Secure CV2X Using COTS Smartphones over LTE Infrastructure

Spandan Mahadevegowda[1(✉)], Ryan Gerdes[1], Thidapat Chantem[1], and Rose Qingyang Hu[2]

[1] Virginia Tech, Arlington, VA, USA
{spandan,rgerdes,tchantem}@vt.edu
[2] Utah State University, Logan, UT, USA
rose.hu@usu.edu

Abstract. With the proliferation of vehicle technologies to support sophisticated features like assisted and autonomous driving, advanced communication protocols like cellular-vehicle-to-everything (CV2X) have been proposed. However, practical large-scale deployments have been hindered due to caveats such as hardware, security, and cellular infrastructure demands. This work presents and evaluates a practical approach to utilizing ARM TrustZone to turn commercial off-the-shelf smartphones into secure CV2X radios that communicate over the LTE network. These smartphone-based CV2x radios communicate with each other via an intermediary server placed outside/within the LTE infrastructure without affecting normal operations of the phone, like using navigation, calls, and music. Vehicle owners would only have to install the CV2X application to use their smartphones as CV2X radios. The approach would boost the adoption of CV2X by reducing the requirement for dedicated hardware and reusing existing infrastructure. In this work, we empirically evaluate the on-device overhead coupled with various network topologies concerning the location of an intermediary server and the LTE infrastructure. We show that our proposed approach can meet the required real-time constraints for safe CV2X operation while ensuring the integrity of the on-device communication from manipulation by remote attackers.

Keywords: CV2X · COTS devices · LTE · Secure communication · TEE · Trustzone

1 Introduction

With the advancements in communication and processing technologies, modern vehicles and traffic infrastructure have become smarter, providing improved safety, security, and overall commuting experience. Today, we have automated

This work was supported by the National Science Foundation (NSF) under grant numbers 2038726 and 1941524.

F. Li et al. (Eds.): SecureComm 2022, LNICST 462, pp. 588–607, 2023.
https://doi.org/10.1007/978-3-031-25538-0_31

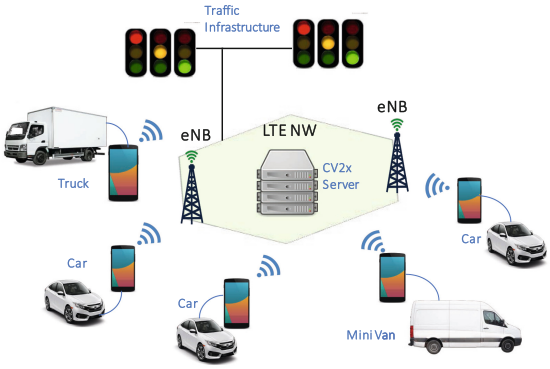


Fig. 1. Secure COTS based CV2X overview

driver assistance, cruise control, pedestrian detection, etc., that can save lives and increase commuter comfort while decreasing commute time. However, establishing connectivity among vehicles and the infrastructure is essential to enable the large-scale integration of such data-driven technologies. Features such as over-the-air updates and diagnostics of vehicle software have only been made possible from such connectivity [1]. One widely explored and investigated solution is the cellular vehicle to everything (CV2X) framework, which uses a high-speed cellular network to interconnect vehicles and traffic infrastructure. However, this has caveats. Enabling CV2X would mean significant investment in developing dedicated software and hardware. Moreover, it would require tedious integration of cellular modules into deployed vehicles or even mean waiting to replace older vehicles with new models. Additionally, enabling such complex communication means the exposure of vehicles to malicious entities. Research has already shown possible attacks and issues on vehicles made by Tesla with wireless/cellular connectivity that can threaten passenger safety and security [2,3].

Also, installing dedicated CV2X radios by vehicle manufacturers will likely require government mandates. Opportunely, modern-day smartphones that possess advanced processing power are ubiquitous. Moreover, these smartphones also come equipped with software and hardware techniques to provide secure data processing and privacy. For example, watching licensed media on Netflix, or securely storing/using fingerprints to perform financial transactions, utilizes said underlying software or hardware security mechanisms, such as TrustZone, a trusted execution environment on ARM-based processors [4]. Given how ubiquitous smartphones are, we could leverage smartphones instead of dedicated cellular modules to securely tether vehicles to the CV2X framework. The smartphone's universal serial bus (USB) port can be connected to the vehicle's on board diagnostics (OBD/OBD-II)port [5] with a dongle.

