11. The TA generates a HMAC tag $k'_2\delta(m)$ on message $m = \{ref, k'_1C(T'_{vreg}), \beta, T_{Apar}, t_s\}$.
12. The TA sends tuple $\{m \parallel k'_2\delta(m)\}$ to the TPD.
13. The TA removes tic and ref under the vehicle id_v in its database.

TPD ↓

1. The TPD corresponding to ref determines the freshness, and accepts the message if $0 \leq t_r - t_s \leq \Delta t$, otherwise drops the message.
2. The TPD computes $k' = d_t.\beta$. This works because $d_t.\beta = d_t.(b.P) = b.(d_t.P) = b.Q_t$.
3. The TPD derives two secret keys k'_1 and k'_2 such that $k'_1 = HKDF(1, k')$ and $k'_2 = HKDF(2, k')$.
4. The TPD generates a new HMAC $k'_2\delta'(m)$ and accepts the message if, and only if, $k'_2\delta'(m) = k'_2\delta(m)$, otherwise drops the message. This is to ensure that m has not been altered.
5. The TPD decrypts ciphertext $k'_1C(T'_{vreg})$ and extracts $T'_{vreg} = \{imCert_v, r_v, salt_u\}$.
6. The TPD stores the $salt_u$ and T_{Apar} , and computes public key Q_v and private key d_v from $imCert_v$ and r_v , respectively.

5.3 Communication Phase

The communication phase consists of three communication types: V2TA, daily handshaking, and V2V communications. Note that there is a hierarchy. A V2TA communication is required for daily handshaking, and a daily handshaking is necessary for V2V communication. Before a vehicle is able to commence V2V communication within a zone (region or area), it must communicate with the TA and receive the zone's necessary credentials. These credentials should be updated daily, and as a result, a daily handshaking between vehicles and the TA is required. The received credentials from the TA enable vehicles to receive broadcast-keys from the zone RSUs. The received credentials from the TA and RSUs enable vehicles to derive ephemeral keys for beacon encryption and message authentication. We explain these communications according to their priority.

5.3.1 Vehicle-to-TA Communication

Figure 2 shows the message exchange in this phase. In Vehicle-to-TA (V2TA) communication all vehicles can communicate with the TA through an RSU in communication range as a message forwarder. The following steps are executed by a vehicle TPD and the TA in this phase:

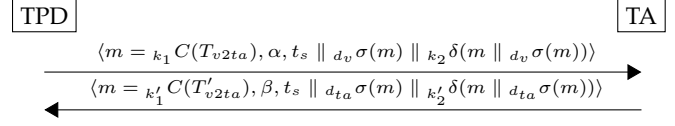


Fig. 2. Vehicle-to-TA communication.

TPD → TA

1. The TPD chooses a random temporary private key $d_t \in \mathbb{R}\mathbb{Z}_q^*$ and computes a temporary public key $Q_t = d_t.P$. The key Q_t is used to establish a Diffie-Hellman shared secret key by the TA when it responds to this request.
2. The TPD chooses a random integer $a \in \mathbb{R}\mathbb{Z}_q^*$, and computes $\alpha = a.P$ and $k = a.Q_t$.
3. The TPD derives two secret keys k_1 and k_2 such that $k_1 = HKDF(1, k)$ and $k_2 = HKDF(2, k)$.
4. The TPD generates a request $T_{v2ta} = \{id_v, Q_t\}$ and computes ciphertext $k_1C(T_{v2ta})$.
5. The TPD generates an ECDSA signature $d_v\sigma(m)$ on message $m = \{k_1C(T_{v2ta}), \alpha, t_s\}$. This is to provide source authentication and non-repudiation.
6. The TPD generates a HMAC tag $k_2\delta(m \parallel d_v\sigma(m))$ on tuple $\{m \parallel d_v\sigma(m)\}$. Finally, the TPD sends tuple $\{m \parallel d_v\sigma(m) \parallel k_2\delta(m \parallel d_v\sigma(m))\}$ to the TA.

TPD ← TA

1. The TA determines the freshness, and accepts the message if $0 \leq t_r - t_s \leq \Delta t$, otherwise drops the message.
2. The TA computes $k = d_{ta}.\alpha$. This works because $d_{ta}.\alpha = d_{ta}.(a.P) = a.(d_{ta}.P) = a.Q_t$.
3. The TA derives two secret keys k_1 and k_2 such that $k_1 = HKDF(1, k)$ and $k_2 = HKDF(2, k)$.
4. The TA generates a new HMAC tag $k_2\delta'(m \parallel d_v\sigma(m))$ and accepts the message if, and only if, $k_2\delta'(m \parallel d_v\sigma(m)) = k_2\delta(m \parallel d_v\sigma(m))$, otherwise rejects the tuple. This is to ensure the tuple has not been altered.
5. The TA decrypts ciphertext $k_1C(T_{v2ta})$ and recovers the request $T_{v2ta} = \{id_v\}$.
6. The TA searches for the vehicle's corresponding implicit certificate $imCert_v$ which is linked to id_v in its database.
7. The TA reconstructs the vehicle's public key Q_v using the certificate (Algorithm 1 §P1, P2).
8. The TA verifies the ECDSA signature and accepts the message if, and only if, $Ver(d_v\sigma(m), Q_v) = 1$. This is to ensure that message m originated from the expected vehicle TPD.
9. The TA checks if the vehicle implicit certificate $imCert_v$ is valid (has not been revoked recently), otherwise stops the process.
10. The TA chooses a random integer $b \in \mathbb{R}\mathbb{Z}_q^*$, and computes $\beta = b.P$ and $k' = b.Q_t$.
11. The TA derives two secret keys k'_1 and k'_2 such that $k'_1 = HKDF(1, k')$ and $k'_2 = HKDF(2, k')$.
12. The TA generates a response T'_{v2ta} to the vehicle TPD request T_{v2ta} , and computes ciphertext $k'_1C(T'_{v2ta})$.

13. The TA generates an ECDSA signature $d_{ta}\sigma(m)$ on message $m = \{k'_1 C(T'_{v2ta}), \beta, t_s\}$. This is to provide source authentication and non-repudiation.

14. The TA generates a HMAC tag $k'_2\delta(m \parallel d_{ta}\sigma(m))$ on tuple $\{m \parallel d_{ta}\sigma(m)\}$. Finally, the TA sends tuple $\{m \parallel d_{ta}\sigma(m) \parallel k'_2\delta(m \parallel d_{ta}\sigma(m))\}$ to the TPD.

TPD ↓

1. The TPD determines the freshness, and accepts the message if $0 \leq t_r - t_s \leq \Delta t$, otherwise drops the message.

2. The TPD computes $k' = d_t.\beta$. This works because $d_t.\beta = d_t.(b.P) = b.(d_t.P) = b.Q_t$.

3. The TPD derives two secret keys k'_1 and k'_2 such that $k'_1 = HKDF(1, k')$ and $k'_2 = HKDF(2, k')$.

4. The TPD generates a new HMAC tag $k'_2\delta'(m \parallel d_{ta}\sigma(m))$ and accepts the message if, and only if, $k'_2\delta'(m \parallel d_{ta}\sigma(m)) = k'_2\delta(m \parallel d_{ta}\sigma(m))$, otherwise rejects the tuple. This is to ensure the tuple is not altered.

5. The TPD verifies the ECDSA signature and accepts the message if, and only if, $Ver(d_{ta}\sigma(m), Q_{ta}) = 1$. This is to ensure that message m originated from the TA.

6. The TPD decrypts ciphertext $k'_1 C(T'_{v2ta})$ and recovers the response T'_{v2ta} .

5.3.2 Daily Handshaking and Broadcast Session

In our scheme, all vehicles are required to perform daily handshaking with the TA and join a VANET Broadcast Session (VBS). This provides an additional security layer that restricts malicious TPDs from joining in V2V communication.

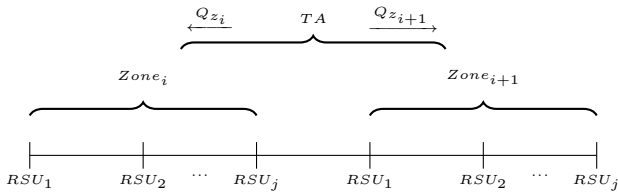


Fig. 3. Two typical zones z_i and z_{i+1} with number of j RSUs, receiving public keys Q_{z_i} and $Q_{z_{i+1}}$ from the TA.

The TA chooses two random private keys $d_z \in \mathbb{R}\mathbb{Z}_q^*$ (here we call master-key) and d_{Bta} (here we call TA's broadcast-key). The master-key d_z is valid only in a specific zone z , and will expire after a period of time (e.g., 24 hours). The TA computes public key $Q_{z_i} = d_{z_i}.P$ for each zone i . The TA regularly updates all authorized RSUs within a zone z_i with a newly computed Q_{z_i} (but no master-key d_z), as shown in Figure 3.

Using the vehicle-to-TA communication provided in Section 5.3.1, vehicles communicate with the TA through an RSU within communication range to forward their requests (T_{v2ta}) to join a VBS within a zone i . The TA approves the authorized vehicles requests, and securely transmits T'_{v2ta} containing d_z and d_{Bta} to their TPDs. Note that all requests from the TPDs to the TA and responses from the TA to the TPDs are encrypted (refer to Section 5.3.1). Then, each authorized vehicle TPD within a zone i calculates the public key $Q_z = d_z.P$. This enables vehicles in the zone to receive RSUs broadcast-keys to use further for V2V communication (beacon exchange). A vehicle entering to a new zone z_{i+1} is required to send a new VBS joining request to the TA.

The new requests will be processed by the TA, and only the authorized (not malicious) vehicle TPD will receive the zone's master-key and the TA's broadcast-key. For the sake of simplicity, we do not repeat the steps for sending the above VBS joining request. This can be done through vehicle-to-TA communication presented in Section 5.3.1. The TA may anytime broadcast a specific signal to a specific zone through the RSUs, and force the vehicle TPDs within the zone to renew their VBS registration by performing a fresh authentication

5.3.3 Vehicle-to-Vehicle Broadcast Communication

In ALI scheme, each RSU j in a zone chooses a private key which we call RSU's broadcast-key. This broadcast-key is valid only for a short period of time (e.g., 10 minutes). Figure 4 shows the message exchange in this phase. The following steps are executed by an RSU and vehicle TPDs in a zone:

TPD ← **RSU**

1. The RSU chooses a random broadcast-key $d_{Brsu} \in \mathbb{R}\mathbb{Z}_q^*$.

2. The RSU chooses random integer $a \in \mathbb{R}\mathbb{Z}_q^*$, and computes $\alpha = a.P$ and $k = a.Q_z$.

3. The RSU derives two secret keys k_1 and k_2 such that $k_1 = HKDF(1, k)$ and $k_2 = HKDF(2, k)$.

4. The RSU computes ciphertext $k_1 C(d_{Brsu})$.

5. The RSU generates an ECDSA signature $d_{rsu}\sigma(m)$ on message $m = \{im Cert_{rsu, k_1} C(d_{Brsu}), \alpha, t_s\}$. This is to provide source authentication and non-repudiation.

6. The RSU generates a HMAC tag $k_2\delta(m \parallel d_{rsu}\sigma(m))$ on tuple $\{m \parallel d_{rsu}\sigma(m)\}$.

7. The RSU continuously (e.g., every 1 second or less) broadcasts tuple $\{m \parallel d_{rsu}\sigma(m) \parallel k_2\delta(m \parallel d_{rsu}\sigma(m))\}$ to the vehicle TPDs in communication range.

TPD ↓

1. The TPD determines the freshness, and accepts the message if $0 \leq t_r - t_s \leq \Delta t$, otherwise drops the message.

2. The TPD computes $k = d_z.\alpha$. This works because $d_z.\alpha = d_z.(a.P) = a.(d_z.P) = a.Q_z$. Note that only authenticated vehicles can receive the TA's master-key d_z in a zone i which is valid for e.g., 24 hours (refer to Section 5.3.2).

