Towards a TEE-based V2V Protocol for Connected and Autonomous Vehicles

Mohit Kumar Jangid
The Ohio State University
jangid.6@osu.edu

Zhiqiang Lin
The Ohio State University
zlin@cse.ohio-state.edu

***Abstract*—Being safer, cleaner, and more efficient, connected and autonomous vehicles (CAVs) are expected to be the dominant vehicles of future transportation systems. However, there are enormous security and privacy challenges while also considering the efficiency and and scalability. One key challenge is how to efficiently authenticate a vehicle in the ad-hoc CAV network and ensure its tamper-resistance, accountability, and non-repudiation. In this paper, we present the design and implementation of Vehicle-to-Vehicle (V2V) protocol by leveraging trusted execution environment (TEE), and show how this TEE-based protocol achieves the objective of authentication, privacy, accountability and revocation as well as the scalability and efficiency. We hope that our TEE-based V2V protocol can inspire further research into CAV security and privacy, particularly how to leverage TEE to solve some of the hard problems and make CAV closer to practice.**

I. Introduction

Connected and Autonomous Vehicles (CAVs) are expected to be the dominant vehicles of future transportation systems. This is due to their benefits including reduced driver stress, increased productivity, increased safety, and reduced energy consumption and pollution [1]. For instance, CAVs in a collaborative network can improve traffic safety and efficiency by exchanging congestion information, accident or emergency situations, and weather conditions. Additionally, it can also reduce the cost of medical care, emergency services, jobs, insurance administration, legal costs, and property damage to benefit the economy [5]. A recent report [29] even estimated that by 2045 nearly half of new vehicle sales could be CAVs.

However, there are enormous challenges to make vehicles connected and autonomous. First, fast and scalable communication is essential for the CAV network due to the high speed of vehicles and changing dynamics of traffic, which demands the cross-vehicle communications to be real-time. Second, securing confidentiality, authentication, and privacy of the CAV communication is also crucial. The involvement of numerous sensors of a vehicle [34], heterogeneous mobile and non-mobile nodes (e.g., the road-side units) in the CAV network exposes a large attack surface for attackers. For example, by compromising the CAV communication network, an attacker can potentially fingerprint or trace a vehicle's trajectory. Moreover, the attacker who can broadcast false information can cause fatal and devastating damages. Third, any misbehavior in the CAV communication should be prohibited and investigated. Therefore, the data management of the CAV network should include dynamic revocation and accountability of (malicious or compromised) vehicles.

Workshop on Automotive and Autonomous Vehicle Security (AutoSec) 2022
24 April 2022, San Diego, CA, USA
ISBN 1-891562-75-4
https://dx.doi.org/10.14722/autosec.2022.23044
www.ndss-symposium.org

To address these challenges, many prior efforts (e.g., [3], [12], [28]) advocate the use of certificate-based authentication to meet the demands of CAV infrastructure security such as authentication as well as scalability and efficiency, inspired by the practice from Internet-based network [19], [21]. Unfortunately, such a certificate-based authentication requires frequent asymmetric key encryption and decryption and communication with centralized public-key infrastructures, thereby hindering their practicality, particularly when considering the fact that multiple on-road vehicles can form an ad-hoc network at arbitrary moment and there is a need to instantly authenticate a vehicle and trust its communication.

In this paper, we propose a new Vehicle-to-Vehicle (V2V) protocol by leveraging the Trusted Execution Environment (TEE) of in-vehicle processors. In particular, we notice many of the security and efficiency demands can be met without involving certificates and frequent asymmetric key encryption and decryption if we can leverage TEE and use a TEE protected temporal (e.g., daily) symmetric key-based communication protocol. Specifically, in such a protocol, many security demands such as confidentiality, authenticity and replay protection can be achieved naturally by using securely provisioned Daily Symmetric (DK) keys protected by TEE. The changing symmetric keys are further stored in the TEE protected sealed storage to prevent a malicious OS or other privileged software from stealing or modifying the sensitive information. Such a TEE based authentication removes the pairwise key exchange overhead when using certificate-based authentication, allowing instantly broadcast of encrypted data.

To strike the balance between privacy and utility, we also choose to associate a temporary random vehicle identifier (TID) with all the vehicle activities to protect driver's (or user's) privacy. In particular, the TID is derived from a Vehicle-specific Root Key (VRK) stored in a sealed storage as well as the servers of trusted authorities such as Bureau of Motor Vehicles (BMV). All CAV network activities, containing TID, are logged by road side units (e.g., the edge servers, and smart traffic lights). Only when there is an accident or security breach, the VRK will be revealed to forensics investigators. Based on the VRK, the forensics investigators can further locate the vehicle owners and hold them accountable if they commit any crimes. Strict access control will be enforced for the database server to ensure only those who have the permission is allowed to review the VRK information. Additionally, our design incorporates an end-to-end life cycle of a vehicle, participating in the CAV network, starting from the vehicle manufacturing phase, followed by vehicle registration phase (after purchased by a customer). The other phases such as DS key renewal phase and the on-road communication phase will execute on various event based conditions as demanded.

Contributions. We make the following contributions:

- We propose a new TEE-based end-to-end V2V protocol based on DS key provision. We show that our protocol meets the basic requirements of security and privacy such as tamper-resilience, non-repudiation, and anonymity, as well as scalability and efficiency.
- We have developed a preliminary prototype to characterize its performance, and demonstrated for an end to end latency our approach is about 8X faster when broadcasting a 39 bytes safety message compared with a PKI-based approach, which requires additional 91.08 milliseconds for key negotiation.
- We also provide insights for features that need to be supported in TEE processors and requirements at various phases of a vehicle lifespan, in order to make CAVs more secure and highly performant.

II. Background

A. Connected and Autonomous Vehicles

A CAV needs to communicate with other on-road vehicles (V2V) or the stationary road infrastructure (V2I), e.g., the traffic lights, edge servers, or relay access points, in order to be more performant. In such a network, the CAV entities can share traffic-safety information, emergency accidents, weather conditions, route suggestions, etc. As such, this V2X communication can potentially improve traffic congestion, reduce traffic accidents, and improve the environment. However, compared to cellular, or WLAN, or Bluetooth wireless networks, the CAV network involves rapidly moving vehicles and requires real-time communication with extremely low latency.

