Securing the Lifeline: Enforcing Read and Write Access Control in Vehicular Safety Services

Muhammad Baqer Mollah, George Zouridakis, and Kyu In Lee
Department of Information Science Technology, University of Houston, Houston, TX 77204 USA
Email: {mmollah, gzourida, klee49}@uh.edu

2026 IEEE International Conference on Cyber Security and Resilience (CSR) | 979-8-3195-0774-7/26/$31.00 ©2026 IEEE | DOI: 10.1109/CSR69374.2026.11759872

***Abstract*—Vehicular safety services increasingly rely on timely dissemination of safety messages among vehicles and roadside infrastructures to prevent accidents and enhance road safety. However, existing security solutions mainly emphasize on authentication and integrity, while offering limited support for fine-grained access control over who is authorized to transmit as well as access to the safety-critical information. This particular limitation exposes vehicular networks to risks, such as unauthorized or malicious false message injection and unauthorized access to sensitive vehicular data. At the same time, traditional public-key security mechanisms are unable to flexibly enforce vehicle role, trust level, or proximity like contextual access policies. Hence, with the evolution of the Internet of Vehicles (IoV), ensuring the security of vehicular safety services has become important. In this paper, we present a security mechanism primarily based on access control encryption that can enforce both read (decrypt) and write (encrypt) access control over the disseminations of safety messages. Performance evaluation results show that the proposed work introduces reasonable computational overhead that makes it practical for vehicular safety applications.**

***Index Terms*—Internet of vehicles, safety messages, hazard notifications, cooperative awareness, access control, security.**

I. Introduction

The Internet of Vehicles (IoV) represents an evolution of intelligent transportation systems by making modern vehicles possible to communicate with other vehicles, roadside infrastructures, and Internet through Vehicle-to-Everything (V2X) technologies [1]. By integrating wireless communication, edge computing, and vehicle's on-board sensing, IoV supports a wide range of applications, from traffic efficiency and infotainment to, most importantly, vehicular safety services. These applications are focused on safety mainly depend on road hazard alerts, exchanging collision warnings, and cooperative awareness messages among other vehicles and roadside infrastructures to support decision-making, reduce the likelihood of collisions, and improve the situational awareness. Since IoV deployments are continuing to expand across both cities and highways, vehicular safety services will play an important role in future safe and efficient transportation systems.

Vehicular safety services are primarily realized by dissemination of safety messages, which can be classified according to their nature as basic safety messages, cooperative awareness messages, and event-triggered hazard notifications [2]. Generally, the information carried by these messages allows vehicles to handle safety-critical situations and coordinate maneuvers, and subsequently respond to dynamically changing traffic situations. However, the open and broadcast nature of wireless vehicular communication essentially introduces a wide attack surfaces that make safety messages vulnerable to adversarial attacks [3], [4]. For instance, malicious entities might trigger unnecessary braking or traffic disruptions by injecting falsified safety alerts, mislead nearby vehicles by replaying outdated messages, or even infer sensitive location information of the sender vehicle by eavesdropping on communications, and such malicious activities could possibly lead to serious risks to both human lives as well as transportation infrastructures.

A. Related Works

Given the safety-critical nature of vehicular communications, it is essential to ensure security of safety messages. For that, several prior works have investigated broadcast encryption (BE) [5], [6] and public key infrastructure (PKI) [7], [8] to secure safety communications. In particular, BE approaches allow a sender to encrypt messages for multiple receivers efficiently. However, such approaches work well for stable groups. Hence, frequent vehicle joins and leaves might require substantial re-keying overhead that essentially imposes unnecessary communication cost and latency. On the other hand, PKI enabled approaches primarily ensure message authentication and integrity through digital certificates. Nevertheless, PKI systems depend heavily on certificate management, revocation lists, and centralized trust authorities, which can introduce scalability challenges, communication overhead, and delayed revocation enforcement in dense vehicular scenarios. And, these approaches provide limited support for access control and lack flexibility on enforcing contextual access policies, such as locations, proximity and roles of the vehicles, or trust levels.

Furthermore, attribute-based encryption (ABE) [9], [10], [11] based schemes have been introduced as a more specific cryptographic mechanism for enforcing policy-assisted access control in vehicular networks by binding decryption capabilities to user attributes. While ABE provides improved flexibility compared to BE and PKI based works, it suffers one common practical limitation, that is, the ABE schemes inherently focus on controlling receiver-side access and lacks control over who can write (encrypt) which leaves the system vulnerable to malicious senders who can still produce valid-looking safety messages. Hence, ABE based schemes are unable to prevent unauthorized message injection, where authenticated but malicious vehicles disseminate misleading safety alerts.

Authorized licensed use limited to: University of Houston. Downloaded on November 10,2026 at 19:29:53 UTC from IEEE Xplore. Restrictions apply.

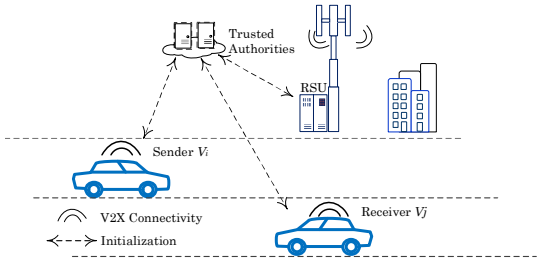


Fig. 1: The considered system model of this proposed work.