3. The TPD derives two secret keys k_1 and k_2 such that $k_1 = HKDF(1, k)$ and $k_2 = HKDF(2, k)$.

4. The TPD generates a new HMAC tag $k_2\delta'(m \parallel d_{rsu}\sigma(m))$ and accepts the message if, and only if, $k_2\delta'(m \parallel d_{rsu}\sigma(m)) = k_2\delta(m \parallel d_{rsu}\sigma(m))$, otherwise rejects the tuple. This is to ensure the tuple has not been altered.

5. The TPD decrypts ciphertext $k_1 C(d_{Brsu})$ and recovers the RSU broadcast-key d_{Brsu} which is valid for a short period of time (e.g., 10 minutes).

6. The TPD reconstructs the RSU's public key Q_{rsu} using the RSU's certificate $im Cert_{rsu}$ (Algorithm 1 §P1, P2).

7. The TPD verifies the ECDSA signature and accepts the message if, and only if, $Ver(d_{rsu}\sigma(m), Q_{rsu}) = 1$. This is to ensure that message m originated from the RSU.

The following steps are executed by a vehicle TPD to broadcast beacon messages in this phase:

TPD → **TPD**

1. The TPD computes a public key Q_{Brsu} corresponding to the received RSU broadcast-key d_{Brsu} such that $Q_{Brsu} = d_{Brsu}.P$.

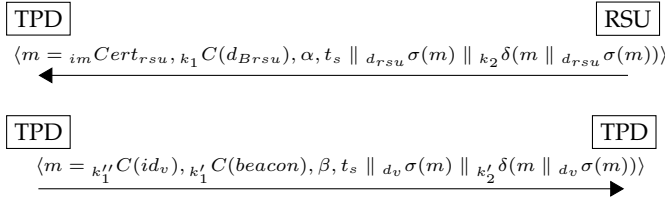


Fig. 4. Vehicle-to-vehicle broadcast communication.

2. The TPD chooses a random integer $b \in_{R} \mathbb{Z}_q^*$, and computes $\beta = b.P$, $k' = (b \times d_{Bta}).Q_{Brsu}$, and $k'' = b.Q_{ta}$.
3. The TPD derives three secret keys k'_1 , k'_2 , and k''_1 such that $k'_1 = HKDF(1, k')$, $k'_2 = HKDF(2, k')$, and $k''_1 = HKDF(1, k'')$.
4. The TPD computes ciphertext $k'_1 C(\text{beacon})$.
5. The TPD computes ciphertext $k''_1 C(id_v)$.
6. The TPD generates an ECDSA signature $d_v \sigma(m)$ on message $m = \{k''_1 C(id_v), k'_1 C(\text{beacon}), \beta, t_s\}$. This signature is verifiable by the TA and used to provide source authentication and non-repudiation.
7. The TPD generates a HMAC tag $k'_2 \delta(m || d_v \sigma(m))$ on tuple $\{m || d_v \sigma(m)\}$. Finally, the TPD broadcasts tuple $\{m || d_v \sigma(m) || k'_2 \delta(m || d_v \sigma(m))\}$ to other TPDs in range.

The following steps are executed by a vehicle TPD to handle the received beacon messages in this phase:

TPD ↓

1. The TPD determines the freshness, and accepts the message if $0 \leq t_r - t_s \leq \Delta t$, otherwise drops the message.
2. The TPD computes $k' = (d_{Brsu} \times d_{Bta}).\beta$. This works because $(d_{Brsu} \times d_{Bta}).\beta = (d_{Brsu} \times d_{Bta}).(b.P) = (b \times d_{Bta}).(d_{Brsu}.P) = (b \times d_{Bta}).(Q_{Brsu})$.
3. The TPD derives two secret keys k'_1 and k'_2 such that $k'_1 = HKDF(1, k')$ and $k'_2 = HKDF(2, k')$.
4. The TPD generates a new HMAC tag $k'_2 \delta'(m || d_v \sigma(m))$ and accepts the message if, and only if, $k'_2 \delta'(m || d_v \sigma(m)) = k'_2 \delta(m || d_v \sigma(m))$, otherwise rejects the tuple. This is to ensure the tuple is not altered.
5. The TPD decrypts ciphertext $k'_1 C(\text{beacon})$ and recovers the beacon message.

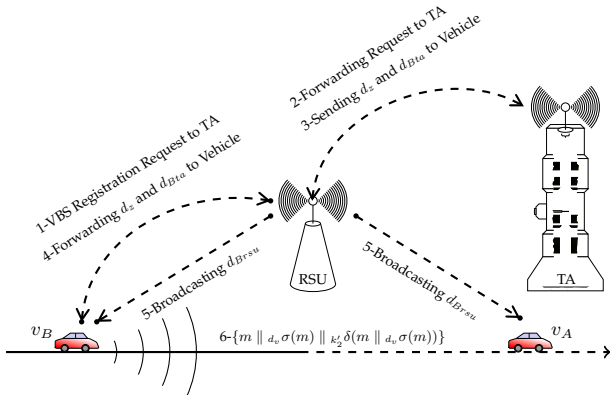


Fig. 5. The daily handshaking and V2V broadcast communication.

Figure 5 shows the V2TA, daily handshaking, and V2V broadcast communication between two vehicles v_A and v_B .

5.4 Identity Tracking and Revocation Phase

Vehicles can send dispute resolution requests to the TA through the V2TA communication (refer to Section 5.3.1). A typical dispute request contains a copy of broadcast message $\{m = k''_1 C(id_v), k'_1 C(\text{beacon}), \beta, t_s || d_v \sigma(m) || k'_2 \delta(m || d_v \sigma(m))\}$ received by a vehicle TPD which requires dispute resolution. The messages broadcast in V2V communication contain ciphertext $k''_1 C(id_v)$ that only the TA is able to recover it for dispute resolution and identity tracking. The TA executes the below steps in this phase:

TA ↓

1. The TA reads the message m , and computes $k'' = d_{ta}.\beta$. This works because $d_{ta}.\beta = d_{ta}.(b.P) = b.(d_{ta}.P) = b.Q_{ta}$.
2. The TA derives secret key k''_1 such that $k''_1 = HKDF(1, k'')$.
3. The TA decrypts ciphertext $k''_1 C(id_v)$ and recovers the vehicle identifier id_v .
4. The TA searches for the vehicle user identifier id_u and the corresponding implicit certificate $imCert_v$ which are linked to id_v in its database.
5. The TA reconstructs the vehicle's public key Q_v using the certificate (Algorithm 1 §P1, P2).
6. The TA verifies the ECDSA signature and accepts the message if, and only if, $Ver(d_v \sigma(m), Q_v) = 1$. This is to ensure that message m originated from the expected vehicle TPD with id_v .

5.5 Vehicle Key-Update Phase

In our scheme, key-update is provided through the use of the vehicle user's mobile phone or the password generator token. The TA can update all TPD related keying materials and the user's salt value $salt_u$ without relying on the TA's or the TPD's previous keys. In addition, the TA can generate a new system parameters TA_{par} including its new public/private key pair, and update the vehicles with the new parameters. The key-update procedure can be called by a vehicle user, or advertised by the TA through the RSUs; for example, at regular intervals (e.g., every 6 months) or when a long-term key is leaked.

If the user does not have a token, the following steps need to be executed by the user and the TA in this phase:

User ⇒ TA

1. The user sends a key-update request to the TA through a Short Message Service (SMS) or the TA's on-line web service.
2. The TA locates id_v and id_u corresponding to the user's mobile phone number in its database.
3. The TA generates a Personal Identification Number (PIN) pin_{phone} for the user.
4. The TA computes a symmetric key $k_{up} = HKDF(pin_{phone}, PWD_u)$, where $PWD_u = HKDF(salt_u, pwd_u)$ is the user's password saved in the TA's database (refer to Section 5.2.2).
5. The TA derives a secret key k_3 such that $k_3 = HKDF(3, k_{up})$.
6. The TA generates a HMAC tag $ref = k_3 \delta(0)$ as a reference number.
7. The TA sends pin_{phone} to the user's phone via SMS.
8. The TA links ref and k_{up} to id_v in its database for a short period of time (e.g., 1 minute).

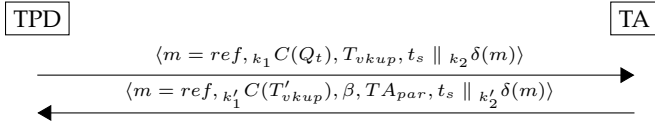


Fig. 6. Vehicle key-update phase.

Figure 6 shows the message exchange in this phase. Users who are provided a hand-held password generator token by the TA present the PIN pin_{token} to proceed in this phase. This PIN is a password generated by the token and is synchronized with the TA. The PIN is valid for a short period of time (e.g., 1 minute).

If the user has a token, the following steps need to be executed by the TA periodically (similar to the token's password changing rate) in this phase:

TA ↓

1. The TA computes a symmetric key $k_{up} = HKDF(pin_{token}, PWD_u)$.
2. The TA derives a secret key k_3 such that $k_3 = HKDF(3, k_{up})$.
3. The TA generates a HMAC tag $ref = k_3 \delta(0)$ as a reference number.
4. The TA links ref and k_{up} to id_v in its database and always replaces a new generated ref value with the old one and updates the link.

The following steps are executed by the vehicle TPD and the TA in this phase:

TPD → **TA**

1. The TPD asks the user to enter the pwd_u through the OBU's computing interface. The pwd_u is the user's registration password which is created in the vehicle registration phase (refer to Section 5.2.2).
2. The TPD loads $salt_u$ and computes $PWD_u = HKDF(salt_u, pwd_u)$.
3. The TPD asks the user to enter the pin_{phone} through the OBU's computing interface and computes a key $k_{up} = HKDF(pin_{phone}, PWD_u)$, if this is a phone-based key-update.
4. The TPD asks the user to enter the pin_{token} through the OBU's computing interface and computes a key $k_{up} = HKDF(pin_{token}, PWD_u)$, if this is a token-based key-update.
5. The TPD derives three secret keys k_1 , k_2 , and k_3 such that $k_1 = HKDF(1, k_{up})$, $k_2 = HKDF(2, k_{up})$, and $k_3 = HKDF(3, k_{up})$.
6. The TPD generates a HMAC tag $ref = k_3 \delta(0)$ as a reference number (similar to the TA's generated ref).
7. The TPD chooses a random temporary private key $d_t \in \mathbb{R}\mathbb{Z}_q^*$ and computes a temporary public key $Q_t = d_t.P$.
8. The TPD chooses a random integer $k_v \in \mathbb{R}\mathbb{Z}_q^*$ and generates a certificate request $T_{vkup} = \{R_v\}$ such that $R_v = k_v.P$ (Algorithm 1 §P1).
9. The TPD computes ciphertext $k_1 C(Q_t)$. Note that the key Q_t must be encrypted because it is used to establish a Diffie-Hellman shared secret key by the TA when it responds to this request. Otherwise, the adversary can capture this information and forge an authentication tag on any malicious response.

10. The TPD generates a HMAC tag $k_2 \delta(m)$ on message $m = \{ref, k_1 C(Q_t), T_{vkup}, t_s\}$.

11. The TPD sends tuple $\{m \parallel k_2 \delta(m)\}$ to the TA. This tuple is only sent once, which does not harm the privacy.