Since CAV will use public road and other supporting infrastructures, it is natural to involve a public trusted authority to manage the network resources. In USA, various agencies can serve this role, such as the Bureau of Motor Vehicles (BMV) in each state. In particular, BMV can keep a secure database and manage access control for its local CAV network nodes. Also, it can provide database access to law enforcement authorities for legal proceedings when needed. Additionally, BMV servers can communicate with CAV network nodes to update important official information such as revocation lists or temporary secrets.

B. Trusted Execution Environment

Trusted Execution Environment (TEE) is a hardware-aided security feature to protect the confidentiality and integrity of runtime code and data from privileged software such as a compromised OS. Intel SGX [9], [22], [31], Arm TrustZone [10], AMD SEV [25] are examples of a few popular TEE-based processor features. In a typical TEE environment, a program is divided into trusted and untrusted code. The trusted code and data are operated inside a dedicated CPU module and stored in protected memory regions. The trusted code is executed within a container known as *enclave*. The protection is rooted in a tamper-proof fused key into the hardware and on-the-fly encryption/decryption of runtime code and data using a memory encryption engine (MEE). Apart from code and data protection, some TEE processors such as Intel SGX provide a sealing mechanism to store the confidential data into the permanent storage of host OS. The sealed data is encrypted using the fused sealing key which is available only within the hardware. To protect the sealed storage replay and support counter-based continuity to the sealed data, Intel SGX also provides trusted monotonic counters [15]. These counter values are provided to the trusted code using a separate trusted hardware module. Further, some TEE processors also provide a remote attestation mechanism that allows a remote authorities or remote TEE program owner, using a non-TEE machine, to verify the integrity of the running trusted code at TEE based machine. It allows the remote authority to provision secrets to trusted code for further operations. Consequently, the trusted code can obtain dynamic secrets on-demand.

III. Overview

This section describes the threat model, assumptions, and scope of our TEE-based V2V protocol in §III-A, and then it discusses the security and privacy goals that our protocol aims to achieve in §III-B.

A. Threat Model, Assumptions, and Scope

We assume that an attacker can sniff the wireless communication traffic of all CAVs; can access, modify, and replay data to any vehicles, as a classic man-in-the-middle (MitM) attacker. Further, she can compromise a vehicle's on-board unit (OBU), denoted by vehicle Integrated Circuit (IC), and probe the IC to perform all MitM attacks in the untrusted code [16]. That is, she also possesses compromised vehicle's attacking capabilities. Fortunately, the trusted components and code, in the IC, are protected by TEE, which is inaccessible to the attacker. Note that side channel attacks against TEE are out of the scope of this paper.

Further, we assume that TEE integrated circuit (IC) designers, simply denoted by TEE designer, will be able to build a lightweight processor (e.g., a RISC-V type) for vehicles where the runtime code and data of the processor, along with sensor (e.g., GPS, and timer) reading and broadcasting modules, will be protected by the TEE isolation and MEE. The current commodity processors such as Intel SGX are already able to do so except the protection of sensors read/write and trusted GPS operations. Also, we assume that the tamper-proof fused keys, by TEE designers, cannot be obtained by an attacker without permanently damaging the processors. Additionally, we assume that the database maintained by the trusted authorities such as the BMV to record CAV-related information, is secured and accessible only to authorized parties.

For simplicity and for considering a generic case for any CAV node, we will describe the design goals and design phases details considering a vehicle as the communicating unit. However, our design generically applies to any other stationary road side units (RSUs), such as traffic lights, edge servers, and relay access points, as long as the corresponding RSU has the TEE features as discussed in §IV. The manufacturing and registration phase for stationary RSUs will be similar to that of a vehicle except that the stationary units will be collectively owned by a single trusted authority. Moreover, most of the TEE features used in our paper are inspired by the Intel SGX [9], [22], [31] TEE design. Therefore, throughout the paper, we will use Intel to denote the TEE designer.

B. Security and Privacy Goals

Based upon the threat model and the scope, we next define the security and privacy goals for all participating vehicles in the CAV network. Particularly, we aim to achieve:

- **Two-way authentication**. When communicating with a specific vehicle, the source vehicle should be able to verify the authenticity of the destination vehicle, and vice-versa.
- **Confidentiality**. An attacker should not be able to obtain any sensitive information, such as encryption keys, vehicle IDs, or vehicle metadata from the sniffed packets obtained from the CAV wireless communication channel, or from the untrusted code of the vehicle processor.
- **Privacy**. For the sniffed packets from the wireless channel or from the untrusted code of the compromised vehicle IC, an attacker should not be able to bind the data with a particular vehicle. In other words, she should not be able to trace or fingerprint a vehicle based on the sniffed packets.
- **Non-repudiation**. The identity of a vehicle cannot be forged. For instance, an identified misbehaving vehicle owner should not be able to prove the existence of a different vehicle than the identified one as obtained from the recorded communication data of the misbehaving incident.
- **Accountability**. Once a misbehaving incident is reported with the corresponding communication data, the law enforcement authorities should be able to trace the misbehaving vehicle deterministically.
- **Dynamic revocation**. Once a misbehaving vehicle is identified and its existence is broadcasted to all vehicles using revoked vehicle list, any vehicle should be able to discard the communication data from the misbehaving vehicle until the misbehavior incident case is resolved.
- **Replay prevention**. All vehicles should be able to discard a replay of previously communicated data from a different time or different geography. Further, any replayed input to the vehicle processor via untrusted code should be rejected.

IV. Protocol Design

This section describes the end-to-end phases of a vehicle participating in the CAV network. At a high level, there are four phases: 1) manufacturing, 2) vehicle registration, 3) DS Key renewal, and 4) on-road communication. The first three phases are illustrated in Figure 1. In the following, we describe these four phases in greater details.