Vehicle owners would only need to install a secure application on their smartphones to use them as a CV2X radio while not significantly impacting the performance of other applications like navigation or music. However, since these cellular devices do not inherently support CV2X protocols and frameworks, we need to provide external servers that will behave as intermediaries to connect

these smartphones to the CV2X framework. Furthermore, connecting cellular devices on commercial networks through intermediary servers to enable CV2X protocol still needs to adhere to low latency, high throughput, etc., for safety-critical CV2X applications. Figure 1 shows a high level overview of the framework. Smartphone tethered vehicles are communicating with each other for the purpose of CV2X via the intermediary server on the long term evolution (LTE) infrastructure. The exchanged information is used to communicate commands or data to the vehicles for assisted driving, warnings, etc. Therefore, our contributions are:

- We provide a framework for CV2X connectivity using commercial-off-the-shelf (COTS) cellular devices and analyze intermediary CV2X server on LTE/4G network considering various topologies. This work is also the first attempt to establish secure CV2X connectivity using COTS cellular devices to the best of our knowledge.
- We provide a secure framework using ARM TrustZone to mitigate security vulnerabilities such as replay and denial of service (DoS) attacks on the COTS devices running the CV2X application.
- We study and evaluate various network topologies by placing the CV2X server in various locations within the cellular infrastructure to ensure the real-time deployability of our approach.
- We evaluate and provide quantitative measurements for the overheads and latency incurred in the framework. Consequently, we analyze and discuss the feasibility of using COTS devices for CV2X.

2 Related Work

The objective of our work is provide a secure framework for CV2X using ARM TrustZone, a trusted execution environment (TEE) while analyzing the feasibility of using LTE as the underlying cellular network. Considerable work on using trusted execution can be found in literature. Similarly work on analysis and evaluating LTE for CV2X has been pursued for sometime. However, since our work looks at an intersection of both these research objectives, we discuss some relevant literature with overlap of such technologies. The use of trusted execution environment for real-time network use cases in internet of things has been studied in [6]. The authors discuss and evaluate how TrustZone can be used to meet IOT real-time requirements while ensuring security, specifically in the case of connected industrial systems. [7] discusses enhancing mobile application security using TEE with user services identity module (USIM) to provide secure billing and payment services. Ali Raza discusses the use of TEE on COTS devices used by public safety officers to secure mission critical services for public safety over private and commercial LTE networks in the work [8]. To enhance privacy and enable private conversations on mobile devices, Amit Ahlawat and Wenliang Du propose and evaluate a TrustZone based secure voice over internet protocol (VoIP) application in [9].

Given, the real-time demands of the CV2X applications, it is essential to study latency, packet drop and throughput of the network extensively. Sheng

Liu et al. provide an empirical study of LTE-4G based CV2X performance in [10]. The authors also study the performance of dedicated short range communication (DSRC), another vehicle-to-vehicle communication protocol and compare the two frameworks. [11] evaluates and studies the use of 4G LTE network for IoT applications with real-time latency demands less than 100ms. The paper discusses the feasibility of using LTE for such applications with fixed packet sized transmission. [12] discusses the latency radio access network bottle neck issue in the LTE network for V2X applications and evaluate the performance for latency critical use cases. [13] describes a study to measure the possibility of automated driving using V2X connectivity with 4G-LTE. The authors use certain assumption to restrict the physical proximity of the LTE subsytems and evaluate a real-world vehicle setup suggesting that V2X applications are possible with LTE under certain conditions.

While these prior works consider dedicated and trusted hardware, our approach tries to utilize smartphones connected to vehicles in lieu of such hardware, aiding in the acceleration of adopting and using CV2X. Additionally, our work tries to analyze and compare various network topologies with our COTS based CV2X device deployed on the same.

3 Preliminaries

This section provides an overview of the underlying technologies, ARM Trustzone, LTE Architectural Overview, and Open Portable Trusted Execution Environment (OP-TEE). We also include details on external constraints. For brevity, we only highlight the components relevant to this work, readers can refer to the cited work for more details.

ARM Trustzone: ARM Trustzone [14] is a hardware extension on ARM Cortex devices that provides a platform for creating and executing trusted execution environments via software. The extension divides the processor execution into secure and non-secure domains. The secure domain has access to the information and peripherals of both domains. In contrast, the non-secure domain can only access its data, code, and peripherals. Software or hardware modules within the secure extension like secure memory controller (SMC) and secure peripheral controller (SPC) control the access permissions. In the case of Cortex-A devices, ARM provides a default secure monitor called the ARM Trusted-Firmware (ARM TF) that acts as an intermediary to securely switch between the domains, preventing information leaks and unauthorized access. Whenever the non-secure domain wants to process critical data or a secure interrupt gets raised, the ARM TF discerns the call and routes it to the secure domain after handling the relevant context switches. Similarly, the ARM TF also handles the switching back to the non-secure domain.

Open Portable Trusted Execution Environment (OP-TEE): OP-TEE [15] is an open-source implementation of TEE that follows the GlobalPlatform Trusted Execution Environment specifications [16]. It consists of a secure kernel with a range of cryptographic libraries that accepts requests from the non-secure domain via a secure monitor to perform cryptographic operations and process critical data.