Motivated by the aforementioned limitations, in this paper, we propose an Access Control Encryption (ACE) based approach to secure vehicular safety services by enforcing sender and receiver authorization at the cryptographic level. Our approach ensures that only policy-compliant vehicles can generate or decrypt safety messages, even in broadcast or multi-receiver communication scenarios.

II. System Description and Preliminaries

In this section, we present the detailed system description, followed by the preliminaries.

A. System Model

As presented the system model in Fig. 1, the proposed vehicular safety services security solution involves three main entities, such as vehicles, roadside units, and trusted authorities. The roles of each entities are described as follows.

Vehicles, identified by V, are the vehicles that are assumed to be either as a sender V_i or receiver V_j. Each vehicle is equipped with an in-vehicle network, which integrates multiple Electronic Control Units (ECUs) linked through a Controller Area Network (CAN) bus, while the **On-Board Unit (OBU)** routes messages (e.g., sensor readings, operational status, and control commands) to/from ECUs as well as manages the communications between internal components and external network.

Roadside Units (RSU) are installed alongside the roads in order to serve the vehicles with their computing and communication capabilities. For this work, the primary task of each RSU involves receiving encrypted ciphertexts from nearby vehicles, sanitize it, store, disseminate to the network by either push or pull way.

Trusted Authorities (TA) are responsible for initializing the whole system by generating public and private parameters along with cryptographic keys and sending necessary keys to the corresponding entities. Besides, they issue a unique identity for every vehicle associated with its own vehicle-identification-number so that the identity of any specific user can be disclosed when required.

B. Adversary Model

As outlined in the system model, we consider both RSU and OBU as honest-but-curious entities, which means they are assumed to execute all required processes correctly, yet they might still attempt to analyze the received messages to infer sensitive information. Additionally, we consider an adversary, denoted as $\mathcal{A}$, having the following capabilities in line with practical scenarios.

The adversary $\mathcal{A}$ can be a passive adversary who might observe traffic over the public communication channels among vehicles, RSU and TA. By eavesdropping on exchanged messages, it might attempt to extract useful information and potentially exploit these observations to compromise or manipulate the in-vehicle network components with future communications through malicious activities, for example, pretending to originate from legitimate source.

The adversary $\mathcal{A}$ can also be an active adversary who might seek to disrupt the normal operation of in-vehicle network by injecting malicious or misleading messages into the network. Examples include forging messages or delaying essential messages. Besides, vehicles with higher levels of automation with minimal/no human intervention, it might be challenging to maintain safe vehicle operation during any unexpected events. Hence, such adversarial activities have a direct impact on the system failures, infrastructure damage, or even threaten human safety.

C. Security Objectives

Based on the aforementioned adversary model, we aim to achieve the following security objectives:

Anonymity: The proposed solution should maintain complete anonymity so that the real identity information of any vehicle should not be revealed to the adversary, RSUs, and OBUs through analysis of transmitted messages, request patterns, or safety message contents.

Confidentiality: The safety messages should remain confidential and inaccessible to the RSUs, OBUs, any intermediate entities, or external adversaries that are not authorized to view its contents, and the intermediate entities should be responsible for only sanitizing and storing the ciphertexts without learning anything about the underlying messages.

Access Control: The approach should ensure fine-grained read and write control over their attributes. Hence, we have to ensure both read and write access restriction in such a way that only authorized sender vehicles can write (encrypt), while only intended receiver vehicles should grant access (read).

Freshness: The vehicular network components may be vulnerable to be deceived by replaying or making force to accept any old messages, hence ensuring message freshness is necessary to verify that messages are recent upon receipt.

Availability: In case of the RSU becomes the target of any denial of service attack or single point of failure, the vehicles should have access to essential services without significant interruption.

D. Bilinear Pairings

Let $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_3$ be cyclic groups of prime order q. Also, let g_1 and g_2 denote generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. We then define a bilinear pairing as $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_3$, which serves as a key cryptographic primitive in this work and satisfies the following three properties:

Authorized licensed use limited to: University of Houston. Downloaded on November 10,2026 at 19:29:53 UTC from IEEE Xplore. Restrictions apply.

- *Bilinearity:* For all $a, b \in \mathbb{Z}_q$ and for any $g_1 \in \mathbb{G}_1$, $g_2 \in \mathbb{G}_2$, the pairing fulfills $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$.
- *Efficient Computability:* There exists an efficient algorithm having capability to compute $e(g_1, g_2)$ for any valid group elements $g_1 \in \mathbb{G}_1$, $g_2 \in \mathbb{G}_2$.
- *Non-degeneracy:* The pairing should satisfy $e(g_1, g_2) \neq 1$ in the sense that it ensures the mapping of the generators of the source groups to the generator of the target group.