1) **Manufacturing Phase:** When a vehicle is manufactured, it needs to collaborate with CPU vendors (e.g., Intel) to install TEE-enabled processor (such as SGX). The manufacturer (or other parties) will also need to develop a software development kit (SDK) for developers to build customized enclave programs which can use the TEE features. The enclave program should be able to communicate with the trusted authorities (e.g., BMV servers), attested by them, and provisioned with vehicle specific secrets such as the vehicle root key (VRK) as discussed below.

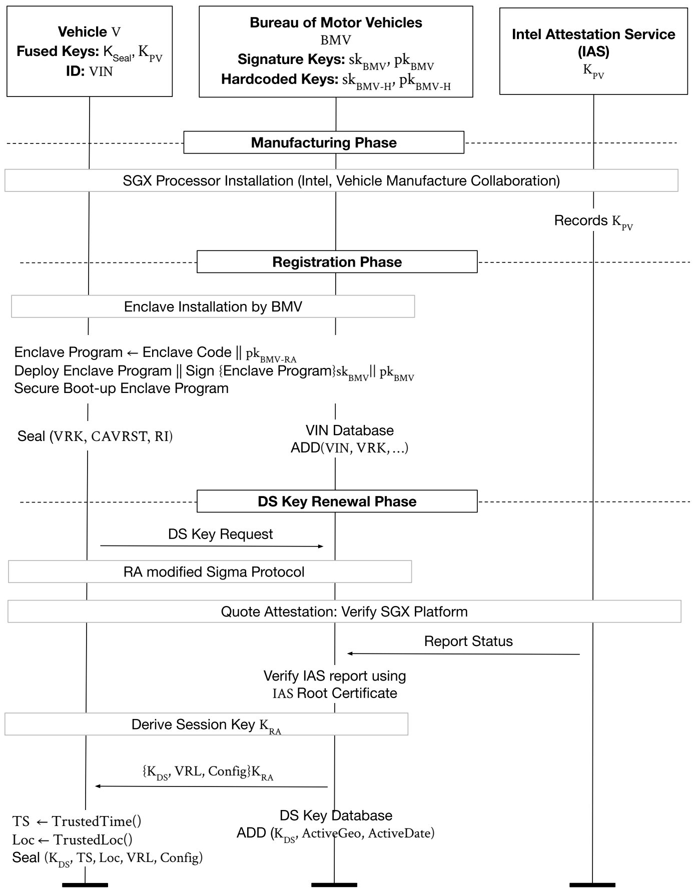

Vehicle V
Fused Keys: K_{Seal}, K_{PV}
ID: VIN
Bureau of Motor Vehicles
BMV
Signature Keys: sk_{BMV}, pk_{BMV}
Hardcoded Keys: sk_{BMV-H}, pk_{BMV-H}
Intel Attestation Service (IAS)
K_{PV}
Manufacturing Phase
SGX Processor Installation (Intel, Vehicle Manufacture Collaboration)
Records K_{PV}
Registration Phase
Enclave Installation by BMV
Enclave Program ← Enclave Code || pk_{BMV-RA}
Deploy Enclave Program || Sign {Enclave Program}sk_{BMV}|| pk_{BMV}
Secure Boot-up Enclave Program
Seal (VRK, CAVRST, RI)
VIN Database
ADD(VIN, VRK, ...)
DS Key Renewal Phase
DS Key Request
RA modified Sigma Protocol
Quote Attestation: Verify SGX Platform
Report Status
Verify IAS report using
IAS Root Certificate
Derive Session Key K_{RA}
{K_{DS}, VRL, Config}K_{RA}
TS ← TrustedTime()
Loc ← TrustedLoc()
Seal (K_{DS}, TS, Loc, VRL, Config)
DS Key Database
ADD (K_{DS}, ActiveGeo, ActiveDate)

Fig. 1: Sequence diagram of the three phases of DS key design. The notation $\{D\}key$ denotes encryption of the any data D with the symmetric key K.

2) **Vehicle Registration Phase:** When a vehicle is sold (or resold), the new owner of the vehicle needs to register the vehicle at a trusted authority such as a BMV office as usual. The new procedure with CAV is that during the registration: the BMV server will install the BMV signed enclave program in the vehicle IC. The TEE IC will use signature keys pk_{BMV},sk_{BMV} for secure bootup verification of the enclave program. The enclave hardcoded keys pk_{BMV-H},sk_{BMV-H} are used for the mutual authentication of BMV and the vehicle in the connection initiation stages of the DS key renewal phase §IV-3. Also, BMV provides the vehicle a unique secret, named vehicle root key (VRK), which is a randomly generated 128-bit hex-string

$$\mathsf{VRK} = CRNG(128)$$

sealed into the enclave. The BMV server will also keep a copy of the VRK along with the Vehicle Identity Number (VIN) as well as other necessary information such as the vehicle owner, his or her metadata: driver license number, address, gender, etc., in its VIN database to trace a misbehaving vehicle and the corresponding owner. The signatures and enclave hardcoded keys also ensure that only BMV signed enclave code will be able to successfully pass the next phase and participate in the CAV network. Moreover, during the registration, the BMV server also initializes the enclave with global CAV parameters such as the Reference-Start-Time (RST) (similar to Unix epoch time) and Rolling Interval (RI), which will be used in the later phases to achieve CAV network security. The global parameter values remain the same across this CAV network. Therefore these values need to be synchronized regularly and stayed consistent across BMV servers. On the other hand, the local parameters will operate within local geographic areas as explained in the later phases (§IV-3 and §IV-4).