TPD ← **TA**

1. The TA determines the freshness, and accepts the message if $0 \leq t_r - t_s \leq \Delta t$, otherwise drops the message.
2. The TA reads reference number ref in message m , and searches for the corresponding id_v linked to ref in its database (the TA drops the message if nothing is found).
3. The TA verifies that the vehicle's implicit certificate $imCert_v$ is not recently revoked because of malicious activity, otherwise stops the process.
4. The TA loads the corresponding k_{up} value linked to id_v and derives two secret keys k_1 and k_2 such that $k_1 = HKDF(1, k_{up})$ and $k_2 = HKDF(2, k_{up})$.
5. The TA generates a new HMAC tag $k_2 \delta'(m)$ and accepts the message if, and only if, $k_2 \delta'(m) = k_2 \delta(m)$. This is to ensure that message m has not been altered.
6. The TA decrypts ciphertext $k_1 C(Q_t)$ and recovers the vehicle TPD's temporary public key Q_t .
7. The TA generates a certificate $imCert_v$ and a private key reconstruction data r_v for the TPD (Algorithm 1 §P1).
8. The TA updates the link between the new generated $imCert_v$ to id_v and id_u , and keeps the record in its database. The TA does not allow any further use of the old certificate.
9. The TA chooses a random integer $b \in \mathbb{R}\mathbb{Z}_q^*$, and computes $\beta = b.P$ and $k' = b.Q_t$.
10. The TA derives two secret keys k'_1 and k'_2 such that $k'_1 = HKDF(1, k')$ and $k'_2 = HKDF(2, k')$.
11. The TA generates a response $T'_{vkup} = \{imCert_v, r_v, salt_u\}$ and computes ciphertext $k'_1 C(T'_{vkup})$.
12. The TA generates a HMAC tag $k'_2 \delta(m)$ on message $m = \{ref, k'_1 C(T'_{vkup}), \beta, T_{Apr}, t_s\}$.
13. The TA sends tuple $\{m \parallel k'_2 \delta(m)\}$ to the TPD.
14. The TA removes k_{up} and ref under the vehicle id_v in its database.

TPD ↓

1. The TPD corresponding to ref determines the freshness, and accepts the message if $0 \leq t_r - t_s \leq \Delta t$, otherwise drops the message.
2. The TPD computes $k' = d_t.\beta$. This works because $d_t.\beta = d_t.(b.P) = b.(d_t.P) = b.Q_t$.
3. The TPD derives two secret keys k'_1 and k'_2 such that $k'_1 = HKDF(1, k)$ and $k'_2 = HKDF(2, k')$.
4. The TPD generates a new HMAC $k'_2 \delta'(m)$ and accepts the message if, and only if, $k'_2 \delta'(m) = k'_2 \delta(m)$, otherwise drops the message. This is to ensure that m has not been altered.
5. The TPD decrypts ciphertext $k'_1 C(T'_{vkup})$ and extracts $T'_{vkup} = \{imCert_v, r_v, salt_u\}$.
6. The TPD stores the $salt_u$ and T_{Apr} , and computes public key Q_v and private key d_v from $imCert_v$ and r_v , respectively.

6 SECURITY AND PRIVACY ANALYSIS

This section defines a security model, and conducts a security proof to show that ALI scheme Γ_{ALI} is indeed provably secure. Next, based on the requirements outlined in the security objectives and design goals (Section 3.3), detailed security and privacy comparisons are carried out.

6.1 Security Model and Definitions

The actual security proof consists of a reduction of the underlying computational problem to an attack against the cryptographic scheme: if there exist an adversary that has a significant advantage against Γ_{ALL} , then there is an adversary to solve the computational problem. In the computational world the PPT adversary \mathcal{A} is modeled as a PPT Turing machine [41]. This section shows that the view of \mathcal{A} in the reduction remains unchanged from the one it has during a real attack.

Definition 2 (Negligible Function). A function $\epsilon : \mathbb{N} \rightarrow \mathbb{R}$ is negligible if for every positive polynomial $p(\cdot)$ there exists an $N \in \mathbb{N}$ such that for all $N < n \in \mathbb{N}$ it holds that $\epsilon(n) < 1/p(n)$ [42].

Definition 3 (Security Model). Based on the network model and the adversary's ability, the security model for Γ_{ALL} is defined through a game played between the PPT adversary \mathcal{A} and a simulator \mathcal{S} which simulates different protocol participants according to the scheme Γ_{ALL} specification and answers all queries of the adversary.

Definition 4 (Adversarial Queries). The adversary \mathcal{A} participates in the actual execution of different protocol instances \mathcal{F} of Γ_{ALL} (e.g., V2V, V2TA) and may, at any time, send the $\text{Hash}()$, $\text{Encrypt}()$, and $\text{Sign}()$ queries to an oracle.

6.2 Formal Security Proof

Claim. The security of Γ_{ALL} (as defined in Section 3.3) is based on the security of PRF, (Enc,Dec), Hash, (Sig,Ver), and HMAC (as defined in Section 4) in function of the security parameter λ .

Proof. (Sketch): The ultimate goal of \mathcal{A} is to maliciously participate in V2V communication. We present 4 Lemmas and construct a sequence of games $G_i, i = 0, \dots, 37$ to prove the security of Γ_{ALL} . The event that adversary \mathcal{A} breaks the security of a protocol \mathcal{F} in game G_i is denoted by win_i .

Game G_0 . [Real protocol] This is the real game $\text{Real-Game}_{\mathcal{S}}^{\Gamma_{\text{ALL}}}(\lambda, \mathcal{A})$ played between \mathcal{A} and \mathcal{S} which simulates different protocols \mathcal{F} (e.g., V2V, V2TA) and their participants (e.g., TPD, TA) according to the natural protocol specification and answers the adversarial queries.

Lemma 1. *The vehicle TPD registration protocol $\mathcal{F}^{\text{vreg}}$ in Γ_{ALL} (refer to Section 5.2.2) is secure in the following sense: the maximum probability of success for \mathcal{A} to forge a valid registration request (on behalf of the TPD) and response (on behalf of the TA) in function of the security parameter λ is:*

$$\text{Succ}_{\mathcal{S}}^{\mathcal{F}^{\text{vreg}}}(\lambda, \mathcal{A}) \leq 2 \frac{1}{2^{p(\lambda)}} + 5 \text{Adv}_{\mathcal{A}(\text{PRF})}^{\text{ind-sec}} + 2 \text{Succ}_{\mathcal{A}(\text{HMAC})}^{\text{sf-cma}} + 2 \text{Adv}_{\mathcal{A}(\text{Enc,Dec})}^{\text{ind-cpa}}. \quad (1)$$

Proof of Lemma 1. We construct a sequence of games $G_i, i = 1, \dots, 8$ to prove Lemma 1.

Game G_1 . [Same random secret ticket tic in $\mathcal{F}^{\text{vreg-req}}$] In this game the simulation aborts if during the interaction the simulator on behalf of the TPD chooses the same random ticket value tic in the vehicle TPD registration protocol

$\mathcal{F}^{\text{vreg-req}}$ during TPD \rightarrow TA registration request. Considering the probability for the collision of two random choices (λ -bit length each) we obtain:

$$|Pr_{\mathcal{A}}[\text{win}_1(\mathcal{F}^{\text{vreg-req}})] - Pr_{\mathcal{A}}[\text{win}_0]| \leq \frac{1}{2^{p(\lambda)}}.$$

Game G_2 . [Pseudo-randomness of k_1, k_2 , and k_3 in $\mathcal{F}^{\text{vreg-req}}$] In this game the simulator on behalf of the TPD chooses random k_1, k_2 , and k_3 instead of computing them using the pseudo random function PRF in the vehicle TPD registration protocol $\mathcal{F}^{\text{vreg-req}}$ during TPD \rightarrow TA registration request, where the uniformly distributed secret tic was used to compute $k_1 = \text{HKDF}(1, tic)$ (secret key for encrypting Q_t), $k_2 = \text{HKDF}(2, tic)$ (secret key for generating HMAC tag), and $k_3 = \text{HKDF}(3, tic)$ (secret key for generating a HMAC tag as a reference ref). Due to the pseudo-randomness of PRF, we obtain:

$$|Pr_{\mathcal{A}}[\text{win}_2(\mathcal{F}^{\text{vreg-req}})] - Pr_{\mathcal{A}}[\text{win}_1]| \leq 3 \text{Adv}_{\mathcal{A}(\text{PRF})}^{\text{ind-sec}}.$$

Game G_3 . [HMAC Forgery in $\mathcal{F}^{\text{vreg-req}}$] In this game the simulation aborts if \mathcal{A} queries on the message $\{m \parallel_{k_2} \delta(m)\}$ in the vehicle TPD registration protocol $\mathcal{F}^{\text{vreg-req}}$ such that $_{k_2} \delta(m)$ is a valid HMAC authentication tag on a TPD \rightarrow TA registration request string $m = \{ref, _{k_1} C(Q_t), T_{vreg}, t_s\}$. To achieve this, \mathcal{A} must find a new message/tag pair, or an old message as long as the output tag was not previously attached to this message by a legitimate TPD. If \mathcal{A} can do this, we can say that \mathcal{A} breaks the SUF-CMA security of the applied HMAC scheme. Thus:

$$|Pr_{\mathcal{A}}[\text{win}_3(\mathcal{F}^{\text{vreg-req}})] - Pr_{\mathcal{A}}[\text{win}_2]| \leq \text{Succ}_{\mathcal{A}(\text{HMAC})}^{\text{suf-cma}}.$$

Game G_4 . [Indistinguishability of $_{k_1} C(Q_t)$ in $\mathcal{F}^{\text{vreg-req}}$] In this game the simulator replaces the ciphertext $_{k_1} C(Q_t)$ with $_{k_1} C(y) = \text{Encrypt}(y, k_1)$ for randomly chosen y in the vehicle TPD registration protocol $\mathcal{F}^{\text{vreg-req}}$ during TPD \rightarrow TA registration request. Note that the temporary public key Q_t is used to establish a Diffie-Hellman shared secret key by the TA when it responds to this request. Due to the IND-CPA property of (Enc,Dec) we obtain:

$$|Pr_{\mathcal{A}}[\text{win}_4(\mathcal{F}^{\text{vreg-req}})] - Pr_{\mathcal{A}}[\text{win}_3]| \leq \text{Adv}_{\mathcal{A}(\text{Enc,Dec})}^{\text{ind-cpa}}.$$

Game G_5 . [Same random secret k' in $\mathcal{F}^{\text{vreg-res}}$] In this game the simulation aborts if during the interaction the simulator on behalf of the TA chooses the same random secret $k' = b.Q_t$ in the vehicle TPD registration protocol $\mathcal{F}^{\text{vreg-res}}$ during TPD \leftarrow TA registration response. Considering the probability for the collision of two random choices (λ -bit length each) we obtain:

$$|Pr_{\mathcal{A}}[\text{win}_5(\mathcal{F}^{\text{vreg-res}})] - Pr_{\mathcal{A}}[\text{win}_4]| \leq \frac{1}{2^{p(\lambda)}}.$$

Game G_6 . [Pseudo-randomness of k'_1 and k'_2 in $\mathcal{F}^{\text{vreg-res}}$] In this game the simulator on behalf of the TA chooses random k'_1 and k'_2 instead of computing them using the pseudo random function PRF in the vehicle TPD registration protocol $\mathcal{F}^{\text{vreg-res}}$ during TPD \leftarrow TA registration response, where the random secret $k' = b.Q_t$ was used to compute $k'_1 = \text{HKDF}(1, k')$ (secret key for encrypting T'_{vreg}) and $k'_2 = \text{HKDF}(2, k')$ (secret key for generating

HMAC tag). Due to the pseudo-randomness of PRF we obtain:

$$|Pr_{\mathcal{A}}[\text{win}_6(\mathcal{F}^{\text{vreg-res}})] - Pr_{\mathcal{A}}[\text{win}_5]| \leq 2Adv_{\mathcal{A}(\text{PRF})\lambda}^{\text{ind-sec}}.$$