E. Access Control Encryption

Access control encryption (ACE) enforces access control by a predicate function $\mathbb{P}$ that authorizes a group of senders S_i to transmit to a certain group of receivers R_j only when satisfying $\mathbb{P}(s_i, r_j) = 1$, where $s_i \in S_i$ and $r_j \in R_j$. In this work, we consider an ACE variant presented in [12] to construct our proposed secure vehicular safety services. This scheme is based on cipher-policy as well as senders' and receivers' attributes managed by two distinct authorities namely sender authority to specify who can write and receiver authority to enable who can read, respectively, hence, it is referred as Cross-Domain Attribute-Based ACE (CD-ABACE). The CD-ABACE scheme basically consists of following operations.

- $\mathsf{RASetup}(k, \mathcal{U}) \rightarrow (pp_{\mathtt{ra}}, \mathsf{mkey}_{\mathtt{ra}})$: It initializes the entire system that runs by the receiver authority. It takes the security parameter k and the global attribute universe $\mathcal{U}$ as input and then generates the public parameters $pp_{\mathtt{ra}}$ and master secret key $\mathsf{mkey}_{\mathtt{ra}}$.
- $\mathsf{SASetup}(k, pp_{\mathtt{ra}}) \rightarrow (pp_{\mathtt{sa}}, \mathsf{mkey}_{\mathtt{sa}})$: Given k and $pp_{\mathtt{ra}}$ as input, the SASetup algorithm executed by the sender authority that outputs the corresponding public parameters $pp_{\mathtt{sa}}$ and master secret key $\mathsf{mkey}_{\mathtt{sa}}$.
- $\mathsf{EKGen}(pp_{\mathtt{ra}}, pp_{\mathtt{sa}}, \mathsf{mkey}_{\mathtt{sa}}, \mathcal{A}_s, \mathbb{P}) \rightarrow (\mathsf{skey}_{\mathcal{A}_s}, \sigma, \omega)$: On input $pp_{\mathtt{ra}}$, $pp_{\mathtt{sa}}$, $\mathsf{mkey}_{\mathtt{sa}}$, the ciphertext attribute set $\mathcal{A}_s$, and the predicate function $\mathbb{P}$, the sender authority runs EKGen algorithm that issues a secret encryption key $\mathsf{skey}_{\mathcal{A}_s}$ associated with $\mathcal{A}_s$ for enforcing the write policy, a signature σ for sender authentication, and a re-randomization token ω.
- $\mathsf{DKGen}(\mathsf{mkey}_{\mathtt{ra}}, \mathcal{A}_r) \rightarrow \mathsf{skey}_{\mathcal{A}_r}$: The receiving authority runs DKGen algorithm that takes as input $\mathsf{mkey}_{\mathtt{ra}}$ and the authorized set of receiver's attributes $\mathcal{A}_r$ and generates a secret decryption key $\mathsf{skey}_{\mathcal{A}_r}$.
- $\mathsf{Encrypt}(pp_{\mathtt{ra}}, pp_{\mathtt{sa}}, msg, \mathsf{skey}_{\mathcal{A}_s}, \sigma, \omega) \rightarrow (\pi, x)$: Given the message msg and outputs of the EKGen algorithm, the sending entity performs Encrypt that returns a zero-knowledge proof π and a public instance $\mathbf{x}$ containing the ciphertext and auxiliary values.
- $\mathsf{Sanitize}(pp_{\mathtt{ra}}, pp_{\mathtt{sa}}, \pi, \mathbf{x}, \mathbb{P}) \rightarrow st/\ \perp$: This algorithm is run by the sanitizer that takes the public parameters, the proof π and its instance $\mathbf{x}$, and the predicate function $\mathbb{P}$. Then, the sanitizer verifies the proof and checks whether the the access control predicate $\mathbb{P}(\mathcal{A}_s, \mathcal{A}_r) = 1$ is satisfied or not. If satisfied, it re-randomizes the ciphertext and returns the sanitized ciphertext $\mathbf{st}$, otherwise, rejects.
- $\mathsf{Decrypt}(pp_{\mathtt{ra}}, pp_{\mathtt{sa}}, \mathbf{st}, \mathsf{skey}_{\mathcal{A}_r}) \rightarrow msg'/\ \perp$: The receiver entity executes this algorithm that takes the public parameters, the sanitized ciphertext st, and the decryption key $\mathsf{skey}_{\mathcal{A}_r}$ as input and retrieves the message msg' with its own attributes along with the decryption key $\mathsf{skey}_{\mathcal{A}_r}$, otherwise rejects.

III. Our Construction

In the following, we present the main construction of our proposed work.