3) **DS Key Renewal Phase:** DS Key renewal phase is initiated on different conditions as described in Figure 2. In this phase, the enclave program in the vehicle's TEE IC will first initiate the remote attestation (RA) [24] with the BMV server to ensure that the enclave is trusted. In the protocol, the BMV server verifies vehicle's SGX platform using quote attestation with Intel IAS server. Note that RA uses a modified Sigma protocol [27] that securely establishes the connection between the enclave program, inside an SGX supported processor, and the enclave developer: BMV in our case. At the end of RA, BMV and vehicle will establish a secure session key K_{RA}. Thereafter, BMV will provide a Daily Symmetric (DS) key, Vehicle Revocation List (VRL) and CAV network configuration (Config) to the vehicle. Here, the VRL is the list of VRKs, representing revoked vehicles, which have been identified as untrustworthy by BMV; Config is the list of various CAV network parameters such as error tolerance range, DS key renewal interval, broadcast interval, etc., which are described in the later phases. These configuration parameters can be updated dynamically to improve the efficiency and security of the CAV network. After receiving the information from BMV, the vehicle processor will seal the data with its current timestamp, TS, and current geolocation, Loc, to vehicle IC storage for later use. Once the DS key is obtained successfully, the later engine starts during that day does not have to get a new DS key if we set the key renewal interval to be a day. Rather, it follows the control flow as shown in Figure 2. The DS key renewal phase can also serve as the opportunity for BMV server to update the existing VRL and update any official information to CAV network.

DS renewal is the central component of our protocol. Every DS key has corresponding active geography (ActiveGeo), where it is valid, and active day (ActiveDay), the period in which it is valid. Particularly, during the vehicle renewing, the DS key received from the same active geography will receive the same key. Further, the key will be valid only for a day within the active geography. Active geography distribution helps the BMV server to manage DS keys locally. Further, it optimizes the server load, key management, data caching, and local forensic investigation. Since most vehicles are used for a daily commute to offices and local market visits, the active geographies can be set to specific concentrated areas, such as cities or counties, or zip codes. Additionally, when a vehicle moves across active geographies, the DS key verification in the on-road-communication (§IV-4) phase will fail. This allows the vehicle to trigger a new DS key renewal request as shown in Figure 2.

For the active time period of the DS key, we considered a trade-off between efficiency, scalability, and security. Specifically, RA protocol involves three servers to provide the secure DS key. The modified Sigma protocol used in the attestation involves a long set of message exchanges. Setting the active period to hourly or minutes basis will be

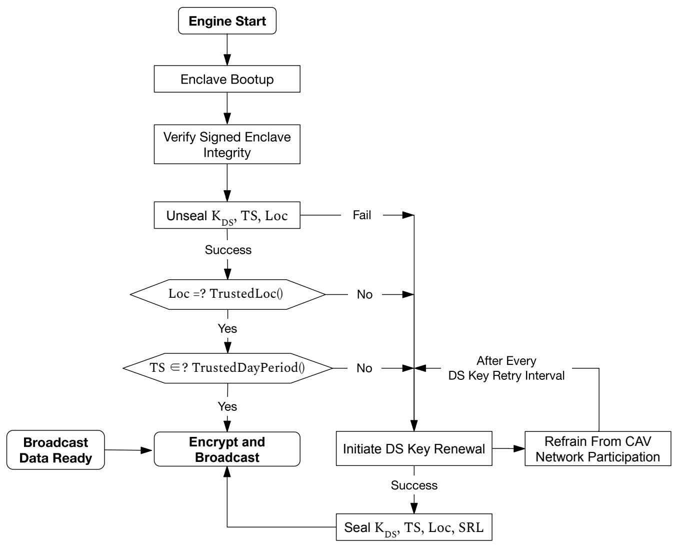


Fig. 2: Control flow of the Vehicle, whenever the engine start or a data is available to broadcast. The operators =? denote equality check; ∈? implies whether TS fall within the ActiveDay time period.

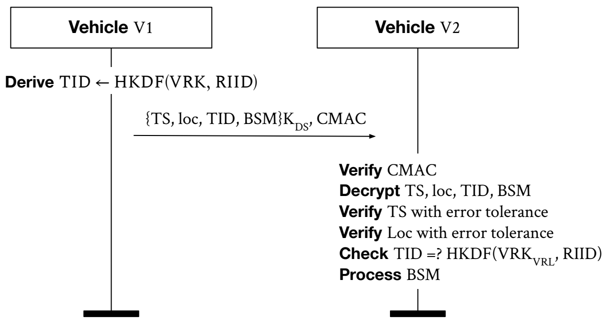
a huge communication load to the BMV and the IAS server. Additionally, it is time costly for vehicle processors to perform frequent RA. On the other hand, setting up a weekly period of DS key exposes a long time to attackers to exploit key reuse [4]. Therefore, we choose a middle ground and set the active DS key period to be per day. Further, this period can be adjusted during the alpha and beta field testing and the penetration testing of our protocol.

4) **On Road Communication Phase:** The DS key renewal phase allows each vehicle to possess the same symmetric key. Consequently, the vehicles can broadcast Basic Safety Messages (BSMs), encrypted with K_{DS}, in one shot (one way) latency. Only authenticated vehicles having the same K_{DS} would be able to decrypt the broadcast data since the same DS key is provided to all vehicles. The format of the broadcast message is:

$$\{\mathsf{TS}, \mathsf{Loc}, \mathsf{BSM}, \mathsf{TID}\}K_{\mathsf{DS}}$$

Here, the format $\{D\}key$ denotes encryption of the any data D with the symmetric key K; TS represents the timestamp of BSM generation; Loc represents the current GPS location of the vehicle. TS and Loc are generated using the Intel signed architectural enclaves, known as platform service enclaves (PSEs), which can (and will need to) provide secure location and time within trusted code. Even though Intel SGX does not support trusted locations yet, it can build such a feature for vehicular processors [30], or other RISC-V processors can be used for this purpose.

In the broadcast message (V2V and V2I), a temporary vehicle identifier (TID) is generated using the VRK, which can be considered a random seed value provided to the vehicle during the registration phase. TID is used to trace back to a misbehaving vehicle by the law enforcement authorities when needed. Inspired by temporary ID generation in digital Contact Tracing [7] designed by Apple and Google, we derive TID on a rolling basis. Specifically, TID is calculated based on



Vehicle V1
Vehicle V2
Derive TID ← HKDF(VRK, RIID)
{TS, loc, TID, BSM}K$_{DS}$, CMAC
Verify CMAC
Decrypt TS, loc, TID, BSM
Verify TS with error tolerance
Verify Loc with error tolerance
Check TID =? HKDF(VRK$_{VRL}$, RIID)
Process BSM

Fig. 3: Sequence of steps taken by vehicle sending and the receiving the broadcast message during on Road Communication in CAV network. The operator =? denote equality check.

a rolling interval, denote by RI, current timestamp TS, and the corresponding vehicle's VRK. The rolling intervals are discretized into unique numbers using an RST synchronized by BMV authorities. The unique rolling interval ID (RIID) is calculated as follows:

$$\mathsf{RIID} = \frac{\mathsf{TS} - \mathsf{RST}}{\mathsf{RI}}$$

Thereafter the TID is generated using a HMAC-based key derivation function (HKDF) as follows:

$$\mathsf{TID} = HKDF(\mathsf{VRK}, \mathsf{RIID})$$

§V-F explains how the broadcast data content can be used by law enforcement to trace back misbehaving vehicles when needed. Similarly, §V-C explains how TID preserves the privacy of a vehicle.