Game G₇. [HMAC Forgery in $\mathcal{F}^{\text{vreg-res}}$] In this game the simulation aborts if \mathcal{A} queries on the message $\{m \parallel_{k'_2} \delta(m)\}$ in the vehicle TPD registration protocol $\mathcal{F}^{\text{vreg-res}}$ such that $k'_2 \delta(m)$ is a valid HMAC authentication tag on a TPD \leftarrow TA registration response string $\{m = \text{ref}, k'_1 C(T'_{vreg}), \beta, TA_{par}, t_s\}$. To achieve this, \mathcal{A} must find a new message/tag pair, or an old message as long as the output tag was not previously attached to this message by the TA. If \mathcal{A} can do this, we can say that \mathcal{A} breaks the SUF-CMA security of the applied HMAC scheme. Thus:

$$|Pr_{\mathcal{A}}[\text{win}_7(\mathcal{F}^{\text{vreg-res}})] - Pr_{\mathcal{A}}[\text{win}_6]| \leq Succ_{\mathcal{A}(\text{HMAC})\lambda}^{\text{suf-cma}}.$$

Game G₈. [Indistinguishability of $k'_1 C(T'_{vreg})$ in $\mathcal{F}^{\text{vreg-res}}$] In this game the simulator replaces the ciphertext $k'_1 C(T'_{vreg})$ with $k'_1 C(y) = \text{Encrypt}(y, k'_1)$ for randomly chosen y in the vehicle TPD registration protocol $\mathcal{F}^{\text{vreg-res}}$ during TPD \leftarrow TA registration response. Note that the response T'_{vreg} contains the ECQV reconstruction public/private data issued by the TA. Due to the IND-CPA property of (Enc,Dec) we obtain:

$$|Pr_{\mathcal{A}}[\text{win}_8(\mathcal{F}^{\text{vreg-res}})] - Pr_{\mathcal{A}}[\text{win}_7]| \leq Adv_{\mathcal{A}(\text{Enc,Dec})\lambda}^{\text{ind-cpa}}.$$

Corollary 1. The vehicle TPD registration protocol $\mathcal{F}^{\text{vreg}}$ provides anonymity and unlinkability in the sense of randomness presented in Games G₁, G₂, G₅, and G₆. Moreover, the certificate request $T_{vreg} = \{R_v = k_v \cdot P\}$ is generated at random using a random k_v . That is, \mathcal{A} has a low probability of success in identifying a vehicle TPD among the set of all vehicles using its transmitted data. \square

Lemma 2. *The V2TA protocol $\mathcal{F}^{\text{v2ta}}$ in Γ_{ALL} (refer to Section 5.3.1) is secure in the following sense: the maximum probability of success for \mathcal{A} to forge a valid V2TA request (on behalf of the TPD) and response (on behalf of the TA) in function of the security parameter λ is:*

$$Succ_S^{\mathcal{F}^{\text{v2ta}}}(\lambda, \mathcal{A}) \leq 2 \frac{1}{2p(\lambda)} + 4Adv_{\mathcal{A}(\text{PRF})\lambda}^{\text{ind-sec}} + Succ_{\mathcal{A}(\text{Sig,Ver})\lambda}^{\text{euf-cma}} + 2Succ_{\mathcal{A}(\text{HMAC})\lambda}^{\text{suf-cma}} + 2Adv_{\mathcal{A}(\text{Enc,Dec})\lambda}^{\text{ind-cpa}}. \quad (2)$$

Proof of Lemma 2. We construct a sequence of games $G_i, i = 9, \dots, 18$ to prove Lemma 2.

Game G₉. [Same random secret k in $\mathcal{F}^{\text{v2ta-res}}$] In this game the simulation aborts if during the interaction the simulator on behalf of the TPD chooses the same random secret $k = a.Q_{ta}$ in the V2TA protocol $\mathcal{F}^{\text{v2ta-res}}$ during TPD \rightarrow TA request. Considering the probability for the collision of two random choices (λ -bit length each) we obtain:

$$|Pr_{\mathcal{A}}[\text{win}_9(\mathcal{F}^{\text{v2ta-res}})] - Pr_{\mathcal{A}}[\text{win}_8]| \leq \frac{1}{2p(\lambda)}.$$

Game G₁₀. [Pseudo-randomness of k_1 and k_2 in $\mathcal{F}^{\text{v2ta-res}}$] In this game the simulator on behalf of the TPD chooses random k_1 and k_2 instead of computing them using the pseudo random function PRF in the vehicle TPD registration protocol $\mathcal{F}^{\text{v2ta-res}}$ during TPD \rightarrow TA request,

where the random secret $k = a.Q_{ta}$ was used to compute $k_1 = HKDF(1, k)$ (secret key for encrypting T_{v2ta}) and $k_2 = HKDF(2, k)$ (secret key for generating HMAC tag). Due to the pseudo-randomness of PRF we obtain:

$$|Pr_{\mathcal{A}}[\text{win}_{10}(\mathcal{F}^{\text{v2ta-res}})] - Pr_{\mathcal{A}}[\text{win}_9]| \leq 2Adv_{\mathcal{A}(\text{PRF})\lambda}^{\text{ind-sec}}.$$

Game G₁₁. [HMAC Forgery in $\mathcal{F}^{\text{v2ta-res}}$] In this game the simulation aborts if \mathcal{A} queries on the message $\{m \parallel_{d_v} \sigma(m) \parallel_{k_2} \delta(m \parallel_{d_v} \sigma(m))\}$ in the V2TA protocol $\mathcal{F}^{\text{v2ta-res}}$ such that $k_2 \delta(m \parallel_{d_v} \sigma(m))$ is a valid HMAC authentication tag on a TPD \rightarrow TA request string $\{m = k_1 C(T_{v2ta}), \alpha, t_s \parallel_{d_v} \sigma(m)\}$. To achieve this, \mathcal{A} must find a new message/tag pair, or an old message as long as the output tag was not previously attached to this message by a legitimate TPD. If \mathcal{A} can do this, we can say that \mathcal{A} breaks the SUF-CMA security of the applied HMAC scheme. Thus:

$$|Pr_{\mathcal{A}}[\text{win}_{11}(\mathcal{F}^{\text{v2ta-res}})] - Pr_{\mathcal{A}}[\text{win}_{10}]| \leq Succ_{\mathcal{A}(\text{HMAC})\lambda}^{\text{suf-cma}}.$$

Game G₁₂. [Signature Forgery in $\mathcal{F}^{\text{v2ta-res}}$] In this game the simulation aborts if \mathcal{A} queries on the message $\{m \parallel_{d_v} \sigma(m)\}$ in the V2TA protocol $\mathcal{F}^{\text{v2ta-res}}$ such that $d_v \sigma(m)$ is a valid ECDSA signature on a TPD \rightarrow TA request string $\{m = k_1 C(T_{v2ta}), \alpha, t_s\}$. To achieve this, \mathcal{A} must find a "new" message/signature pair, which is to say that \mathcal{A} breaks the EUF-CMA security of the applied ECDSA (Sig, Ver) scheme. Thus:

$$|Pr_{\mathcal{A}}[\text{win}_{12}(\mathcal{F}^{\text{v2ta-res}})] - Pr_{\mathcal{A}}[\text{win}_{11}]| \leq Succ_{\mathcal{A}(\text{Sig,Ver})\lambda}^{\text{euf-cma}}.$$

Game G₁₃. [Indistinguishability of $k_1 C(T_{v2ta})$ in $\mathcal{F}^{\text{v2ta-res}}$] In this game the simulator replaces the ciphertext $k_1 C(T_{v2ta})$ with $k_1 C(y) = \text{Encrypt}(y, k_1)$ for randomly chosen y in the V2TA protocol $\mathcal{F}^{\text{v2ta-res}}$ during TPD \rightarrow TA request. Note that the request $T_{v2ta} = \{id_v, Q_t\}$ contains the vehicle identifier id_v and the temporary public key Q_t which is used to establish a Diffie-Hellman shared secret key by the TA when it responds to this request. Due to the IND-CPA property of (Enc,Dec) we obtain:

$$|Pr_{\mathcal{A}}[\text{win}_{13}(\mathcal{F}^{\text{v2ta-res}})] - Pr_{\mathcal{A}}[\text{win}_{12}]| \leq Adv_{\mathcal{A}(\text{Enc,Dec})\lambda}^{\text{ind-cpa}}.$$

Game G₁₄. [Same random secret k' in $\mathcal{F}^{\text{v2ta-res}}$] This game in the V2TA protocol $\mathcal{F}^{\text{v2ta-res}}$ during TPD \leftarrow TA response proceeds exactly as G₅. Thus:

$$|Pr_{\mathcal{A}}[\text{win}_{14}(\mathcal{F}^{\text{v2ta-res}})] - Pr_{\mathcal{A}}[\text{win}_{13}]| \leq \frac{1}{2p(\lambda)}.$$

Game G₁₅. [Pseudo-randomness of k'_1 and k'_2 in $\mathcal{F}^{\text{v2ta-res}}$] This game in the V2TA protocol $\mathcal{F}^{\text{v2ta-res}}$ during TPD \leftarrow TA response proceeds exactly as G₆ except that the random secret $k' = b.Q_t$ is used to compute $k'_1 = HKDF(1, k')$ (secret key for encrypting T'_{v2ta}) and $k'_2 = HKDF(2, k')$ (secret key for generating HMAC tag). Thus:

$$|Pr_{\mathcal{A}}[\text{win}_{15}(\mathcal{F}^{\text{v2ta-res}})] - Pr_{\mathcal{A}}[\text{win}_{14}]| \leq 2Adv_{\mathcal{A}(\text{PRF})\lambda}^{\text{ind-sec}}.$$

Game G₁₆. [HMAC Forgery in $\mathcal{F}^{\text{v2ta-res}}$] In this game the simulation aborts if \mathcal{A} queries on the message $\{m \parallel_{d_{ta}} \sigma(m) \parallel_{k'_2} \delta(m \parallel_{d_{ta}} \sigma(m))\}$ in the V2TA protocol $\mathcal{F}^{\text{v2ta-res}}$ such that $k'_2 \delta(m \parallel_{d_{ta}} \sigma(m))$ is a valid HMAC authentication tag on a TPD \leftarrow TA response string $\{m = k'_1 C(T'_{v2ta}), \beta, t_s \parallel_{d_{ta}} \sigma(m)\}$. To achieve this, \mathcal{A} must find

a new message/tag pair, or an old message as long as the output tag was not previously attached to this message by the TA. If \mathcal{A} can do this, we can say that \mathcal{A} breaks the SUF-CMA security of the applied HMAC scheme. Thus:

$$|Pr_{\mathcal{A}}[\text{win}_{16}(\mathcal{F}^{v2ta-res})] - Pr_{\mathcal{A}}[\text{win}_{15}]| \leq Succ_{\mathcal{A}(HMAC)\lambda}^{suf-cma}.$$

Game G₁₇. [Signature Forgery in $\mathcal{F}^{v2ta-res}$] In this game the simulation aborts if \mathcal{A} queries on the message $\{m \parallel d_v \sigma(m)\}$ in the V2TA protocol $\mathcal{F}^{v2ta-res}$ such that $d_v \sigma(m)$ is a valid ECDSA signature on a TPD \leftarrow TA response string $\{m = k'_1 C(T'_{v2ta}), \beta, t_s\}$. To achieve this, \mathcal{A} must find a “new” message/signature pair, which is to say that \mathcal{A} breaks the EUF-CMA security of the applied ECDSA (*Sig, Ver*) scheme. Thus:

$$|Pr_{\mathcal{A}}[\text{win}_{17}(\mathcal{F}^{v2ta-res})] - Pr_{\mathcal{A}}[\text{win}_{16}]| \leq Succ_{\mathcal{A}(Sig, Ver)\lambda}^{euf-cma}.$$

Game G₁₈. [Indistinguishability of $k'_1 C(T'_{v2ta})$ in $\mathcal{F}^{v2ta-res}$] In this game the simulator replaces the ciphertext $k'_1 C(T'_{v2ta})$ with $k'_1 C(y) = \text{Encrypt}(y, k'_1)$ for randomly chosen y in the V2TA protocol $\mathcal{F}^{vreg-req}$ during TPD \leftarrow TA response. Note that the response $k'_1 C(T'_{v2ta})$ may contain the daily handshaking credentials to join in the V2V communication. Due to the IND-CPA property of (Enc, Dec) we obtain:

$$|Pr_{\mathcal{A}}[\text{win}_{18}(\mathcal{F}^{v2ta-res})] - Pr_{\mathcal{A}}[\text{win}_{17}]| \leq Adv_{\mathcal{A}(Enc, Dec)\lambda}^{ind-cpa}.$$

Corollary 2. The V2TA protocol \mathcal{F}^{v2ta} provides anonymity in the sense of indistinguishability presented in Games G₁₃ and G₁₈. That is, \mathcal{A} has a low probability of success in identifying a vehicle TPD among the set of all vehicles using its transmitted data.