A. System Setup

In this phase, for specific sender and receiver vehicles V_i and V_j having corresponding pseudo identities id_i and id_j, the receiver authority (RA) first runs RASetup that takes security parameter k as input and performs the group generator $\mathcal{G}(k)$ to obtain the tuple $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, q, e)$ as global setup, where $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3$ are the three cyclic groups of prime order q, $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$ are the generators, and e is the bilinear pairing. Then, it uniformly chooses random integer $\mathbf{r} \in \mathbb{Z}_q^*$ and a collision-resistant cryptographic hash function $H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$. Given the global attribute universe set having n attributes $\mathcal{U} = \{a_1, a_2, ..., a_n\}$, it computes $h_i = g_2^{\mathbf{r}^i} \in \mathbb{G}_2$ for each attribute index $i \in \{0, 1, 2, ..., n-1\}$ and $\mathbf{g}_2 = g_1^{\mathbf{r}^2} \in \mathbb{G}_1$. Last, it stores the master secret key $\mathsf{mkey}_{\mathtt{ra}} = (g_1, \mathbf{r})$ and disseminates the following system public parameters to all participating entities within the vehicular network.

$$pp_{\mathtt{ra}} = (\mathbf{g}_2, \{h_i\}_{i=1}^{n-1}, e(g_1, g_2)^{\mathbf{r}}, H_1). \tag{1}$$

After this, given the global bilinear group configuration $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, q, e)$ and public parameters of RA $pp_{\mathtt{ra}}$, the sender authority (SA) runs SASetup that setups its own master secret key and public parameters while ensuring cross-domain compatibility. For that, it chooses a random group element $\gamma \leftarrow \mathbb{G}_2$ and integer $\nu \leftarrow \mathbb{Z}_q$, computes $\mathbf{g}_2^{\nu} = g_1^{\mathbf{r}^2 \nu}$, and then sets the signing key as $\mathsf{sk}_{\mathtt{sa}} = \nu$ and verification key as $\mathsf{vk}_{\mathtt{sa}} = \mathbf{g}_2^{\nu}$. Further, it establishes the non-interactive zero-knowledge proof (NIZK) setup by generating a common reference string crs and a trapdoor τ given by the security parameter k and a specified relation $\mathtt{R}_{\mathtt{L}}$. Finally, the master key is given by $\mathsf{mkey}_{\mathtt{sa}} = (\mathsf{sk}_{\mathtt{sa}}, \tau)$ and the public parameters are released as

$$pp_{\mathtt{sa}} = (\gamma, \mathsf{vk}_{\mathtt{sa}}, \mathtt{R}_{\mathtt{L}}, \mathsf{crs}). \tag{2}$$

B. Entity Registration

When the sender and receiver vehicles V_i and V_j having corresponding pseudo identities id_i and id_j join to the vehicular network, they have to go through a registration phase, where the RA and SA issue necessary cryptographic components for them. Accordingly, the SA runs EKGen where it first retrieves the group settings, public parameters, and own secret key as well as defines the access policy set $\mathcal{A}_s$ and the predicate function $\mathbb{P}$. Then, it constructs the policy polynomial as

$$\mathrm{Z}_{\mathcal{A}_s}(x) = \prod_{i=1}^{n} (x - \kappa_i)^{\overline{\mathbb{P}[i]}}, \tag{3}$$

Authorized licensed use limited to: University of Houston. Downloaded on November 10,2026 at 19:29:53 UTC from IEEE Xplore. Restrictions apply.

where $\kappa_i = \{H_1(a_i)\}_{a_i \in \mathcal{U}}$ and $\mathtt{p}[i]$ represent the attribute encoding and the binary indicator of the complement set of $\mathcal{A}_s$, respectively. And, to derive the encryption key $\mathsf{skey}_{\mathcal{A}_s}$ embedded with the specific policy structure $\mathcal{A}_s$ so that only authorized senders can encrypt under this policy, it aggregate the public parameter sequence h_i using the coefficients z_i of the policy polynomial $\mathrm{Z}_{\mathcal{A}_s}$. Since the public parameters are defined as $h_i = g_2^{\mathbf{r}^i}$, this aggregation eventually simplifies to $g_2^{\mathbf{r}\mathrm{Z}_{\mathcal{A}_s}(\mathbf{r})}$. Mathematically, we can write as follows

$$\mathsf{skey}_{\mathcal{A}_s} = \prod_{i=1}^{n-1} h_{i+1}^{\mathbf{z}_i} = g_2^{\mathbf{r}\mathrm{Z}_{\mathcal{A}_s}(\mathbf{r})}. \tag{4}$$

Besides the encryption key $\mathsf{skey}_{\mathcal{A}_s}$, the SA signs on it for sender authentication and generates a token for sanitization phase. As such, the SA picks a random value $\mathbf{t} \in \mathbb{Z}_q^*$ to compute the re-randomization token $\omega = g_2^{1/\mathbf{t}}$. Next, it uses the secret signing key $\mathsf{sk}_{\mathsf{sa}}$ to generate the signature $\sigma = (\sigma_1, \sigma_2, \sigma_3)$, where $\sigma_1 = \mathbf{g}_2^{\mathbf{t}}$, $\sigma_2 = \mathsf{skey}_{\mathcal{A}_s}^{\mathsf{sk}_{\mathsf{sa}}/\mathbf{t}}$, and $\sigma_3 = \sigma_2^{\mathsf{sk}_{\mathsf{sa}}/\mathbf{t}} g_2^{1/\mathbf{t}}$.