Long-Term Evolution (LTE): LTE [17] is the 4th generation of the universal terrestrial radio access network (UTRAN). The main components of the LTE infrastructure are the cellular devices or user devices(UE) like smartphones, radio access network (RAN), Evolved UTRAN Node (eNB), and the Evolved Packet Core (EPC). The UEs connect to the eNB at a given locality over RAN. The eNB schedules and controls the access of the UE connections and resources. Multiple eNBs connect to the EPC via dedicated interfaces specified by 3rd Generation Partnership Project (3GPP). The EPC is primarily composed of four components, (i) the Mobility Management Entity (MME), (ii) Home Subscriber Server (HSS), (iii)Serving Gateway (SGW), and (iv) Packet Gateway (PGW). MME and HSS are responsible for the user control signaling that authorizes and collects billing data, handles authorization and identification of users. The latter two components, SGW and PGW are the packets switching components that connect the UEs across networks or the internet for calls/data.

Latency Requirements: Many CV2X applications such as driver assistance, auotmated braking, etc. are safety critical in nature. Therefore, we have explicit latency requirements for exchanging information. European Telecommunications Standards Institute (ETSI) and U.S Department of Transportation (DOT) have analyzed various traffic and vehicle scenarios and laid out latency demands corresponding for the same [18]. These latency requirements are based on the scenario for vehicular applications. For example, a 100 ms latency is required for periodic vehicle-2-vehicle/pedestrian/infrastructure applications such as emergency electronic brake, stop sign assistance, lane change assistance, etc. Similarly, 1000 ms latency is mandated for vehicle-2-infrastructure warnings, while pre-crash sensing applications have a requirement of 20 ms. In order to cover majority of the scenarios and establish a basic CV2X framework, we set our latency constraint for 100 ms.

Basic Safety Messages (BSMs): BSM is a standardized messaging specification developed for V2X applications. Originally standardized as an ASN.1 encoded message in the SAE J2735 [19] specification, it carries information related to vehicle position, speed, direction and other safety extension data. In our work, we restrict the payload to carry only the position, speed and some vehicle information primarily obtained using a GPS [20,21] module. However we refer to this payload as BSM in the rest of the paper and it has an average size of 67 bytes.

4 System and Threat Models

The smartphones/user equipment (UE) capabilities and architecture considered for our CV2X framework, as well as attacker capabilities are discussed.

4.1 System Model

UE and Vehicles. The UE is an ARM Cortex-A device with TrustZone hardware extension and secure timers. It houses an LTE modem and a GPS module with typical cell phone peripherals. The TrustZone extension divides the processing/OS (Fig. 2) into secure (red) and non-secure domains (green). The device also has secure hardware extensions to control and configure access rights to peripherals and memory like the Secure Peripheral and Secure Memory Controllers.

Here, we perform the required cryptographic and secure processing using trusted applications and return relevant data to the non-secure world.

Finally, we expect that the drivers and/or the passengers connect their smartphones with the vehicle via the OBD-II port. Please note that from here on, we refer to smartphones and cellular devices as UEs. Vehicles tethered to the CV2X network via the UEs are referred to as UE enabled vehicles.

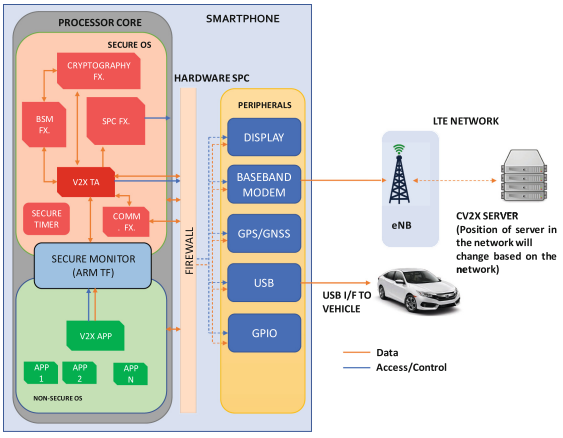


Fig. 2. Architectural overview of the UE with secure CV2X application

LTE Network and Server. There has been considerable research on using Device to Device (D2D) and dedicated side channel PC5 [22,23] for V2X communication on the LTE network. However, D2D and PC5 solutions require additional hardware provisions, which are not guaranteed to be available on LTE modules of smartphones. Therefore, to support majority of smartphones, we instead deploy a CV2X server facilitating the connection of various UE-enabled vehicles. The server after receiving BSMs from UEs, filters relevant BSMs based on GPS proximity and positional data within the BSM. These filtered BSMs

are sent as response to relevant UEs. Given the latency constraint of the CV2X safety-critical applications, we assume that the server is equipped with sufficient processing capabilities and can communicate at the required bandwidths with negligible overheads in data transmission.