Corollary 3. The V2TA protocol \mathcal{F}^{v2ta} provides unlinkability in the sense of randomness presented in Games G₉, G₁₀, G₁₄, and G₁₅. That is, \mathcal{A} has a low probability of success in linking a TPD’s requests among the set of all TPDs.

Corollary 4. The V2TA protocol \mathcal{F}^{v2ta} provides non-repudiation in the sense of signature unforgeability in Games G₁₂, G₁₇. That is, no legitimate party can deny a signed and transmitted message. \square

Lemma 3. *The V2V protocol \mathcal{F}^{v2v} in Γ_{ALL} (refer to Section 5.3.3) is secure in the following sense: the maximum probability of success for \mathcal{A} to forge a valid broadcast message is a negligible function of the security parameter λ and*

$$Succ_{\mathcal{S}}^{\mathcal{F}^{v2v}}(\lambda, \mathcal{A}) \leq 2 \frac{1}{2^{p(\lambda)}} + 5 Adv_{\mathcal{A}(PRF)\lambda}^{ind-sec} + 2 Succ_{\mathcal{A}(Sig, Ver)\lambda}^{euf-cma} + 2 Succ_{\mathcal{A}(HMAC)\lambda}^{suf-cma} + 3 Adv_{\mathcal{A}(Enc, Dec)\lambda}^{ind-cpa}. \quad (3)$$

Proof of Lemma 3. We construct a sequence of games $G_i, i = 19, \dots, 29$ to prove Lemma 3.

Game G₁₉. [Same random secret k in \mathcal{F}^{v2v}] In this game the simulation aborts if during the interaction the simulator on behalf of the RSU chooses the same random secret $k = a.Q_z$ in the V2V protocol \mathcal{F}^{v2v} during TPD \leftarrow RSU broadcasting. Considering the probability for the collision of two random choices (λ -bit length each) we obtain:

$$|Pr_{\mathcal{A}}[\text{win}_{19}(\mathcal{F}^{v2v})] - Pr_{\mathcal{A}}[\text{win}_{18}]| \leq \frac{1}{2^{p(\lambda)}}.$$

Game G₂₀. [Pseudo-randomness of k_1 and k_2 in \mathcal{F}^{v2v}] In this game the simulator on behalf of the RSU chooses random k_1 and k_2 instead of computing them using the pseudo random function PRF in the V2V protocol \mathcal{F}^{v2v} during TPD \leftarrow RSU broadcasting, where the random secret $k = a.Q_z$ was used to compute $k_1 = HKDF(1, k)$ (secret key for encrypting $d_{Br_{rsu}}$) and $k_2 = HKDF(2, k)$ (secret key for generating HMAC tag). Due to the pseudo-randomness of PRF we obtain:

$$|Pr_{\mathcal{A}}[\text{win}_{20}(\mathcal{F}^{v2v})] - Pr_{\mathcal{A}}[\text{win}_{19}]| \leq 2 Adv_{\mathcal{A}(PRF)\lambda}^{ind-sec}.$$

Game G₂₁. [HMAC Forgery in \mathcal{F}^{v2v}] In this game the simulation aborts if \mathcal{A} queries on the message $\{m \parallel d_{rsu} \sigma(m) \parallel k_2 \delta(m \parallel d_{rsu} \sigma(m))\}$ in the V2V protocol \mathcal{F}^{v2v} such that $k_2 \delta(m \parallel d_{rsu} \sigma(m))$ is a valid HMAC authentication tag on a TPD \leftarrow RSU broadcast string $\{m = im Cert_{rsu}, k_1 C(d_{Br_{rsu}}), \alpha, t_s \parallel d_{rsu} \sigma(m)\}$. To achieve this, \mathcal{A} must find a new message/tag pair, or an old message as long as the output tag was not previously attached to this message by a legitimate RSU. If \mathcal{A} can do this, we can say that \mathcal{A} breaks the SUF-CMA security of the applied HMAC scheme. Thus:

$$|Pr_{\mathcal{A}}[\text{win}_{21}(\mathcal{F}^{v2v})] - Pr_{\mathcal{A}}[\text{win}_{20}]| \leq Succ_{\mathcal{A}(HMAC)\lambda}^{suf-cma}.$$

Game G₂₂. [Signature Forgery in \mathcal{F}^{v2v}] In this game the simulation aborts if \mathcal{A} queries on the message $\{m \parallel d_{rsu} \sigma(m)\}$ in the V2V protocol \mathcal{F}^{v2v} such that $d_{rsu} \sigma(m)$ is a valid ECDSA signature on a TPD \leftarrow RSU broadcast string $\{m = im Cert_{rsu}, k_1 C(d_{Br_{rsu}}), \alpha, t_s\}$. To achieve this, \mathcal{A} must find a “new” message/signature pair, which is to say that \mathcal{A} breaks the EUF-CMA security of the applied ECDSA (*Sig, Ver*) scheme. Thus:

$$|Pr_{\mathcal{A}}[\text{win}_{22}(\mathcal{F}^{v2v})] - Pr_{\mathcal{A}}[\text{win}_{21}]| \leq Succ_{\mathcal{A}(Sig, Ver)\lambda}^{euf-cma}.$$

Game G₂₃. [Indistinguishability of $k_1 C(d_{Br_{rsu}})$ in \mathcal{F}^{v2v}] In this game the simulator replaces the ciphertext $k_1 C(d_{Br_{rsu}})$ with $k_1 C(y) = \text{Encrypt}(y, k_1)$ for randomly chosen y in the V2V protocol \mathcal{F}^{v2v} during TPD \leftarrow RSU broadcasting. Note that the RSU’s broadcast-key $d_{Br_{rsu}}$, along with the daily handshaking credentials, is used to establish a Diffie-Hellman shared secret key in the V2V communication (TPD \rightarrow TPD). Due to the IND-CPA property of (Enc, Dec) we obtain:

$$|Pr_{\mathcal{A}}[\text{win}_{23}(\mathcal{F}^{v2v})] - Pr_{\mathcal{A}}[\text{win}_{22}]| \leq Adv_{\mathcal{A}(Enc, Dec)\lambda}^{ind-cpa}.$$

Game G₂₄. [Same random secret k' in \mathcal{F}^{v2v}] This game in the V2V protocol \mathcal{F}^{v2v} during TPD \rightarrow TPD broadcasting proceeds exactly as G₆ except that the random secret is $k' = (b \times d_{Bta}).Q_{Br_{rsu}}$. Thus:

$$|Pr_{\mathcal{A}}[\text{win}_{24}(\mathcal{F}^{v2v})] - Pr_{\mathcal{A}}[\text{win}_{23}]| \leq \frac{1}{2^{p(\lambda)}}.$$

Game G₂₅. [Pseudo-randomness of $k'_1, k'_2,$ and k''_1 in \mathcal{F}^{v2v}] In this game the simulator on behalf of the TPD chooses random $k'_1, k'_2,$ and k''_1 instead of computing them using the pseudo random function PRF in the V2V protocol \mathcal{F}^{v2v} during TPD \rightarrow TPD broadcast communication, where the secret random $k' = (b \times d_{Bta}).Q_{Br_{rsu}}$ was used to compute $k'_1 = HKDF(1, k')$ (secret key for encrypting *beacon*) and $k'_2 = HKDF(2, k')$ (secret key for generating

HMAC tag). The secret key $k'' = b.Q_{ta}$ is used to compute $k'_1 = HKDF(1, k'')$ (secret key for encrypting id_v). Due to the pseudo-randomness of PRF, we obtain:

$$|Pr_{\mathcal{A}}[\text{win}_{25}(\mathcal{F}^{v2v})] - Pr_{\mathcal{A}}[\text{win}_{24}]| \leq 3Adv_{\mathcal{A}(PRF)\lambda}^{\text{ind-sec}}.$$

Game G₂₆. [HMAC Forgery in \mathcal{F}^{v2v}] In this game the simulation aborts if \mathcal{A} queries on the message $\{m \parallel_{d_v} \sigma(m) \parallel_{k'_2} \delta(m \parallel_{d_v} \sigma(m))\}$ in the V2V protocol \mathcal{F}^{v2v} such that $_{k'_2} \delta(m \parallel_{d_v} \sigma(m))$ is a valid HMAC authentication tag on a TPD→TPD V2V broadcast string $\{m =_{k'_1} C(id_v),_{k'_1} C(beacon), \beta, t_s \parallel_{d_v} \sigma(m)\}$. To achieve this, \mathcal{A} must find a new message/tag pair, or an old message as long as the output tag was not previously attached to this message by a legitimate TPD. If \mathcal{A} can do this, we can say that \mathcal{A} breaks the SUF-CMA security of the applied HMAC scheme. Thus:

$$|Pr_{\mathcal{A}}[\text{win}_{26}(\mathcal{F}^{v2v})] - Pr_{\mathcal{A}}[\text{win}_{25}]| \leq Succ_{\mathcal{A}(HMAC)\lambda}^{\text{suf-cma}}.$$

Game G₂₇. [Signature Forgery in \mathcal{F}^{v2v}] In this game the simulation aborts if \mathcal{A} queries on the message $\{m \parallel_{d_v} \sigma(m)\}$ in the V2V protocol \mathcal{F}^{v2v} such that $_{d_v} \sigma(m)$ is a valid ECDSA signature on a TPD→TPD V2V broadcast string $\{m =_{k'_1} C(id_v),_{k'_1} C(beacon), \beta, t_s\}$. To achieve this, \mathcal{A} must find a “new” message/signature pair, which is to say that \mathcal{A} breaks the EUF-CMA security of the applied ECDSA (*Sig, Ver*) scheme. Thus:

$$|Pr_{\mathcal{A}}[\text{win}_{27}(\mathcal{F}^{v2v})] - Pr_{\mathcal{A}}[\text{win}_{26}]| \leq Succ_{\mathcal{A}(Sig, Ver)\lambda}^{\text{euf-cma}}.$$

Game G₂₈. [Indistinguishability of $_{k'_1} C(beacon)$ in \mathcal{F}^{v2v}] In this game the simulator replaces the ciphertext $_{k'_1} C(beacon)$ with $_{k'_1} C(y) = \text{Encrypt}(y, k'_1)$ for randomly chosen y in the V2V protocol \mathcal{F}^{v2v} during TPD→TPD broadcasting. Due to the IND-CPA property of (Enc,Dec) we obtain:

$$|Pr_{\mathcal{A}}[\text{win}_{28}(\mathcal{F}^{v2v})] - Pr_{\mathcal{A}}[\text{win}_{27}]| \leq Adv_{\mathcal{A}(Enc, Dec)\lambda}^{\text{ind-cpa}}.$$

Game G₂₉. [Indistinguishability of $_{k'_1} C(id_v)$ in \mathcal{F}^{v2v}] In this game the simulator replaces the ciphertext $_{k'_1} C(id_v)$ with $_{k'_1} C(y) = \text{Encrypt}(y, k'_1)$ for randomly chosen y in the V2V protocol \mathcal{F}^{v2v} during TPD→TPD broadcasting. Due to the IND-CPA property of (Enc,Dec) we obtain:

$$|Pr_{\mathcal{A}}[\text{win}_{29}(\mathcal{F}^{v2v})] - Pr_{\mathcal{A}}[\text{win}_{28}]| \leq Adv_{\mathcal{A}(Enc, Dec)\lambda}^{\text{ind-cpa}}.$$

Corollary 5. The V2V protocol \mathcal{F}^{v2v} provides anonymity in the sense of indistinguishability presented in Games G₂₈ and G₂₉. That is, \mathcal{A} has a low probability of success in identifying a vehicle TPD among the set of all vehicles using its transmitted data.