Next, the RA runs DKGen that generates the private decryption key $\mathsf{skey}_{\mathcal{A}_r}$ for the receiver vehicle V_j given the attribute policy $\mathcal{A}_r$. This policy is particularly mapped to a receiver polynomial by calculating their corresponding attribute encoding κ_i and the binary indicator of the complement of the policy $\mathtt{b}[i]$ as

$$\mathrm{Z}_{\mathcal{A}_r}(x) = \prod_{i=1}^{n} (x - \kappa_i)^{\overline{\mathtt{b}[i]}} \tag{5}$$

At the end, the key is given by $\mathsf{skey}_{\mathcal{A}_r} = g_1^{1/\mathrm{Z}_{\mathcal{A}_r}(\mathsf{r})}$.

C. Safety Message Preparation

Given the safety message msg, the sender vehicle V_i first randomly chooses a symmetric secret key $\mathsf{skey}_{\mathsf{sym}} \leftarrow \{0,1\}^k$, where k is the security parameter. This key $\mathsf{skey}_{\mathsf{sym}}$ is used to encrypt the message msg that returns the ciphertext $\mathbf{ct}_{\mathsf{msg}}$. Next, it also generates and appends a current timestamp T_i to ensure freshness and prevent replay attacks. Then, it runs $\mathsf{Encrypt}$ algorithm that randomly chooses $\mathbf{a}, \mathbf{b} \in \mathbb{Z}_q^*$, and computes the ciphertext as tuple $\mathbf{ct} = (\mathbf{ct}_1, \mathbf{ct}_2, \mathbf{ct}_3)$, where

$$\begin{aligned} \mathbf{ct}_1 &= \mathsf{skey}_{\mathsf{sym}} \cdot e(g_1, g_2)^{\mathbf{ar}}, \\ \mathbf{ct}_2 &= \mathsf{ekey}^{\mathbf{a}}, \\ \mathbf{ct}_3 &= \mathbf{g}_2^{-\mathbf{a}}. \end{aligned} \tag{6}$$

With the signature re-randomization token ω, the V_i after that computes to enable sender anonymity $\sigma_1' = \sigma_1^{1/\mathbf{b}}$, $\sigma_2' = \sigma_1^{\mathbf{b}}$, and $\sigma_3' = \sigma_3^{\mathbf{b}^2} \omega^{\mathbf{b}(1-\mathbf{b})}$, which form a re-randomized signature as $\sigma' = (\sigma_1', \sigma_2', \sigma_3')$.

Afterwards, the V_i defines the public instance as $\mathbf{x} = (\sigma', \mathsf{vk}_{\mathsf{sa}}', \mathsf{ekey}', \mathbf{ct})$, where $\mathsf{vk}_{\mathsf{sa}}' = \mathsf{vk}_{\mathsf{sa}}^{1/\mathbf{b}}$ and $\mathsf{ekey}' = \mathsf{ekey}^{\mathbf{b}}$ and the corresponding witness as $\mathbf{w} = (\mathsf{ekey}, \sigma, msg, \mathbf{a}, \mathbf{b})$. Finally, the NIZK proof is generated as $\pi = \mathsf{ZKPoK}\{\mathbf{w} : \mathsf{R}_{\mathrm{L}}(\mathbf{x}, \mathbf{w}) = 1\}$, where $\mathsf{ZKPoK}(\cdot)$ is the proof of knowledge of the hidden witness $\mathbf{w}$ that satisfies a given relation R_{L}. Here, $(\pi, \mathbf{x})$ enables to prove without requiring to know any secret that it possesses the randomness, a valid encryption key ekey, the key is correctly signed by the authorized sender V_i, and the ciphertext $\mathbf{ct}$ is generated using this key. Subsequently, it submits $(id_i, \mathbf{ct}_{\mathsf{msg}}, \pi, \mathbf{x}, T_i)$ as part of a request to be sanitized and disseminated to the vehicular network.

Once received the sanitization request, the nearby road-side unit (RSU) proceeds to verify the signature σ', proof π, and system's predicate function $\mathbb{P}(\mathcal{A}_s, \mathcal{A}_r)$. If one them is not satisfied, it drops the $\mathbf{ct}$, otherwise it proceeds to sanitization step that begins by re-randomizing the ciphertext to ensure unlinkability. For that, it randomly chooses $\mathbf{c} \in \mathbb{Z}_q^*$ and then calculates the following components to get the sanitized ciphertext $\mathbf{st} = (\mathbf{st}_1, \mathbf{st}_2, \mathbf{st}_3)$.

$$\begin{aligned} \mathbf{st}_1 &= \mathbf{c_1} \cdot e(g_1, g_2)^{\mathbf{cr}}, \\ \mathbf{st}_2 &= \mathbf{ct}_2 \cdot g_2^{\mathbf{cr}\mathrm{Z}_{\mathcal{A}_s}(\mathbf{r})}, \\ \mathbf{st}_3 &= \mathbf{ct}_3 \cdot \mathbf{g}_2^{-\mathbf{c}}. \end{aligned} \tag{7}$$

After the sanitization processes, the RSU disseminates the final ciphertexts $(\mathbf{ct}_{\mathsf{msg}}, \mathbf{st})$ along with the timestamp among the nearby vehicles and other RSUs.