Also, note that for the road side infrastructures, we assume that they will have TEE protected DS key as well. They will also communicate with the regional BMV servers to refresh the DS key. Otherwise, they will not be able to decrypt the broadcast message received from CAVs. Also, since they are public infrastructures (no privacy), they do not need any TID-type of identity in the message, and instead they will have a unique infrastructure ID in their broadcast message.

V. Analysis

In this section, we describe how our TEE-based V2V protocol achieves the proposed security and privacy goals.

A. DS Key Confidentiality

The secure bootup of verification of the BMV signed enclave establishes the integrity of the enclave program. The security of RA protocol is established by the enclave hardcoded keys, signature keys and a shared fused provision key K_{PV} which is shared among only vehicle SGX hardware and Intel Attestation Services (IAS). The quote attestation [24] using the provisioning key, as shown in Figure 1, establishes secure authentication of the in-vehicle SGX platform. During the modified Sigma protocol [27] the session key K_{RA} is accessed only inside the trusted code. Combining the secure bootup, SGX platform authentication, and the trusted code execution during RA protocol protects the DS key leak to attackers.

After securely obtaining the DS key from a BMV server, each vehicle seals the confidential data (e.g., DS keys, VRK, VRL). Since the sealed data is encrypted with fused keys accessible only within the SGX hardware (not visible to the software), the sealed data is inaccessible to the host OS attacker. When the vehicle engine and the processor are running, the confidential data stays inside the protected and encrypted memory regions. These data are decrypted by TEE hardware on the fly. Therefore any untrusted code application or compromised vehicle attacker would not be able to obtain the confidential information. Further, all the V2V communication operations (encryption, decryption, verification, obtaining trusted location or trusted time) occur within the trusted code and therefore cannot be obtained by an attacker.

B. Two way authentication and integrity

The authentication, in our design, relies on the security of the DS key as explained in §V-A. Since each vehicle receives the same DS key from the trusted authority, e.g., the corresponding BMV, only authenticated vehicle will be able to decrypt the data and verify the integrity using CMAC verification. Without the valid DS Key, no vehicle will be able to join the network and communicate with others.

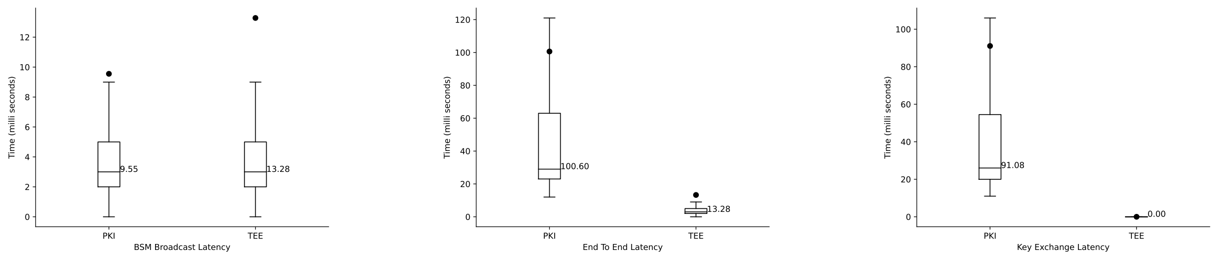
C. Privacy

The broadcast data, $\{\mathsf{TS}, \mathsf{Loc}, \mathsf{BSM}, \mathsf{TID}\}\mathsf{K}_{\mathsf{DS}}$, contains only one identifier value: TID. The rest of the encrypted values do not belong to a specific vehicle. Therefore, TID calculations should account for vehicle privacy. In order to preserve the anonymity of a vehicle, TID is changed on every rolling interval. By frequently changing TID, it should be fairly complex for an attacker to fingerprint a vehicle based upon TID information present in the broadcast data. Tracing a vehicle is further hardened by the high speed tracing demand of rapidly moving vehicles. Meanwhile, note that the attacker only observes a broadcast data cipher, i.e., she does not see the TID in plaintext. She needs other resources to break the encrypted broadcast data before obtaining the TID in the first place.

D. Replay Protection

Once a BSM is generated from current traffic, it is useful only for a certain period of time denoted as the BSM validity period. After that period, the BSM utility is deprecated. By including timestamp and location in the encrypted broadcast packets the attacker cannot replay the packets outside the BSM validity period or at a different geographic location. She can only replay the packets within the BSM validity period from the same location error tolerance range. In particular, as shown in the on-road communication phase sequence (Figure 3) the receiving vehicle verifies the data validity by comparing the timestamp TS and location Loc attribute, present in the broadcast data, with the trusted time and location obtained from the TEE processor. A replay of broadcast data with an invalid timestamp or location, obtained from a previous communication, will be discarded. Note that a replay of the broadcast data that contains the attributes which fall within the error tolerance range is still possible. However, by considering the rapidly changing traffic environment and by choosing a wise error tolerance range, the probability of such a replay attack can be made negligible.

Further, an attacker from a compromised vehicle can feed an old sealed data into the vehicle IC processor to broadcast invalid information. In this case, trusted monotonic counters can be used to prevent the replay. Specifically, a vehicle



(a) BSM Broadcast Latency (b) End To End Latency (c) Key Exchange Latency

Fig. 4: Latencies of TEE and PKI based V2V protocol prototypes out of 500 peer to peer message broadcast.

can store the monotonically increasing counter value in the sealed package whenever it uses the sealed data. The trusted monotonic counter values are protected by the hardware and are inaccessible to the software. Therefore, each sealed data counter value can be verified against the recent trusted monotonic counter value to discard the old replay of sealed data.