Corollary 6. The V2V protocol \mathcal{F}^{v2v} provides unlinkability in the sense of randomness presented in Games G₂₄, G₂₅. That is, \mathcal{A} has a low probability of success in linking a vehicle broadcast messages among the set of all vehicles.

Corollary 7. The V2V protocol \mathcal{F}^{v2v} provides non-repudiation in the sense of signature unforgeability in Game G₂₂. That is, no legitimate TPD can deny a signed message. \square

Lemma 4. *The vehicle key-update protocol \mathcal{F}^{vkup} in Γ_{ALI} (refer to Section 5.5) is secure in the following sense: the maximum*

probability of success for \mathcal{A} to forge a valid key-update request (on behalf of the TPD) and response (on behalf of the TA) in function of the security parameter λ is:

$$Succ_{\mathcal{S}}^{\mathcal{F}^{vkup}}(\lambda, \mathcal{A}) \leq 2\frac{1}{2^{p(\lambda)}} + 5Adv_{\mathcal{A}(PRF)\lambda}^{\text{ind-sec}} + 2Succ_{\mathcal{A}(HMAC)\lambda}^{\text{suf-cma}} + 2Adv_{\mathcal{A}(Enc, Dec)\lambda}^{\text{ind-cpa}}. \quad (4)$$

Proof of Lemma 4. We construct a sequence of games $G_i, i = 30, \dots, 37$ to prove Lemma 4.

Game G₃₀. [Same random secret k_{up} in $\mathcal{F}^{vkup-req}$] In this game the simulation aborts if during the interaction the simulator on behalf of the TPD chooses random k_{up} in the vehicle key-update protocol $\mathcal{F}^{vkup-req}$ during TPD→TA key-update request. Considering the probability for the collision of two random choices (λ -bit length each) we obtain:

$$|Pr_{\mathcal{A}}[\text{win}_{30}(\mathcal{F}^{vkup-req})] - Pr_{\mathcal{A}}[\text{win}_{29}]| \leq \frac{1}{2^{p(\lambda)}}.$$

Game G₃₁. [Pseudo-randomness of k_1, k_2 , and k_3 in $\mathcal{F}^{vkup-req}$] This game in the vehicle key-update protocol $\mathcal{F}^{vkup-req}$ during TPD→TA key-update request proceeds exactly as G₂ except that the random secret k_{up} is used to compute k_1, k_2 , and k_3 . Thus:

$$|Pr_{\mathcal{A}}[\text{win}_{31}(\mathcal{F}^{vkup-req})] - Pr_{\mathcal{A}}[\text{win}_{30}]| \leq 3Adv_{\mathcal{A}(PRF)\lambda}^{\text{ind-sec}}.$$

Game G₃₂. [HMAC Forgery in $\mathcal{F}^{vkup-req}$] In this game the simulation aborts if \mathcal{A} queries on the message $\{m \parallel_{k_2} \delta(m)\}$ in the vehicle key-update protocol $\mathcal{F}^{vkup-req}$ such that $_{k_2} \delta(m)$ is a valid HMAC authentication tag on a TPD→TA key-update request string $\{m = ref,_{k_1} C(Q_t), T_{vkup}, t_s\}$. To achieve this, \mathcal{A} must find a new message/tag pair, or an old message as long as the output tag was not previously attached to this message by a legitimate TPD. If \mathcal{A} can do this, we can say that \mathcal{A} breaks the SUF-CMA security of the applied HMAC scheme. Thus:

$$|Pr_{\mathcal{A}}[\text{win}_{32}(\mathcal{F}^{vkup-req})] - Pr_{\mathcal{A}}[\text{win}_{31}]| \leq Succ_{\mathcal{A}(HMAC)\lambda}^{\text{suf-cma}}.$$

Game G₃₃. [Indistinguishability of $_{k_1} C(Q_t)$ in $\mathcal{F}^{vkup-req}$] In this game the simulator replaces the ciphertext $_{k_1} C(Q_t)$ with $_{k_1} C(y) = \text{Encrypt}(y, k_1)$ for randomly chosen y in the vehicle key-update protocol $\mathcal{F}^{vkup-req}$ during TPD→TA key-update request. Due to the IND-CPA property of (Enc,Dec) we obtain:

$$|Pr_{\mathcal{A}}[\text{win}_{33}(\mathcal{F}^{vkup-req})] - Pr_{\mathcal{A}}[\text{win}_{32}]| \leq Adv_{\mathcal{A}(Enc, Dec)\lambda}^{\text{ind-cpa}}.$$

Game G₃₄. [Same random secret k' in $\mathcal{F}^{vkup-res}$] This game in the vehicle key-update protocol $\mathcal{F}^{vkup-res}$ during TPD←TA key-update response proceeds exactly as G₅. Thus:

$$|Pr_{\mathcal{A}}[\text{win}_{34}(\mathcal{F}^{vkup-res})] - Pr_{\mathcal{A}}[\text{win}_{33}]| \leq \frac{1}{2^{p(\lambda)}}.$$

Game G₃₅. [Pseudo-randomness of k'_1 and k'_2 in $\mathcal{F}^{vkup-res}$] This game in the vehicle key-update protocol $\mathcal{F}^{vkup-res}$ during TPD←TA key-update response proceeds exactly as G₆. Thus:

$$|Pr_{\mathcal{A}}[\text{win}_{35}(\mathcal{F}^{vkup-res})] - Pr_{\mathcal{A}}[\text{win}_{34}]| \leq 2Adv_{\mathcal{A}(PRF)\lambda}^{\text{ind-sec}}.$$

Game G₃₆. [HMAC Forgery in $\mathcal{F}^{vkup-res}$] In this game the simulation aborts if \mathcal{A} queries on the message $\{m \parallel_{k'_2} \delta(m)\}$ in the vehicle key-update protocol

$\mathcal{F}^{\text{vkup-res}}$ such that $k'_2 \delta(m)$ is a valid HMAC authentication tag on a TPD \leftarrow TA key-update response string $\{m = \text{ref}, k'_1 C(T'_{\text{vkup}}), \beta, T A_{\text{par}}, t_s\}$. To achieve this, \mathcal{A} must find a new message/tag pair, or an old message as long as the output tag was not previously attached to this message by a legitimate TPD. If \mathcal{A} can do this, we can say that \mathcal{A} breaks the SUF-CMA security of the applied HMAC scheme. Thus:

$$|Pr_{\mathcal{A}}[\text{win}_{36}(\mathcal{F}^{\text{vkup-res}})] - Pr_{\mathcal{A}}[\text{win}_{35}]| \leq Succ_{\mathcal{A}(\text{HMAC})\lambda}^{\text{suf-cma}}.$$

Game G₃₇. [Indistinguishability of $k'_1 C(T'_{\text{vkup}})$ in $\mathcal{F}^{\text{vkup-res}}$] In this game the simulator replaces the ciphertext $k'_1 C(T'_{\text{vkup}})$ with $k'_1 C(y) = \text{Encrypt}(y, k'_1)$ for randomly chosen y in the vehicle key-update protocol $\mathcal{F}^{\text{vkup-res}}$ during TPD \leftarrow TA response. Note that T'_{vkup} contains new key-material. Due to the IND-CPA property of (Enc,Dec) we obtain:

$$|Pr_{\mathcal{A}}[\text{win}_{37}(\mathcal{F}^{\text{vkup-res}})] - Pr_{\mathcal{A}}[\text{win}_{36}]| \leq Adv_{\mathcal{A}(\text{Enc,Dec})\lambda}^{\text{ind-cpa}}.$$

Corollary 8. The vehicle key-update protocol $\mathcal{F}^{\text{vkup}}$ provides anonymity and unlinkability in the sense of randomness presented in Games G₃₀, G₃₁, G₃₄, and G₃₅. Moreover, the certificate request $T_{\text{vkup}} = \{R_v = k_v.P\}$ is generated at random using a random k_v . That is, \mathcal{A} has a low probability of success in identifying a vehicle TPD among the set of all vehicles using its transmitted data. \square

The probability of \mathcal{A} in winning Game G₃₇ is:

$$Pr_{\mathcal{A}}[\text{win}_{37}] \leq \begin{cases} 0 \\ 2^{-p(\lambda)} \end{cases}.$$

Combining Equations 1, 2, 3, and 4 yields the result stated as the following theorem. \square

Theorem 9. *If PRF is pseudo random, (Enc,Dec) is IND-CPA secure, Hash is collision-resistant, (Sig,Ver) is EUF-CMA secure, and HMAC is SUF-CMA secure (as defined in Section 4), then Γ_{ALI} is secure in the following sense: the maximum probability of success for \mathcal{A} to break the security of Γ_{ALI} (as defined in Section 3.3) is a negligible function of the security parameter λ as follows:*

$$Succ_{\mathcal{S}}^{\Gamma_{\text{ALI}}}(\lambda, \mathcal{A}) \leq Succ_{\mathcal{S}}^{\mathcal{F}^{\text{vreg}}}(\lambda, \mathcal{A}) + Succ_{\mathcal{S}}^{\mathcal{F}^{\text{v2ta}}}(\lambda, \mathcal{A}) + Succ_{\mathcal{S}}^{\mathcal{F}^{\text{v2v}}}(\lambda, \mathcal{A}) + Succ_{\mathcal{S}}^{\mathcal{F}^{\text{vkup}}}(\lambda, \mathcal{A}). \quad (5)$$

6.3 Security and Privacy Comparisons

The scheme presented by Raya and Hubaux [12] does not provide unlinkability because a vehicle changes its anonymous certificate and private key only after having used them for multiple messages [23].

Studer et al.'s scheme [14] does not provide unlinkability and identity privacy, as a vehicle broadcasts its fixed certificate once a second [43].

Wang et al. [15] propose a system-key update protocol. This procedure is performed if a cryptographic key breach occurs, or if a cryptographic key is unknowingly accessed. However, the procedure uses an old system key to encrypt and decrypt a new system key (key wrapping). This is absolutely a vulnerability as it produces dependencies among keying material. If a past system key is compromised, a new system key is very easily compromised.

Camenisch et al. [18] rely on symmetric authenticated encryption using temporary keys which are exchanged among vehicles. As the scheme relies on symmetric-key cryptography, it cannot provide non-repudiation, and as a result, no proper traceability and revocation could be performed in the case of malicious activity. A malicious TPD can deny the message transmission, so that the TA is unable to trace it.

Except for the ALI and Wang et al. [15] schemes, none of the schemes listed in Table 1 take a system-key update protocol into consideration (for long-term cryptographic keying material). However, the system-key update protocol presented by Wang et al. [15] is vulnerable to a known-key attack.

The master-key and broadcast-key used in the ALI scheme are periodically updated (e.g., 24 hours and 10 minutes, refer to Section 5.3.2 and Section 5.3.3). This passive revocation approach requires vehicles to regularly request new short-term credentials. These requests will be rejected if the corresponding long-term credential has been revoked. Moreover, ALI satisfies the perfect forward secrecy security requirement and provides a secure key-update mechanism that does not produce dependencies among keying material (it does not rely on the old credentials to receive new ones).

Similar to the scheme presented by Camenisch et al. [18], ALI focuses on authentication with beacon content encryption, where the encrypted beacons and authentication tags are generated using different random keys for each new message. This approach prevents an observer from identifying which beacon to link to a particular vehicle. As a result, ALI fully satisfies the unlinkability requirement. The other schemes listed in Table 1 only focus on authentication without considering beacon content encryption, and as a result, they fail to provide unlinkability.