D. Safety Message Recovering

Upon receiving the ciphertexts $(\mathbf{ct}_{\mathsf{msg}}, \mathbf{st})$ along with the timestamp T_i from the RSU, the OBU of receiver vehicle V_j first verifies the validity of the timestamp for freshness by checking the condition $|T_{\text{current}} - T_i| \leq \Delta T$ whether it falls within a predefined time window ΔT. If the condition is not satisfied, the received content is considered outdated or compromised and consequently is discarded to prevent replay attacks. Otherwise, the OBU proceeds to recover the symmetric secret key $\mathsf{skey}_{\mathsf{sym}}'$ using its own attributes and secret decryption key dkey by running $\mathsf{Decrypt}$ algorithm. In particular, if the receiver's attribute set $\mathcal{A}_r$ satisfies the ciphertext policy $\mathcal{A}_s$, that is, the policy attributes are a subset of the receiver's attributes $\mathcal{A}_s \subseteq \mathcal{A}_r$, it reconstructs define $\mathtt{c}[i] = \mathtt{b}[i] - \mathtt{p}[i]$ and construct the polynomial as $\mathrm{Q}(x) = \prod_{i=1}^{n} (x - \kappa_i)^{\mathtt{c}[i]}$. In case the OBU lacks direct connection with the RA and to facilitate offline operations, it might be necessary to pre-load the the secret key dkey on OBU. Next, the key $\mathsf{skey}_{\mathsf{sym}}'$ is recovered by using the following pairing operations:

$$\mathsf{skey}_{\mathsf{sym}}' = \frac{\mathbf{st}_1}{\left[e(\mathbf{st}_3, \prod_{i=1}(h_{i-1})^{\mathbf{q}_i}) \cdot e(\mathsf{dkey}, \mathbf{st}_2)\right]^{1/\mathbf{q}_0}}. \tag{8}$$

Finally, the desired message msg' can be recovered by utilizing $\mathsf{skey}_{\mathsf{sym}}'$.

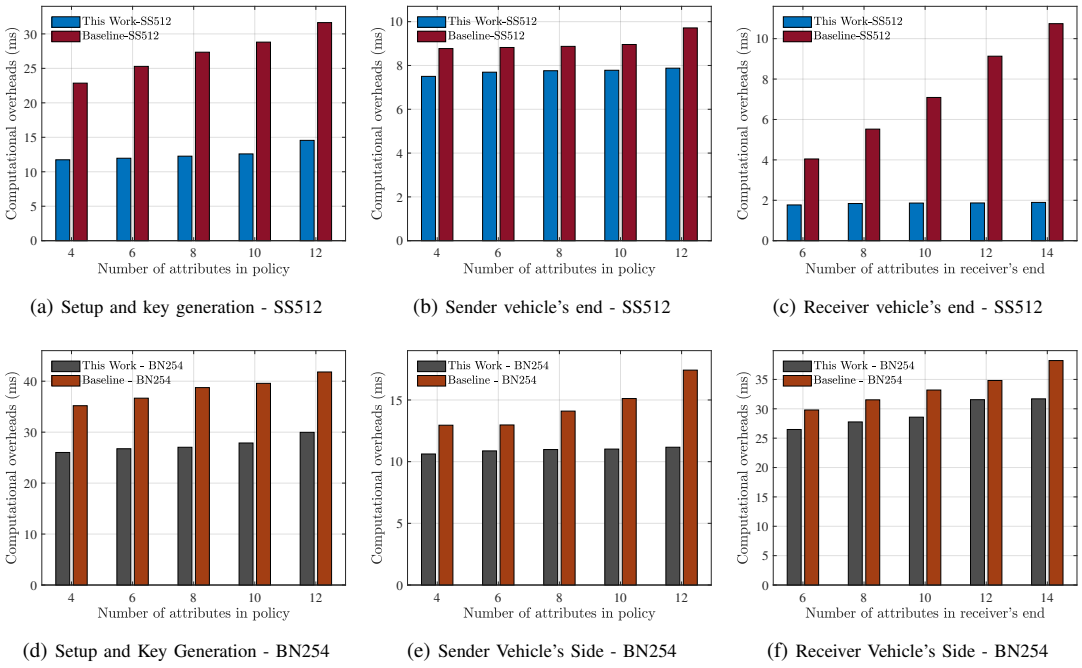
IV. Performance and Security Evaluation

In this section, we evaluate the performance of the proposed work and then present the security analysis as follows.

A. Experiments and Results

In practice, the vehicles' on-board units (OBUs) and roadside units (RSUs) typically possess significantly higher computational capabilities than the in-vehicle components, such electronic control units. Accordingly, we perform the

Authorized licensed use limited to: University of Houston. Downloaded on November 10,2026 at 19:29:53 UTC from IEEE Xplore. Restrictions apply.