For the LTE network, we consider the most common setup of the commercially available 3GPP standardized architecture [17]. We do not delve into the specifics of the LTE network itself, given its vast expanse of intricacies and technicality. Instead, we look at the LTE at the network level, which suffices for the most part of our work. UEs attach to the eNB over RAN, and multiple eNBs connect to the EPC. Though the eNBs are interconnected over the X2 interface for handovers of moving UEs, we currently ignore this to simplify analysis. Data packets for general applications or calls on the UE take the traditional user data plane to connect to the internet or other UEs over the SPGW. Data/BSMs to and from the secure CV2X application can either follow the traditional path or modified paths depending on the position of the CV2X server within the LTE network. The four potential locations for the CV2X server would be (i) on the internet, (ii) at the end of the EPC, (iii) as a part of the EPC, and (iv) at the end of eNB as an edge computing server. We analyze the above network topologies in detail in Sects. 6 and 7, for ease of deployability and latency.

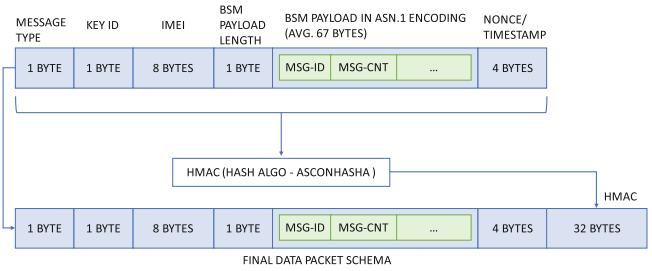


Fig. 3. Modified BSM data packet including security overhead

4.2 Threat Model

We consider an attacker that can access the smartphone remotely on the non-secure domain to gain complete access and control of the UE connected to the external world via internet or wireless interfaces. Since the attacker has complete control of the non-secure kernel, they have complete access to the LTE modem and the GPS module. Therefore, the attacker can read, modify and delete their respective data. Thus, the attacker can either corrupt/replay data or launch denial of service (DoS) attacks on the modem and GPS modules.

It is assumed that the attacker does not have physical access to the system. Any attack that requires physical access is considered out-of-scope for this work. Also, we assume that the integrity of the ARM Trustzone and OPTEE cannot be compromised. We consider that the CV2X application is installed in a secure

environment and not manipulated before the installation, ensuring the integrity of the secure domain application. In this work, we also assume that the attacker is not capable of modifying user input by taking control of user interfaces such as physical buttons/touchscreen. However, this is a minor limitation and can be addressed as discussed in Sect. 5.4

5 Secure CV2X Framework

We now discuss the architecture, functionality and application flow of our Secure CV2X framework on COTS UE. Our approach considers both regular and attacker controlled (see threat model in Sect. 4.2) scenarios.

5.1 Secure CV2X Architecture

Our approach deploys a CV2X framework using UEs tethered to the vehicle over the OBD-II port. Commands or data for the vehicle are transmitted based on the exchanged location, speed, and direction information with nearby vehicles and infrastructure.

Recall from Sect. 3 that such information exchange requires very low latency. Inability to meet these demands, for example, not exchanging BSMs in a timely manner with nearby vehicles when performing maneuvers such as lane changes, may threaten passenger safety. Additionally, attackers may manipulate data and inject delays to enforce hazardous operating conditions. Therefore, to

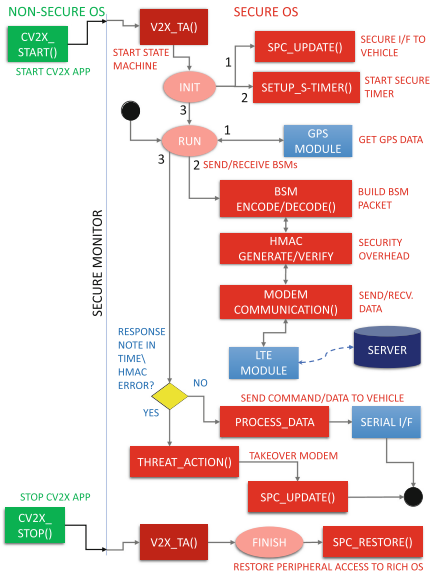


Fig. 4. Secure CV2X framework application flow

enable secure and low overhead BSM exchange, our CV2X application (Fig. 2) is deployed within the secure domain of the vehicle owner's TrustZone equipped UE. Our framework consists of a trusted application (CV2X TA) that sets up the CV2X state machine for periodically exchanging BSMs with other vehicles. The secure OS, OP-TEE, is modified to support the CV2X TA. The changes include the ability to encode and decode the ASN.1 BSM specification, support cryptography operations for encryption and hashing, modifying the SPC to control access to peripherals, drivers to utilize the LTE and GPS modems, and secure timer interrupt configuration to periodically trigger BSM exchanges.

Before we delve into how the CV2X TA operates, we first discuss the modifications required to the BSM data packet.