TABLE 1
Comparisons of Security and Privacy Properties

Properties	Schemes						
	[12]	[13]	[14]	[15]	[17]	[18]	ALI
Authentication	✓	✓	✓	✓	✓	✓	✓
Identity privacy	✓	✓	×	✓	✓	✓	✓
Unlinkability	×	×	×	×	×	✓	✓
Revocation	✓	✓	×	✓	✓	×	✓
Non-repudiation	✓	✓	✓	✓	✓	×	✓
Secure key-update	×	×	×	×	×	×	✓

—Note: The property is satisfied [✓]. The property is not satisfied [×].

ALI uses an EUF-CMA secure inner ECDSA signature on a message to ensure source authentication and non-repudiation in V2TA and V2V communications (refer to Section 4.2). To ensure integrity in different phases, ALI uses the encrypt-then-MAC method [44] where the encryption scheme is IND-CPA secure (refer to Section 4.1.1) and the MAC is SUF-CMA secure (refer to Section 4.1.2). This method implies that decryption will only be carried on ciphertext that passed an integrity test. The adversary thus cannot observe use of the decryption key on ciphertext which is chosen randomly, since that ciphertext will not pass the MAC verification and will not be decrypted. Hence, the ECIES used in ALI should provide Indistinguishabil-

ity under Chosen-Ciphertext Attacks (IND-CCA) [20]: the adversary has an additional capability over the IND-CPA, calling an encryption or decryption oracle for encrypting or decrypting arbitrary messages before obtaining the challenge ciphertext.

7 PERFORMANCE ANALYSIS

The investigation covers the Average Computation Time (ACT) of different operations providing ≈ 128 -bit security level used in the aforementioned schemes, including: Elliptic Curve Cryptography (ECC) base point scalar multiplication (T_{mul}^{scal}), ECDSA-256 signature generation (T_{sign}^{ecdsa}) and verification (T_{ver}^{ecdsa}), ECQV-256 public-key certificate reconstruction (T_{rec}^{ecqv}), AES-128 CTR mode encryption (T_{enc}^{aes}) and decryption (T_{dec}^{aes}), HKDF-256 (T_{hkd}^{hkd}), SHA-256 hash (T_{sha}^{hash}), HMAC-256 (T_{sha}^{hmac}), and optimal Ate pairing-382 (T_{pair}^{oate}). The investigation is performed using the newest stable release branch (1.1.1 series) of the OpenSSL software library [45]. All ECC operations (except for GISIS) are evaluated over the NIST P-256 curve (achieving 128-bit security level [46]). As GISIS is a pairing-based scheme [47], [48], the elliptic curve arithmetic and optimal Ate pairing computations [49] are over the Barreto-Naehrig (BN) curve [50] in the evaluation, where the minimum bit-length of p is estimated as 382 bits (achieving 127-bit security level [51]), an optimistic parameter to use in cryptographic pairings for 128-bit security level (384 bits p) [25]. The implementation for GISIS is performed using the newest efficient and stable release branch (0.5.0 series) of the RELIC cryptographic meta-toolkit [52]. The experiment for each cryptographic operation involved 10^6 trials in Debian Linux distribution running on an Intel Core 2 Duo Processor T6570 (2Megabytes Cache, 2.10 GHz, 4GB RAM). To have more useful results, this research practically estimates the cryptographic overhead of the above-mentioned schemes on packet size. Table 2 lists the results of investigation.

7.1 Computational Overhead

The message generation cost comparison is shown in Table 3. The ALI needs four scalar multiplications, three HKDF calculations, two AES encryptions, one HMAC computation, and one ECDSA signature generation. After Lin et al. [13], the ALI scheme has the highest broadcast message signing cost among the other schemes, approximately 0.72 ms. Note that vehicles broadcast one beacon in each short time frame. That is, the signature verification speed is far more important than signature generation speed.

The message verification cost comparisons is shown in Table 3. After Wang et al. [15] and Huang et al. [17] which have similar verification costs, the ALI scheme has the lowest broadcast message verification delay among the other schemes (approximately 0.15 ms). The ALI scheme can approximately verify 7 messages in one millisecond, and 700 messages each 100 ms. This satisfies verification of the upper bound of the received beacons (222 beacons) discovered in the simulation studies presented by Bae et al. [10], [16]. Note that message verification in the schemes presented by Raya and Hubaux [12], Studer et al. [14], and Bae et al. [16] includes CRL checking through string comparison which could be ignored.

7.2 Communication Overhead

Table 4 shows the communication overhead for one message across the schemes discussed in this paper. The V2V communication in the ALI scheme (refer to Section 5.3.3) produces 149 bytes of overhead in total, including: one AES ciphertext $k_1' C(id_v) = 17$ bytes (for 17 digits id_v), one ECIES parameter (ECC point in compressed size) $\beta = 32$ bytes, one time stamp $t_s = 4$ bytes, one ECDSA signature $d, \sigma(m) = 64$ bytes, one HMAC $k_2' \delta(m \parallel d, \sigma(m)) = 32$ bytes. Note that the AES-CTR encrypted beacon $k_1' C(beacon)$ has a size similar to a beacon before encryption, for example $k_1' C(beacon) = 100$ bytes where $beacon = 100$ bytes. For this reason, the encrypted beacon size is not included in communication overhead. Please refer to Table 2 to find the cryptographic overhead corresponding to each payload.

TABLE 4
Comparisons of V2V Authentication Overhead on Packet Size

Properties	Schemes							
	[12]	[13]	[14]	[15]	[16]	[17]	[18]	ALI
Size (byte)	128	432	160	64	97	64	240	149

7.3 Packet Loss Ratio (PLR) Evaluation

This section focuses on a scenario representing the case of safety-critical message broadcasting to support coordination and awareness between vehicles on the highway. The scenario considers a uniform presence of vehicles moving on a highway with the number of 12 lanes (6 in each direction), where each lane is 3 m wide, similar to the simulation study presented by [16]. It is assumed that vehicles travel at a fixed speed 30 m/s (108 Km/h) with an inter-vehicle space 30 m. Moving vehicles generate, sign, and broadcast 249 bytes of messages containing 149 bytes of authentication overhead plus 100 bytes of encrypted beacon, every 100 ms (update rate of 10 hertz) over a 300 m communication range for 186 seconds. The scenario also considers an RSU that transmits broadcast-keys each 100 ms to the roadside, and the two vehicles v_A and v_B , located in the middle of the highway. The vehicles v_A and v_B correspond to a maximum of received messages from 240 vehicles in their communication range. For the two vehicles, v_A and v_B , the safety-critical messages are signed and verified using the ALI V2V broadcast authentication scheme providing 128-bit security level.

This study uses Veins [53], an open source vehicular network simulation framework. It utilizes the models provided in the OMNeT++ discrete event simulator [54] and SUMO simulation of urban mobility [55] for network simulation and vehicular movement, respectively. The scenario is implemented in the American IEEE 1609 C-ITS standard using the detailed models presented by Eckhoff and Sommer [56]. The beacons are generated in accordance with IEEE 802.11p physical layer and medium access control specification [57], similar to the simulation study presented by [16].

Let pl denote the number of lost packets, N_{tx} denote the number of broadcast packets from v_B , and N_{rx} denote the number of received packets from v_A . Then, pl will be

TABLE 2
The Average Computation Time and Cryptographic Overhead of Different Operations

	$scal_{mul}$	$ecdsa_{sign}$	$ecdsa_{ver}$	$ecqv_{rec}$	aes_{enc}	aes_{dec}	$hkdf_{sha}$	$hash_{sha}$	$hmac_{sha}$	$oate_{pair}$
ACT (sec)	1.4×10^{-4}	1.5×10^{-4}	3.4×10^{-4}	2.2×10^{-4}	6.8×10^{-7}	8.9×10^{-7}	3.7×10^{-6}	4.8×10^{-7}	3.7×10^{-6}	5.4×10^{-3}
Overhead (byte)	32	64	-	32	-	-	-	32	32	48

TABLE 3
The Computational Overhead of Different V2V Authentication Schemes During Message Generation and Verification

Schemes	Message Generation Overhead (sec)	Message Verification Overhead (sec)
[12]	$T_{sign}^{ecdsa} = 1.5 \times 10^{-4}$	$6.8 \times 10^{-4} = T_{ver}^{ecdsa} + cert T_{ver}^{ecdsa}$
[13]	$3T_{pair}^{oate} + T_{sha}^{hash} = 1.62 \times 10^{-2}$	$2.7 \times 10^{-2} + x^* = 5T_{pair}^{oate} + T_{sha}^{hash} + (2 \times \sum_{i=1}^{crl} T_{pair}^{oate})$
[14]	$T_{sign}^{ecdsa} + T_{sha}^{hmac} = 1.53 \times 10^{-4}$	$6.87 \times 10^{-4} = T_{ver}^{ecdsa} + cert T_{ver}^{ecdsa} + 2T_{sha}^{hmac}$
[15]	$7T_{sha}^{hash} + T_{sha}^{hmac} = 7.06 \times 10^{-6}$	$4.18 \times 10^{-6} = T_{sha}^{hash} + T_{sha}^{hmac}$
[16]	$T_{sign}^{ecdsa} = 1.5 \times 10^{-4}$	$5.6 \times 10^{-4} = T_{ver}^{ecdsa} + T_{rec}^{ecqv}$
[17]	$T_{sha}^{hash} + T_{sha}^{hmac} = 4.18 \times 10^{-6}$	$4.18 \times 10^{-6} = T_{sha}^{hash} + T_{sha}^{hmac}$
[18]	$3T_{enc}^{aes} = 2.04 \times 10^{-6}$	$2.46 \times 10^{-6} = T_{enc}^{aes} + 2T_{dec}^{aes}$
ALI	$4T_{mul}^{scal} + 3T_{sha}^{hkdf} + 2T_{enc}^{aes} + T_{sha}^{hmac} + T_{sign}^{ecdsa} = 7.26 \times 10^{-4}$	$1.52 \times 10^{-4} = T_{mul}^{scal} + 2T_{sha}^{hkdf} + T_{dec}^{aes} + T_{sha}^{hmac}$

[x^*] The value $x = (2 \times \sum_{i=1}^{crl} T_{pair}^{oate})$, where crl denotes the number of elements in the CRL.

$pl = N_{tx} - N_{rx}$, and PLR will be $PLR = (pl \times 100)/N_{tx}$, in 100 ms. Simulation results show that both vehicles v_A and v_B arrived to the highway after 8 seconds, and have sent 1772 messages and received 1769 messages during 186 seconds simulation time. That is, the ALI V2V broadcast communication has 0.16% PLR for 249 bytes of message length. The simulation represented in this section is repeated to compare the recorded PLR with those in Raya and Hubaux [12], Lin et al. [13], Studer et al. [14], Wang et al. [15], Bae et al. [16], Huang et al. [17], and Camenisch et al. [18]. Results are almost identical, $\approx 0.1\%$.

8 CONCLUSION

This paper presented a new secure, efficient, and privacy-preserving scheme: ALI broadcast authentication with encryption. ALI combined message authentication with beacon encryption to provide a higher level of anonymity compared to existing schemes. ALI uses public-key functionality via symmetric techniques, enabled by in-vehicle TPDs, permitting authorized vehicles to anonymously exchange beacons with other vehicles within communication range. This anonymity was conditional: a trusted authority can resolve disputes and track malicious vehicles. ALI can be used for both V2I and V2V communications. Moreover, this novel scheme utilized the ECQV public/private key pair in the C-ITS protocol design, for the first time. ALI has low computational and communication overhead compared to the schemes based on pseudonymous certificates and group signatures. The in-depth security analysis supported by formal security proof demonstrated that ALI can effectively satisfy the security (IND-CCA) and privacy requirements. The scheme produced only 149 bytes of cryptographic overhead and handled authentication of approximately 700

broadcast messages every 100 ms on a 2.10 GHz Intel Core 2 Duo Processor, suitable for heavy traffic conditions.