E. Accountability and Non-repudiation

Since each piece of broadcast data contains TID derived from vehicle VRK and the timestamp TS. Given the timestamp TS_M and the TID_M, and the broadcasted data from a misbehaving vehicle M, the BMV server database can be searched to find the corresponding VRK and TS combination that calculates the suspicious vehicle VRK_M and VIN_M. Further, since the VRK value in each vehicle is unique and random, two vehicles cannot produce the same TID. As a result, the misbehaving vehicle owner cannot present another vehicle producing a broadcast message with the same TID_M.

As shown in Figure 1, BMV keeps two databases: VIN database (registration phase) and DS key database (DS Key renewal phase). Only authorized parties can access these databases. Therefore, for any broadcasted data information only authorized officials can trace back to a vehicle. Once a misbehaving incident with the respective broadcasting data is reported, the authorities can look into DS key database to get the DS key for the incident date $\mathsf{ActiveDate}$ and incident active geography $\mathsf{ActiveGeo}$. After extracting TID_M and TS_M from the misbehaving incident broadcast data, the authorities can probe into the BMV VIN database to trace the misbehaving vehicle VIN_M. In particular, the RIID_M for the misbehaving vehicle M can be calculated as:

$$\mathsf{RIID}_M = \frac{\mathsf{TS}_M - \mathsf{RST}}{\mathsf{RI}}$$

Thereafter, all VRK_i entries in the database can be probed to match the misbehaving broadcast TID_M as following:

$$\mathsf{TID}_M =?\ HKDF(\mathsf{VRK}_i, \mathsf{RIID}_M)$$

F. Revocation

Once a vehicle receives VRL at the DS key renewal phase, it can check every incoming broadcast data for revoked status. In particular, after extracting the TS_S, TID_S from the broadcast data of a source vehicle S, the broadcast receiving vehicle (as well as the road side infrastructures) can probe the all the VRK_i entries in the VRL to identify the revoked vehicle as follows:

$$\mathsf{TID}_S =?\ HKDF(\mathsf{VRK}_i, \mathsf{RIID}_S)$$

where the RIID_S is calculated as:

$$\mathsf{RIID}_S = \frac{\mathsf{TS}_S - \mathsf{RST}}{\mathsf{RI}}$$

If the TID_S is matched with any entry in the VRL, the receiving vehicle can simply discard the broadcast data, or report to law enforcement on the location of S.

VI. Preliminary Results

A. Experiment Setup

We have implemented a preliminary prototype of our TEE-based V2V protocol using sender and receiver enclave programs with two Intel SGX enabled laptops. The programs use Intel SGX APIs for encryption-decryption, AES-GCM encryption/decryption, and sealing-unsealing operations. Further, both applications use socket programming to send and receive broadcast data over wireless adapters available on the laptop. In our experiments the sending device stocks Intel© i7-8550U CPU @ 1.80GHz × 4, 16GB RAM, Linux Mint 20.1 OS with Intel® Dual Band Wireless-AC 7265 wireless adapter. The receiving device is equipped with Intel® i5-6200U CPU @ 2.30GHz × 4. 4GB RAM, Ubuntu 18.04.6 LTS with Intel Corporation Wireless 3165 wireless network adapter. As of this phase of prototype in the enclave programs, we have not implemented trusted time, trusted location, and trusted monotony counter APIs as these trusted services are not available in the current version of Intel SGX SDK.

In order to compare with the PKI-based approach, we also implemented a preliminary certificate-based V2V protocol. Initially, both devices were equipped with trusted root CA certificates. In the protocol, both devices first exchange their CA-signed certificate (two-way exchange) to each other along with the public parameter of the ECDH key exchange algorithm. The received certificates are verified using the root CA certificate on both ends. At the end of the two-way key certificate exchange, both devices derive a shared the ECDH symmetric key. This key is used to exchange the broadcast data in the third message exchange. Overall the two-way communication authenticates the two devices and the final third exchange allows them to broadcast the BSM.

B. Results

In the preliminary performance tests, we obtained an average latency over 500 runs for peer to peer end-to-end message preparation and broadcast from sending device to message receive and processing at the receiving device. The TEE based prototype recorded an average latency of 13.28 milliseconds.

Whereas, for PKI based prototype, 91.08 milliseconds latency was recorded for key exchange and 100.6 milliseconds for complete key exchange and broadcast transmission. The overall metrics are presented in Figure 4.

Note that an on-road vehicle could be broadcasting to many other N surrounding vehicles simultaneously. In that case, any optimization applied to one peer-to-peer latency will result in N fold speedup in communication. Having a pre-shared key in our V2V design has such advantage of not requiring key establishment latency.

Since our prototype is still in the development phase with the basic optimization and architectural assumptions, the sights drawn at this moment are for introductory guide and building intuition, so far these programs are not optimized to the full capacity from an algorithm and library implementation perspective. Also, the latencies in our evaluation include environmental noises of OS processes switch, programming language abstraction, wireless packet drops which are present in both the experiments.

We observed that both protocol latencies are closer to the defined limits of 100-150 milliseconds [18], [32]. Also, the relative comparison between both the protocol suggests that the TEE-based protocol could be faster and practically usable.

VII. Limitations and Future Work

The proposed TEE-based V2V protocol is only an initial step, and it certainly has limitations. For instance, the security of our protocol is anchored upon the the confidentiality of DS key. If an attacker gets access to this key, she can exploit the CAV network within the DS key active geography and the active day. Meanwhile, compared to non-TEE-based V2V protocol, our protocol requires an extra latency of protected code operations (obtaining trusted location, trusted time), and context switch among trusted and untrusted code execution which can have additional delay. In addition, many operations of our protocol is not optimized, particularly its scalability and efficiency. In the following, we outline a few avenue for further improvement of our protocol.