5.2 Modified BSM Data Packet

Figure 3 shows a modified BSM data packet with security overhead. Important additions include the following fields: a) "Message Type" to identify type based on the SAE J2735 specification [19], b) "Key ID" to allow us to identify rotated keys (to prevent brute-force key extraction) for HMAC [24] computation, c) "IMEI" that allows the CV2X server to identify individual UEs communicating with it without needing to perform an expensive lookup via the Home Subscriber System (HSS), d) "Payload Length" to support BSM packet compression by dropping unused bytes, and e) nonce that consists of a truncated timestamp and ascon-hash-a [25] based HMAC generated from the payload to validate integrity of data.

5.3 CV2X TA Operation

As seen in Fig. 4, the user starts the CV2X application from the non-secure OS, after they connect the UE to the vehicle. The non-secure CV2X application invokes a Secure Monitor Call (SMC) via the OP-TEE kernel driver which instructs the secure monitor, ARM TF, to invoke the CV2X TA. This sets up the CV2X state machine, initializes the interfaces to connect with the vehicle to perform secure communication, sets up the secure timer, and changes the internal state machine to the "RUN" state. No further user interaction is required. The secure timer is configured to be triggered periodically based on the latency requirements discussed in Sect. 3. Note only code in the secure domain can modify the secure timer.

The secure timer interrupt invokes the CV2X state machine which performs the following actions. It fetches positional information from the GPS/GNSS module, and the direction and status of the vehicle via the tethered interface. The information is ASN.1 encoded similar to the original SAE J2375 [19] specification for BSMs. The keys for HMAC generation are stored in the secure domain during

the installation of the CV2X application. The CV2X TA utilizes the current timestamp as a nonce, rotates through the store to select a key to generate an HMAC based on BSM payload, which are then combined to create our final packet (Sect. 5.2). This packet is sent to the CV2X server. The server responds with data containing BSM packets of nearby vehicles. The response HMAC is verified before the payload is utilized for making vehicle maneuvering decisions. The state machine can be shut down by the user from the non-secure OS at the end of the commute.

5.4 Security Analysis Under Attack Conditions

Based on our threat model in Sect. 4.2, since the attacker may control the non-secure domain, the attacker could gain control over two components of our framework: LTE (and GPS) modem, and user input. We shall now look at how the system may operate when each of these components is compromised.

Compromised LTE+GPS Modem. The attacker may assume complete control over the shared LTE modem. The attacker then could modify, store and replay or deny the use of the LTE modem to the secure domain. Considering Fig. 4 and the elements of the data packet, we can detect an attack or discrepancy in the CV2X operation as follows:

- If the attacker manipulates the received data, the HMAC verification of the received data would fail. Here, we consider that the server is not compromised and use HMAC to validate any manipulations by an attacker at the device or the LTE network.
- If the attacker reads and saves a server response on the modem, then replays this message to the device to thwart the CV2X operation. The nonce in the data can be checked to validate the BSM payload freshness.
- If the attacker performs a denial of service attack on the modem by bombarding the modem with requests, or deletes data before it is read by the secure side. An internal alive counter is implemented to check for timeliness.

In all the cases, the secure application assumes that the smartphone is compromised and invokes the secure peripheral controller (SPC) function to modify the SPC to explicitly assign access rights of the modem and GPS to the secure domain. Once the access permissions are changed, the non-secure domain or the attacker in the non-secure domain cannot access the modem or data lines connected to the modem or GPS. The CV2X application would now continue in the secure safety mode and follows the flow as described in Sect. 5.3. Though the mitigation action is strict, it is necessary to ensure passenger safety. Similarly, attacker intent to disrupt any power to the UE or modem can be thwarted by taking over the power module of the device on the secure domain during the execution of the CV2X application.

Compromised User Input. The capability to start and stop the application is provided to the user. All other functionality of the CV2X application, such as generating BSM, computing HMAC and transmission are done automatically in the secure domain and are outside attacker control based on our threat model. By controlling user input, the attacker may have the capability to stop the CV2X application by overriding the user interface on the non-secure side. This can be countered by reassigning a General-Purpose I/O physical button (such as the volume rocker button), which becomes the sole mechanism to signal the CV2X application to stop. This ensures that if every other user input mechanism is compromised, the attacker cannot remotely shut-off the CV2X application while the vehicle is in motion.

6 Network Configuration and Topologies

As mentioned in the system model, we consider four network topologies based on the possible placement of the CV2X server within the LTE network as shown in the Fig. 8. In this section, we discuss each configuration in detail.

CV2X Server on the Internet. As shown in Fig. 5a, the CV2X server can be placed on the internet. However to support the real-time latency demands of the vehicular applications, the CV2X server (i) must be located at a geographical location closer to the UE to minimize round-trip latency, and (ii) must be on a high bandwidth network where its traffic is prioritized for real-time applications. Deploying the CV2X server on the internet does not require any additional modifications or constraints from the LTE infrastructure in terms of additional hardware or standards, other than the automatic selection and broadcast of the IP address of the nearest CV2X server.