(a) Setup and key generation - SS512 (b) Sender vehicle's end - SS512 (c) Receiver vehicle's end - SS512

(d) Setup and Key Generation - BN254 (e) Sender Vehicle's Side - BN254 (f) Receiver Vehicle's Side - BN254

Fig. 2: The comparative performance results in terms of computational overheads under SS512 and BN254 pairings.

experiment on a laptop computer, that is, Lenovo IdeaPad Slim 5i, where we confiture multiple virtual machines by Oracle VirtualBox[1] to run Ubuntu 18.04 LTS operating system with a single CPU core and 4GB RAM. To be specific, it hosts three virtual machines, namely VM1, VM2, and VM3 to implement the sender vehicle V_i's OBU (OBU1), the RSU and the receiver vehicle V_j's OBU (OBU2), respectively. And, for implementing the pairing-based cryptographic operations, we incorporate the Charm cryptographic library[2] in Python 3.6. Specifically, we utilize AES-CBC mode for symmetric cryptographic operations.

Further, we choose the cross-domain access control encryption scheme presented in [13] as our baseline. Except using identity-based broadcast encryption, both our adopted and their scheme follow common construction principles for the same purposes, that is, enforcing fine-grained read and write permissions while ensuring confidentiality and participant's privacy. Accordingly, these common aspects make the considered baseline a fair reference for performance evaluation. Afterwards, to make a realistic performance evaluation, we consider the message payloads, an executable shell script (.sh), of roughly 100 bytes. Finally, for modeling the write and read attributes to use this experiment in accordance with a practical scenario, we choose a lane closure scenario so that the recipient vehicle can exclusively decrypt/receive a specific instruction to execute in advance to change the lane and reduce speed, where the write policies are defined by vehicle roles (e.g., emergency), zone attributes (e.g., work-zone-C), and high trust-levels (e.g., ≥ 8.0), while the read policies are selected by direction (e.g., north), lane (e.g., right-lane), and distance (≤ 2 miles).

Fig. 2 illustrates a comparative performance evaluation under two pairing configurations (i.e., SS512 and BN254) across three major computational operations, such as setups and key generations, safety message preparation, and message recovering. Here, the computational overheads are measured as a basic of the number of attributes embedded in the access policy, and each experimental operation is executed over five times to prepare the average results. And, we vary the number of attributes from 4 to 12 and 6 to 14 at sender side and receiver side, respectively in order to analyze scalability, and the results shows the impact of attribute set size on computational overheads. On the other hand, in terms of sanitization overhead, we observe that it remains nearly constant with only slight fluctuations as the number of attributes increases. This is because the sanitization operation involves verification of the re-randomized signature, the proof, and the system predicate function, followed by the ciphertext re-randomization. in particular, our proposed construction incurs a sanitization overhead of 11.25 ms under SS512 and 81.85 ms under BN254 settings on average. In comparison, the baseline work achieves 7.30 ms and 11.30 ms, respectively on average. However, ours additional overheads might arise due to proof verification and ciphertext re-randomization.

B. Security Evaluation

Following the adversary model and security objectives discussed at Section II, we now present how our proposed work can fulfill each of those objectives.

Anonymity: A passive adversary $\mathcal{A}$ may attempt to infer sensitive information (i.e., sender's vehicle identity). To achieve

[1]https://www.virtualbox.org/

[2]https://github.com/JHUISI/charm

Authorized licensed use limited to: University of Houston. Downloaded on November 10,2026 at 19:29:53 UTC from IEEE Xplore. Restrictions apply.

this, the adversary $\mathcal{A}$ may closely observe access requests from/to sending vehicle CV_i, roadside units, or recipient vehicle CV_j. However, in this work, the sending and recipient vehicles operate under pseudonymous identities instead of Vehicle Identification Number (VIN) like real identities. And, we do not use any sensitive attribute values embedded within the access policies. As a result, our proposed work effectively preserves the privacy of the involved vehicles.

Confidentiality: The proposed work ensures the confidentiality of respective contents of safety messages while transferring or storage. In particular, the confidentiality is obtained by associated encryption/decryption schemes. If any authorized receiver vehicles would like to recover the message, it first has to extract the symmetric key $\mathsf{skey}_{\mathsf{sym}}$. As indicated in Eq. (8), successful key recovery requires reconstructing the bilinear pairing term. And, this reconstruction mainly depends on the secret key dkey and associated receiver's given attribute set $\mathcal{A}_r$. Hence, we claim that this work can ensure confidentiality from any malicious passive adversary $\mathcal{A}$.

Access Control: It is straightforward that our work can inherently enforce both read (decrypt) and write (encrypt) restrictions by the underlying construction of access control encryption, where both sender and recipients (V_i, V_j) are required to specify the access policies in the form of attribute sets $\mathcal{A}_s$ and $\mathcal{A}_r$, respectively. In particular, the RSU disseminates the safety messages once the ciphertext $\mathbf{ct}$ meets the system's predicate function $\mathbb{P}(\mathcal{A}_s, \mathcal{A}_r)$, which confirms that our proposed work can effectively resist defend any unauthorized access to the safety messages as well as malicious message injection by any unauthorized sender.

Freshness: This proposed work is capable of preventing replay attacks or forcefully accepting outdated message a timestamp-based freshness mechanism, where each message transmitted by a vehicle includes a timestamp T_i. A message is accepted only if the time difference between the received timestamp and the current system time is within a predefined threshold ΔT. This ensures to defend by any passive adversary $\mathcal{A}$ attempting to retransmit outdated messages.