Scalability. DS Key renewal from a large set of vehicles could be a huge traffic load for a BMV server. This can be mitigated by storing the BMV server databases into dedicated cloud server managed by stationary Roadside Infrastructure Units (RIUs). The RIUs can act as a gateway to lighten up the network bandwidth load. Further, RIUs can use distributed programming algorithms using leader election and consensus algorithm to provide reliability and duplication of the BMV database. The size of database can be reduced by caching DS key databases based on the geographical locations.

Efficient Authentication. Each vehicle can broadcast a safety message encrypted by DS key, in on-shot latency, without needing any other communication before or after the broadcast. One shot latency, avoids the latencies of session key establishment, certificate verification and reduces the use of heavy asymmetric key encryption or decryption. Also, as long as the vehicle processor is powered on, the TEE processor will use protected memory regions to keep confidential data (e.g., DS keys, VRK, VRL). This would avoid unsealing operations latency on every broadcast.

Efficient Revocation. In addtion to updating the VRL at every DS key renewal phase, the vehicle network could utilize various revocation list update algorithms [14], [17] to keep an updated VRL. Additionally, a vehicle can use probabilistic algorithms with statistical distribution models [35] to speedup revocation and improve confidence on security.

Efficient Tracing of Misbehaving Vehicle. Misbehaving vehicle trace search can be optimized based upon the location used in the broadcast data. In particular, the BMV server database search can be probabilistically prioritized to search only of the vehicles within the location data obtained from the broadcast data of the misbehaving vehicle.

VIII. Related Work

Previous research efforts have used various approaches to solve partial components of CAV communication. These approaches [19], [21] have used public key infrastructure (PKI) servers, digital certificates, group key cryptography pseudonym identifiers, bi-linear mapping, etc. Meanwhile, various task groups: IEEE 1609.2 [6], IEEE 820.11bd [2], 3rd Generation Partnership Project (3GPP) [8], Security Credential Management System (SCMS) [13] have been actively working on standardizing and improving CAV communication. Additionally various survey papers [19], [21] clearly describe CAV research overarching motivations (road safety, transport efficiency, and economic benefits); security objectives; threat model; CAV jargon; challenges of scalability and fast operations for vehicle ad hoc network demand.

Close to our work, CVShield [23] uses Arm TrustZone to prevent data spoofing by isolating sensor and processor communications RIUs. It assumes software compromise in the vehicle processor and provides initial performance results on TEE sensor data read operations. It optimizes read operations to improve communication efficiency. Lie et al. [30] provide software abstraction for vehicle sensors to prevent false fabrication of data. It uses TEE features such as, software attestation and sealing storage to build trust and protect sensor IO. Our work build-up from these efforts and assume that the TEE designers can build in-vehicle TEE IC to protect both sensor and the processor. Thereafter, we design an end-to-end CAV communication protocol, leveraging TEE features, for CAV network security and optimization.

Our privacy protection design is inspired by the recent digital contact tracing protocols [36] such as the Apple and Google notification exposure [7]. Particularly, we borrow the same idea of using the ephemeral ID for a vehicle, and also use a root key to derive the vehicle ephemeral ID. The matching of a misbehaving vehicle in the centralized database, as well as whether other vehicles have encountered it, also works similarly as the matching of contagious COVID-19 patient.

Finally, there are also numerous efforts of using TEEs for various client side security applications, and these include web application sandboxing by AccTee [20], Tor security and privacy enhancement by SGX-Tor [26], and cheating prevention of computer games [11], [33]. Our protocol design is inspired by these efforts, but go beyond by exploring the new applications of TEE for CAVs.

IX. Conclusion

We have presented a TEE-based V2V protocol of an end-to-end CAV communication network that meets the basic security and optimization requirements, such as confidentiality, authentication, replay protection, privacy, and revocation.

Moreover, our protocol also allows dynamic configuration of CAV network parameters to improve efficiency without changing the core design infrastructure. As future work, we will perform more feasibility and performance studies to strengthen the design choices and formally prove the security of our protocol. We hope our TEE-based design can inspire the further improvement of current CAV communication standards.

Acknowledgement

We thank the anonymous reviewers for their invaluable feedback. This research was partially supported by NSF Awards 2112471 and 2118491, as well as a Cisco Research Grant. Any opinions, findings, and conclusions or recommendations in this paper are those of the authors and do not necessarily reflect the views of the NSF and Cisco.

References

[1] "Ensuring American Leadership in Automated Vehicletechnologies," https://www.transportation.gov/sites/dot.gov/files/docs/policy-initiatives/automated-vehicles/360956/ensuringamericanleadershipav4.pdf, (Accessed on 12/24/2021).

[2] "IEEE P802.11-task Group Bd (ngv)," https://www.ieee802.org/11/Reports/tgbd_update.htm.

[3] "Intelligent Transportation Systems Joint Program office," [Online] https://www.its.dot.gov/index.htm.

[4] "NIST Recommendation for Key Management: Part 1, sp 800-57 part 1 rev. 5, 2020," [Online] https://csrc.nist.gov/publications/detail/sp/800-57-part-1/rev-5/final.

[5] "Traffic Safety: raising Spending Could Save Lives and Money," https://www.cnbc.com/2015/12/14/traffic-safety-raising-spending-could-save-lives-and-money.html. Accessed on 01/04/2022.

[6] "IEEE Standard for Wireless Access in Vehicular Environments–security Services for Applications and Management Messages," *IEEE Std 1609.2-2016 (Revision of IEEE Std 1609.2-2013)*, pp. 1–240, 2016.

[7] "Privacy-Preserving Contact Tracing," *Google and Apple*, 2020, https://covid19.apple.com/contacttracing.

[8] G. T. 22.891, "Feasibility Study on New Services and Markets Technology Enablers," 2016.

[9] I. Anati, S. Gueron, S. Johnson, and V. Scarlata, "Innovative Technology for CPU Based Attestation and Sealing," in *Proceedings of the 2nd international workshop on hardware and architectural support for security and privacy (HASP'13)*, vol. 13. ACM New York, NY, USA, 2013.

[10] A. ARM, "Security Technology Building A Secure System Using Trustzone Technology," *ARM Limited*, 2009.