Further, placing the CV2X server on the internet provides straightforward inter-connectivity between different network providers where various operators can use existing packet data networks (PDNs) to reach the same CV2X server in a given locality. This mechanism makes the CV2X server agnostic to the origin and destination of the data, but needs to still maintain the latency requirements of the applications even with tight bounds on geographical distance and high bandwidth networks. The round trip latency in this case, is the sum of the latency from (i) the EPC to the server over the internet, (ii) radio access network, and (iii) the S1 interface connecting eNB and EPC.

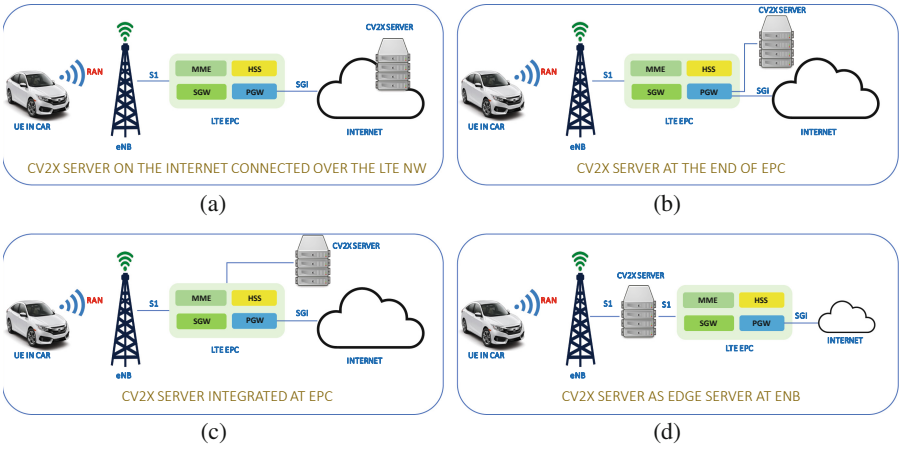


Fig. 5. Network topologies based on the position of the CV2X server in the LTE infrastructure

CV2X Server at the End of the EPC. As in Fig. 5b, the CV2X server is closer to the core LTE infrastructure when placed at the end of the EPC. This approach uses a dedicated subsystem at the end of the EPC core connected to the SPGW for CV2X, similar to IP multimedia systems (IMS) [26], which provide voice over internet protocol(VoIP), conference calls, etc. This placement strategy effectively eliminates the delays and network demands of public internet PDNs. The network round-trip delay is now reduced to latency in (i) radio access network, and (ii) the S1 interface connecting eNB and the EPC. Further, similar to IMS infrastructure, multiple network providers can exchange information across dedicated CV2X servers at the end of individual EPC centers.

CV2X Server as Part of EPC. As shown in Fig. 5c, the CV2X server can also be placed as a subsystem within the EPC core network connected to the mobility management entity (MME). However, such integration would require modifications to the standard interfaces within the LTE infrastructure. Moreover, exchanging BSMs and data across different network providers in this scenario would either require the development of novel standards for EPC subsystems or require implementing wrapper interfaces and modules to synchronize CV2X servers within EPCs across networks which could add computation and latency overheads. Due to the impracticality of placing the CV2X server as a part of the EPC, such a strategy is not included for further analysis.

CV2X Server at End of the eNB. By placing the CV2X server at the end of the eNB, we envision a mobile edge computing (MEC) platform [27,28] for the LTE infrastructure, as shown in Fig. 5d. A middlebox implementation [27] of an MEC server was recently proposed to reduce data processing latency without significant changes to the standard LTE network. This MEC server also acts

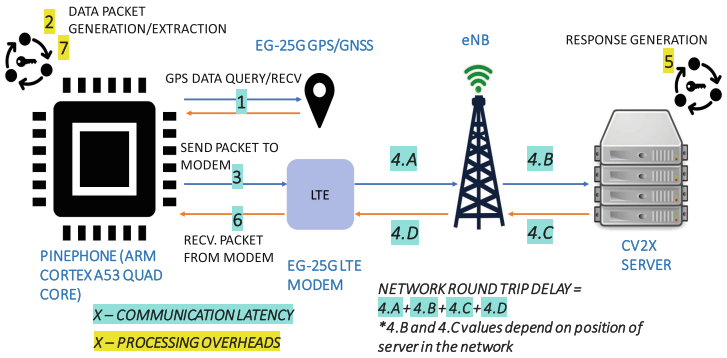


Fig. 6. Processing overheads and network latency in Secure CV2X framework

as a filter for processing local packets at the edge while forwarding other data packets to the SPGW. We modify this MEC server approach to integrate our CV2X server application, which could require having a CV2X server at every eNB. Alternatively, a server with increased computation capabilities at one eNB could be shared with other eNBs via the X2 interface.