REFERENCES

- [1] M. A. R. Bae, *Privacy-preserving authentication and key management for cooperative intelligent transportation systems*. PhD thesis, Queensland University of Technology, 2021.
- [2] P. Papadimitratos, A. D. La Fortelle, K. Evenssen, R. Brignolo, and S. Cosenza, "Vehicular communication systems: Enabling technologies, applications, and future outlook on intelligent transportation," *IEEE Communications Magazine*, vol. 47, pp. 84–95, November 2009.
- [3] S. Goudarzi, A. H. Abdullah, S. Mandala, S. A. Soleymani, M. A. R. Bae, M. H. Anisi, and M. S. Aliyu, "A systematic review of security in vehicular ad hoc network," in *Proc. 2nd Symp. WSCN*, pp. 1–10, 2013.
- [4] M. A. R. Bae, L. Simpson, X. Boyen, E. Foo, and J. Pieprzyk, "Authentication strategies in vehicular communications: a taxonomy and framework," *EURASIP Journal on Wireless Communications and Networking*, vol. 2021, no. 1, pp. 1–50, 2021.
- [5] L. Reyzin et al., "RE: Comments on NHTSA notice of proposed rule for FMVSS No. 150," *V2V Communications (Docket No. NHTSA-2016-0126)*, 2017.
- [6] S. A. Soleymani, A. H. Abdullah, W. H. Hassan, M. H. Anisi, S. Goudarzi, M. A. Rezazadeh Bae, and S. Mandala, "Trust management in vehicular ad hoc network: a systematic review," *EURASIP Journal on Wireless Communications and Networking*, vol. 2015, p. 146, May 2015.
- [7] K. Han, A. Weimerskirch, and K. G. Shin, "Automotive cybersecurity for in-vehicle communication," in *IQT QUARTERLY*, vol. 6 of 1, pp. 22–25, 2014.
- [8] J. Siegel, D. Erb, and S. Sarma, "Algorithms and architectures: A case study in when, where and how to connect vehicles," *IEEE Intelligent Transportation Systems Magazine*, vol. 10, pp. 74–87, Spring 2018.
- [9] "NXP Semiconductors; RoadLINK SAF5400 single chip modem." <https://www.nxp.com>. Accessed: 2020-02-10.
- [10] M. A. R. Bae, L. Simpson, X. Boyen, E. Foo, and J. Pieprzyk, "A model to evaluate reliability of authentication protocols in C-ITS safety-critical applications," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 9, pp. 9306–9319, 2021.

- [11] CAMP, "Vehicle Safety Communications Project: Task 3 Final Report - Identify Intelligent Vehicle Safety Applications Enabled by DSRC," Tech. Rep. DOT HS 809 859, National Highway Traffic Safety Administration - U. S. Department of Transportation (USDOT), March 2005.
- [12] M. Raya and J.-P. Hubaux, "The security of vehicular ad hoc networks," in *Proceedings of the 3rd ACM Workshop on Security of Ad Hoc and Sensor Networks, SASN '05*, (New York, NY, USA), pp. 11–21, ACM, 2005.
- [13] X. Lin, X. Sun, P. Ho, and X. Shen, "GSIS: A secure and privacy-preserving protocol for vehicular communications," *IEEE Transactions on Vehicular Technology*, vol. 56, pp. 3442–3456, Nov 2007.
- [14] A. Studer, F. Bai, B. Bellare, and A. Perrig, "Flexible, extensible, and efficient VANET authentication," *Journal of Communications and Networks*, vol. 11, pp. 574–588, Dec 2009.
- [15] F. Wang, Y. Xu, H. Zhang, Y. Zhang, and L. Zhu, "2FLIP: A two-factor lightweight privacy-preserving authentication scheme for VANET," *IEEE Transactions on Vehicular Technology*, vol. 65, pp. 896–911, Feb 2016.
- [16] M. A. R. Bae, L. Simpson, E. Foo, and J. Pieprzyk, "Broadcast authentication in latency-critical applications: On the efficiency of IEEE 1609.2," *IEEE Transactions on Vehicular Technology*, vol. 68, pp. 11577–11587, Dec 2019.
- [17] J. Huang, Y. Qian, and R. Q. Hu, "Secure and efficient privacy-preserving authentication scheme for 5G software defined vehicular networks," *IEEE Transactions on Vehicular Technology*, 2020.
- [18] J. Camenisch, M. Drijvers, A. Lehmann, G. Neven, and P. Towa, "Zone encryption with anonymous authentication for V2V communication," in *2020 IEEE European Symposium on Security and Privacy (EuroS P)*, pp. 405–424, 2020.
- [19] A. J. Menezes, S. A. Vanstone, and P. C. V. Oorschot, *Handbook of Applied Cryptography*. Boca Raton, FL, USA: CRC Press, Inc., 1st ed., 1996.
- [20] M. Bellare and C. Namprempre, "Authenticated encryption: Relations among notions and analysis of the generic composition paradigm," in *Advances in Cryptology — ASIACRYPT 2000* (T. Okamoto, ed.), (Berlin, Heidelberg), pp. 531–545, Springer Berlin Heidelberg, 2000.
- [21] M. J. Dworkin, "Sp 800-38c. recommendation for block cipher modes of operation: The ccm mode for authentication and confidentiality," 2004.
- [22] M. Abdalla, M. Bellare, and P. Rogaway, "DHAES: An encryption scheme based on the Diffie-Hellman problem.," *IACR Cryptol. ePrint Arch.*, vol. 1999, p. 7, 1999.
- [23] L. Buttyán, T. Holczér, and I. Vajda, "On the effectiveness of changing pseudonyms to provide location privacy in VANETs," in *Security and Privacy in Ad-hoc and Sensor Networks* (F. Stajano, C. Meadows, S. Capkun, and T. Moore, eds.), (Berlin, Heidelberg), pp. 129–141, Springer Berlin Heidelberg, 2007.
- [24] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," in *Advances in Cryptology – CRYPTO 2004* (M. Franklin, ed.), (Berlin, Heidelberg), pp. 41–55, Springer Berlin Heidelberg, 2004.
- [25] M. A. R. Bae, L. Simpson, X. Boyen, E. Foo, and J. Pieprzyk, "On the efficiency of pairing-based authentication for connected vehicles: Time is not on our side!," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 3678–3693, 2021.
- [26] A. Perrig, R. Canetti, J. D. Tygar, and D. Song, "The TESLA broadcast authentication protocol," *RSA Cryptobytes*, vol. 5, no. 2, pp. 2–13, 2002.
- [27] R. A. Uzcategui, A. J. D. Sucre, and G. Acosta-Marum, "Wave: A tutorial," *IEEE Communications Magazine*, vol. 47, pp. 126–133, May 2009.
- [28] M. Just, *Diffie-Hellman Key Agreement*, pp. 154–154. Boston, MA: Springer US, 2005.
- [29] M. Dworkin, "Recommendation for block cipher modes of operation. methods and techniques," tech. rep., National Inst of Standards and Technology Gaithersburg MD Computer security Div, 2001.
- [30] P. J. Bond, "The keyed-Hash Message Authentication Code (HMAC)," *Federal Information Processing Standards Publication, FIPS PUB*, vol. 198, pp. 1–21, 2002.
- [31] Q. Dang, "Changes in Federal Information Processing Standard (FIPS) 180-4, secure hash standard," *Cryptologia*, vol. 37, no. 1, pp. 69–73, 2013.
- [32] H. Krawczyk, "Cryptographic extraction and key derivation: The HKDF scheme," in *Advances in Cryptology – CRYPTO 2010* (T. Rabin, ed.), (Berlin, Heidelberg), pp. 631–648, Springer Berlin Heidelberg, 2010.
- [33] M. Bellare, "New proofs for NMAC and HMAC: security without collision resistance," *Journal of Cryptology*, vol. 28, no. 4, pp. 844–878, 2015.
- [34] B. Kaliski, "PKCS #5: Password-Based Cryptography Specification Version 2.0." RFC 2898, Sept. 2000.
- [35] D. Johnson, A. Menezes, and S. Vanstone, "The Elliptic Curve Digital Signature Algorithm (ECDSA)," *International journal of information security*, vol. 1, no. 1, pp. 36–63, 2001.
- [36] FIPS, "Federal Information Processing Standard (FIPS) 186-4," *National Institute of Science and Technology*, July 2013.
- [37] M. Campagna, "Sec 4: Elliptic curve qu-vanstone implicit certificate scheme (ECQV)," *Standards for Efficient Cryptography, Version*, vol. 1, 2013.
- [38] D. R. L. Brown, "Generic groups, collision resistance, and ECDSA," *Designs, Codes and Cryptography*, vol. 35, no. 1, pp. 119–152, 2005.
- [39] L. C. Washington, *Elliptic curves: number theory and cryptography*. Chapman and Hall/CRC, 2008.
- [40] D. K. Gillmor, "Negotiated Finite Field Diffie-Hellman Ephemeral Parameters for Transport Layer Security (TLS)." RFC 7919, Aug. 2016.
- [41] A. M. Turing, "On computable numbers, with an application to the entscheidungsproblem," *Proceedings of the London mathematical society*, vol. 2, no. 1, pp. 230–265, 1937.
- [42] O. Goldreich, *Foundations of cryptography: volume 1, basic tools*. Cambridge university press, 2007.
- [43] Y.-C. Hu and K. P. Laberteaux, "Strong VANET security on a budget," in *Proceedings of Workshop on Embedded Security in Cars (ESCAR)*, vol. 6, pp. 1–9, Citeseer, 2006.
- [44] P. Gutmann, "Encrypt-then-MAC for Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)." RFC 7366, Sept. 2014.
- [45] The OpenSSL Project, "OpenSSL: The open source toolkit for SSL/TLS." www.openssl.org, April 2003.
- [46] IEEE, "IEEE standard for wireless access in vehicular environments—security services for applications and management messages - amendment 1," *IEEE Std 1609.2a-2017 (Amendment to IEEE Std 1609.2-2016)*, pp. 1–123, Oct 2017.
- [47] M. A. R. Bae, "Implementation and performance analysis of identity-based authentication in wireless sensor networks," Master's thesis, Universiti Teknologi Malaysia, 2014.
- [48] B. Palaniswamy, S. Camtepe, E. Foo, L. Simpson, M. A. Rezazadeh Bae, and J. Pieprzyk, "Continuous authentication for VANET," *Vehicular Communications*, vol. 25, p. 100255, 2020.
- [49] F. Vercauteren, "Optimal pairings," *IEEE Transactions on Information Theory*, vol. 56, no. 1, pp. 455–461, 2009.
- [50] P. S. L. M. Barreto and M. Naehrig, "Pairing-friendly elliptic curves of prime order," in *Selected Areas in Cryptography* (B. Preneel and S. Tavares, eds.), (Berlin, Heidelberg), pp. 319–331, Springer Berlin Heidelberg, 2006.
- [51] Y. Sakemi, T. Kobayashi, and T. Saito, "Pairing-Friendly Curves," Internet Engineering Task Force, Nov. 2019. Work in Progress.
- [52] D. F. Aranha and C. P. L. Gouvêa, "RELIC is an Efficient Library for Cryptography." <https://github.com/relic-toolkit/relic>.
- [53] C. Sommer, Z. Yao, R. German, and F. Dressler, "Simulating the influence of IVC on road traffic using bidirectionally coupled simulators," in *IEEE INFOCOM Workshops 2008*, pp. 1–6, April 2008.
- [54] A. Varga, "The OMNeT++ discrete event simulation system," in *In ESM01*, 2001.
- [55] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO - Simulation of Urban Mobility," *International Journal On Advances in Systems and Measurements*, vol. 5, pp. 128–138, December 2012.
- [56] D. Eckhoff and C. Sommer, "A Multi-Channel IEEE 1609.4 and 802.11p EDCA Model for the Veins Framework," in *5th ACM/ICST International Conference on Simulation Tools and Techniques for Communications*, (Desenzano, Italy), ACM, March 2012.
- [57] IEEE, "IEEE standard for information technology—telecommunications and information exchange between systems local and metropolitan area networks—specific requirements - part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications," *IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012)*, pp. 1–3534, Dec 2016.