[11] E. Bauman and Z. Lin, "A case for protecting computer games with sgx," in *Proceedings of the 1st Workshop on System Software for Trusted Execution (SysTEX'16)*, Trento, Italy, December 2016.

[12] N. Bißmeyer, H. Stübing, E. Schoch, S. Götz, J. P. Stotz, and B. Lonc, "A Generic Public Key infrastructure for Securing Car-to-x Communication," in *18th ITS World Congress, Orlando, USA*, vol. 14, 2011.

[13] B. Brecht, D. Therriault, A. Weimerskirch, W. Whyte, V. Kumar, T. Hehn, and R. Goudy, "A Security Credential Management System for V2x Communications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 12, pp. 3850–3871, 2018.

[14] J. Caubet, C. Gañán, O. Esparza, J. L. Muñoz, J. Mata-Díaz, and J. Alins, "certificate Revocation List Distribution System for The KAD Network," *The Computer Journal*, vol. 57, no. 2, pp. 273–280, 05 2013.

[15] S. Cen and B. Zhang, "Trusted Time and Monotonic Counters with Intel Software Guard Extensions Platform Services," *Intel White Paper*, 2017.

[16] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, T. Kohno *et al.*, "Comprehensive Experimental Analyses of Automotive Attack Surfaces." in *USENIX Security Symposium*, vol. 4. San Francisco, 2011, p. 2021.

[17] I. Curry and P. C. Van Oorschot, "Computer Security System and Method With on Demand Publishing of Certificate Revocation Lists," Oct. 3 2000, uS Patent 6,128,740.

[18] M. H. C. Garcia, A. Molina-Galan, M. Boban, J. Gozalvez, B. Coll-Perales, T. Şahin, and A. Kousaridas, "A tutorial on 5g nr v2x communications," *IEEE Communications Surveys Tutorials*, vol. 23, no. 3, pp. 1972–2026, 2021.

[19] A. Ghosal and M. Conti, "Security Issues and Challenges in V2X: A Survey," *Computer Networks*, vol. 169, p. 107093, 2020.

[20] D. Goltzsche, M. Nieke, T. Knauth, and R. Kapitza, "Acctee: A webassembly-based two-way sandbox for trusted resource accounting," in *Proceedings of the 20th International Middleware Conference*, 2019, pp. 123–135.

[21] H. Hasrouny, A. E. Samhat, C. Bassil, and A. Laouiti, "VANet security challenges and solutions: A survey," *Vehicular Communications*, vol. 7, pp. 7–20, 2017.

[22] M. Hoekstra, R. Lal, P. Pappachan, V. Phegade, and J. Del Cuvillo, "Using Innovative Instructions to Create Trustworthy Software Solutions." *HASP@ ISCA*, vol. 11, no. 10.1145, pp. 2 487 726–2 488 370, 2013.

[23] S. Hu, Q. A. Chen, J. Joung, C. Carlak, Y. Feng, Z. M. Mao, and H. X. Liu, "CVSHIELD: Guarding Sensor Data in Connected Vehicle With Trusted Execution Environment," in *Proceedings of the Second ACM Workshop on Automotive and Aerial Vehicle Security*, 2020, pp. 1–4.

[24] S. Johnson, V. Scarlata, C. Rozas, E. Brickell, and F. Mckeen, "Intel Software Guard Extensions: EPID Provisioning and Attestation Services," *White Paper*, vol. 1, no. 1-10, p. 119, 2016.

[25] D. Kaplan, J. Powell, and T. Woller, "AMD SEV-SNP: Strengthening VM Isolation with Integrity Protection and More," Technical Report. Advanced Micro Devices Inc. https://www. amd. com/system ..., Tech. Rep., 2020.

[26] S. Kim, J. Han, J. Ha, T. Kim, and D. Han, "Enhancing security and privacy of tor's ecosystem by using trusted execution environments," in *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*, 2017, pp. 145–161.

[27] H. Krawczyk, "sigma: The 'SIGn-and-MAc'approach to Authenticated Diffie-Hellman and Its use in The IKE Protocols," in *Annual International Cryptology Conference*. Springer, 2003, pp. 400–425.

[28] H. Krishnan, "Vehicle Safety Communications Project,‖ camp vehicle safety communications consortium, february 15, 2006."

[29] T. Litman, "Autonomous Vehicle Implementation Predictions implications for transport planning," Victoria Transport Policy Institute, (Accessed on 12/24/2021).

[30] H. Liu, S. Saroiu, A. Wolman, and H. Raj, "Software Abstractions for Trusted Sensors," in *Proceedings of the 10th international conference on Mobile systems, applications, and services*, 2012, pp. 365–378.

[31] F. McKeen, I. Alexandrovich, A. Berenzon, C. V. Rozas, H. Shafi, V. Shanbhogue, and U. R. Savagaonkar, "Innovative Instructions and Software Model for Isolated Execution." *Hasp@ isca*, vol. 10, no. 1, 2013.

[32] G. Naik, B. Choudhury, and J.-M. Park, "Ieee 802.11bd amp; 5g nr v2x: Evolution of radio access technologies for v2x communications," *IEEE Access*, vol. 7, pp. 70 169–70 184, 2019.

[33] S. Park, A. Ahmad, and B. Lee, "Blackmirror: Preventing wallhacks in 3d online fps games," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020, pp. 987–1000.

[34] R. Quinonez, J. Giraldo, L. Salazar, E. Bauman, A. Cardenas, and Z. Lin, "{SAVIOR}: Securing autonomous vehicles with robust physical invariants," in *29th USENIX Security Symposium (USENIX Security 20)*, 2020, pp. 895–912.

[35] A. Rao, A. Sangwan, A. A. Kherani, A. Varghese, B. Bellur, and R. Shorey, "Secure V2V Communication With Certificate Revocations," in *2007 Mobile Networking for Vehicular Environments*. IEEE, 2007, pp. 127–132.

[36] H. Wen, Q. Zhao, Z. Lin, D. Xuan, and N. Shroff, "A study of the privacy of covid-19 contact tracing apps," in *International Conference on Security and Privacy in Communication Networks*, 2020.