7 Evaluation and Analysis

We discuss the proof of concept CV2X application built on open source UE and evaluate the considered network topologies. We describe the hardware, latency measurements, and simulations to understand the real-world scalability. Source code and relevant documentation can be found at https://github.com/spandan-m/secure_cv2x.

7.1 Hardware Setup

We use the ARM Cortex-A based Pinephone [29] as the UE to set up our proof of concept. The device houses a 64-bit quad-core Cortex A53 chipset at 1.152 GHz with 3 GB of LPDDR3 RAM. The device runs Arch ARM linux with kernel v5.8 as the rich OS (Non-secure domain) and OP-TEE OS v3.14.0-rc1 as the secure OS (Secure domain). The user application runs on the rich OS and allows the user to request the ARM Trusted Firmware to load the CV2X TA on OP-TEE.

CV2X TA: We implement most of the features of the CV2X TA as described in Sect. 5.1. However, since our aim is to measure latency, we omit some features such as:

– Since previous work [30,31] have already shown the effectiveness of the countermeasures, our goal here is to ensure that latency constraints are still met

when the countermeasures are used. We assume that there is no attacker during our testing, so threat mitigation mechanisms, such as controlling modem access using SPC, are skipped. However, all packet processing code are considered in order to report overheads accurately.
- For testing latencies with respect to different number of BSM response packets, our CV2X server implementation sends the required number of responses containing random data. We do not implement a realistic CV2X server for simulating network overhead. Our tests incur negligible server overheads in generating random but valid HMAC verifiable data.
- We gather the latency for GPS separately as it is unfeasible to establish assisted GPS within our lab setup.

Table 1. UE processing and communication overhead measurements

	Action	No. BSMs sent as response from server	Mean [ms]	95% CI [ms]
UE SEND	GPS Module data request and fetch	–	1.2652	1.25874–1.27172
	Building data packet + HMAC	–	0.0100	0.01002–0.01005
	Sending data packet to modem + ACK from Modem	–	1.8080	1.80800–1.80801
UE RECV	Receive data from Modem	1 BSM	4.8460	4.84371–4.84845
		5 BSM	6.3721	6.36952–6.37471
		10 BSM	8.3028	8.30093–8.30472
	Extract BSMs and Verify HMAC	1 BSM	0.0098	0.00984–0.00992
		5 BSM	0.0080	0.00800–0.00801
		10 BSM	0.0080	0.0080–0.0080
TOTAL OVERHEAD ON UE (SEND+RECV)		**1 BSM**	**7.9388 ms**	
		5 BSM	**9.4459 ms**	
		10 BSM	**11.3938 ms**	

Network Setup for Server on the Internet: As detailed in Sect. 6. We require a geographically close server with high processing and bandwidth capabilities. So, we use a server on amazon web services located about 27 miles from the test zone. We test the latency across an urban, suburban and highway terrain with vehicle speeds varying between 20–60 miles/hr.

Network Setup for Server at the End of EPC: In this topology, we established a private LTE network with the Open Air Interface (OAI) LTE eNB and EPC stacks on two computers. The eNB and EPC were hosted on Ubuntu 18.04 LTS on an Intel Core i5-6500 quad-core CPU running at 3.2 GHz with 8 GiB and 16 GiB of RAM, respectively. For the radio interface, we use USRP B210 and test the setup for 100 resource blocks of bandwidth.

Network Setup for the Server at the End of eNB: The network setup is similar to the network setup used in the case of the server at the end of EPC. Additionally, we use another computer for the MEC in between the eNB and EPC running the OAI stacks. The MEC setup runs Ubuntu 20.04 LTS on an Intel i7-10750H quad-core CPU at 2.6 GHz with 32 GiB of RAM.

7.2 Latency Evaluation and Analysis - Hardware POC

To understand the overheads and network latency of the proposed approach, it is better to split the same as shown in Fig. 6. We can split the overhead into two components: One from the processing of data on the UE and communication to/from the modem, and the second is the network round trip delay of the LTE infrastructure. Expanding further, during the sending phase, we have overhead to request the GPS/GNSS modem for GPS data. Then we have the processing delay in building a BSM packet, computing, and appending HMAC. Finally, we send the combined packet to the modem via UART at 3Mbit/s baud and wait for the modem to provide an acknowledgment. The overheads during the receive phase are the same, except that they occur in the inverse order of operation and do not include GPS query. The network round trip delay varies depending on the network topology discussed in earlier sections.