Availability: Ensuring availability is necessary for vehicular safety services. This proposed work mitigates availability related threats, such as Denial-of-Service (DoS), Distributed DoS (DDoS), or Single Point of Failure (SPoF) attacks by leveraging road-side units. Since the safety messages are replicated across multiple RSUs, the essential service continuity for the participants can be ensured in case any individual node will be compromised. In this way, this work avoids centralized bottlenecks and becomes resilient to availability type attacks.

V. Conclusion

In this paper, we have presented a security solution for vehicular safety services. By defining a realistic adversary model, we captured practical security threats, and then constructed the solutions with an access control encryption scheme to enable fine-grained read (decrypt) and write (encrypt) access control. Additionally, we presented a detailed security analysis to show how this proposed work could be resilient against cyber attacks following the adversary models. At the end, we performed a thorough experimental evaluation to validate the effectiveness, which indicated that the proposed solution could maintain practical performance in terms of computational overheads. For the future work, we will consider integration of blockchain technology to support decentralized management, auditability, and accountability.

Acknowledgments

This work was supported by the US Department of Transportation (USDOT) Tier 1 University Transportation Center (UTC) Transportation Cybersecurity Center for Advanced Research and Education (CYBER-CARE) (Grant No. 69A3552348332); and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2024-00463802).

References

[1] M. Ali, A. Nauman, M. A. Jamshed, S. M. Kim, and J. Kim, "Vehicles-to-everything standardization, services and enhancements for intelligent transportation systems," *IEEE Communications Standards Magazine*, vol. 9, no. 4, pp. 127–134, 2025.

[2] G. T. Su and R. R. Rajkumar, "Enhanced safety messages (ESM): A practical alternative to V2X basic safety messages," in *2025 IEEE 101st Vehicular Technology Conference (VTC2025-Spring)*. IEEE, 2025, pp. 1–7.

[3] E. Khezri, H. Hassanzadeh, R. O. Yahya, and M. Mir, "Security challenges in internet of vehicles (IoV) for ITS: A survey," *Tsinghua Science and Technology*, vol. 30, no. 4, pp. 1700–1723, 2025.

[4] M. B. Mollah, H. Wang, and H. Fang, "Evaluating vulnerabilities of connected vehicles under cyber attacks by attack-defense tree," in *International Conference on Computing, Networking and Communication (ICNC)*. IEEE, 2026, pp. 1–6.

[5] J. Li, F. Zhou, Q. Xie, N. Mu, and Y. Liu, "Efficient conditional privacy-preserving heterogeneous broadcast signcryption for collision warning in VANETs," *IEEE Transactions on Network and Service Management*, vol. 23, pp. 224–238, 2026.

[6] B. Li, L. Deng, H. Liu, Y. Mou, Y. Qu, and N. Wang, "A CCA-secure pairing-free broadcast proxy re-encryption scheme for VANETs," *IEEE Transactions on Vehicular Technology*, vol. 74, no. 11, pp. 17 999–18 011, 2025.

[7] S. Khan, M. Khan, M. A. Khan, F. Luo, M. Usman, A. Irshad, K. Wu, L. Wang, and M. A. Khan, "PBatch: Pseudonym certificate batch authentication with generative AI-based cache for cooperative intelligent transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, 2025.

[8] N. Asavisanu, T. Khezresmaeilzadeh, R. Sequeira, H. Qiu, F. Ahmad, K. Psounis, and R. Govindan, "CATS: A framework for cooperative autonomy trust & security," *IEEE Transactions on Vehicular Technology*, vol. 74, no. 7, pp. 10 092–10 108, 2025.

[9] W. Wei, S. Qi, X. Yang, W. Zhao, J. Gu, K. Saleem, and Y. Qi, "AACS: A secure attribute based data access control for cloud-aided internet of vehicles using SGX," *IEEE Transactions on Vehicular Technology*, vol. 74, no. 2, pp. 2032–2047, 2025.

[10] J. Cui, J. Zhang, L. Wei, M. Zhu, H. Zhong, and G. Min, "Multi-authority ciphertext-policy attribute-based encryption with hidden policy for securing internet-of-vehicles," in *2024 IEEE 23rd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. IEEE, 2024, pp. 213–221.

[11] C. Wang, Y. Ming, H. Liu, Y. Deng, M. Yang, and J. Feng, "Puncturable registered ABE for vehicular social networks: Enhancing security and practicality," *IEEE Transactions on Dependable and Secure Computing*, vol. 22, no. 6, pp. 5998–6011, 2025.

[12] M. Sedaghat and B. Preneel, "Cross-domain attribute-based access control encryption," in *International Conference on Cryptology and Network Security*. Springer, 2021, pp. 3–23.

[13] X. Wang and S. S. Chow, "Cross-domain access control encryption: Arbitrary-policy, constant-size, efficient," in *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2021, pp. 748–761.

Authorized licensed use limited to: University of Houston. Downloaded on November 10,2026 at 19:29:53 UTC from IEEE Xplore. Restrictions apply.