Table 2. Roundtrip latency for considered network topologies. We provide the mean and 95% CI for values collected for 5000 Send-Receive Cycles * Ideal Case: eNB-EPC negligible channel delay

No. BSMs sent as response	Latency for server on Internet [ms]	Latency for server at the end of EPC* [ms]	Latency for server at eNB [ms]
1	61.410 [60.391–62.429]	29.139 [29.039–29.238]	28.576 [28.369–28.783]
5	77.039 [75.514–78.564]	26.974 [26.773–27.175]	27.900 [27.783–28.018]
10	84.415 [81.922–86.908]	40.085 [40.577–41.132]	37.015 [37.015–37.576]

We present the results in Tables 1 and 2 It is clearly evident that the COTS UE incurs minimal overhead for processing and exchanging data with the modem. We have a total overhead of 7.9388 ms when a single BSM/UE is sent as response from the server for every BSM. The average increases to 11.3938 ms for 10 BSM/UE as response from the server to every BSM. Therefore, even for 10 BSMs, the overall overhead is about 10% of a single core utilization on a low end open-source UE processor running at 1.152 GHz. We can expect even lesser overhead for processing on devices with proprietary hardware running at higher clock speeds. Additionally, the latency of communicating with the LTE

modem can be further reduced by increasing the baud rate or choosing a parallel communication protocol if the hardware permits.

The network round-trip times (RTT) vary depending on the network topology. In the first scenario with the server on the internet, we see an average round trip delay of 61–84 ms depending on the number of BSMs that are sent in response to the UEs. These latency values could enable CV2X applications with latency constraints of 100ms but not guarantee the same. In the case of server at end of EPC and end of eNB, we obtain even lower RTT since the server is much closer to the UEs. However, note that for the topologies of the server at the end of EPC and end of eNB, the network RTT represents a lower bound (as both the cases use a private LTE setup). Also, in the real world, the EPC would not be as close to the eNB, and a single EPC services numerous eNBs. Thus, latency values would be higher than the values obtained above; however, placing servers with consideration for geographical proximity could provide satisfactory latency [13]. Additionally, the setup for the server at eNB with USRP antenna inconsistencies has very few UEs to measure any effects of Radio access resource contention and collisions. The measurements above hold if enough bandwidth is provided to the CV2X UEs. Therefore, in the worst-case scenario, when there are 100s of UEs trying to compete for resources from a single eNB, the latency exponentially increases, though mitigation may be possible [32]. Therefore, in the next Section, we try to evaluate and empirically determine the number of UEs that can meet CV2X latency constraints for a single eNB considering available bandwidth.

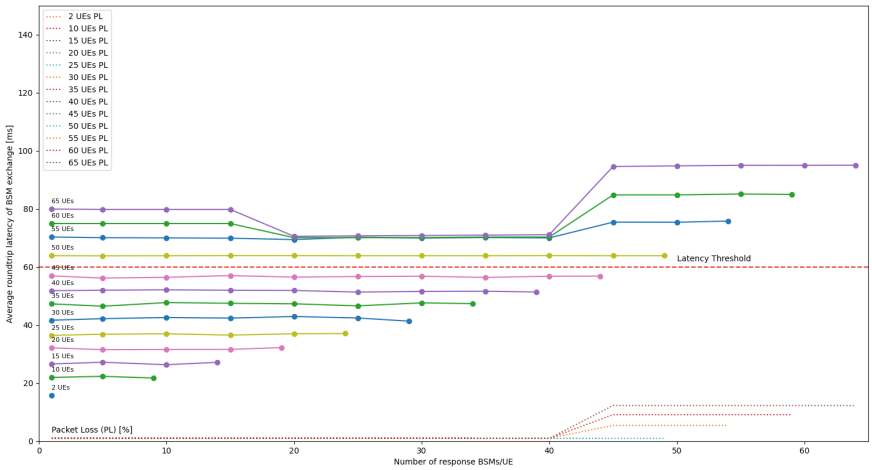


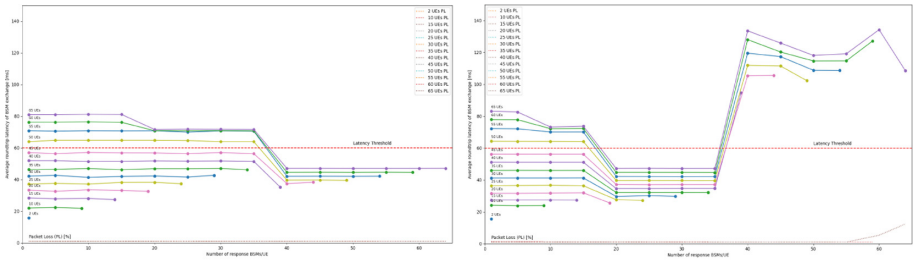
Fig. 7. Average latency and packet loss for 100 RBs (in increasing number of UEs starting from 2 and then in multiples of 5)

7.3 Simulation Evaluation

As discussed earlier, the primary bottleneck occurs in the radio access network of the LTE. Considering the most feasible topology for our use case(CV2X server as MEC at eNB), we modify the ns3 simulator(v3.26) [33] with MEC support [34] for our CV2X application. We wish to determine the possible number of UEs that a single eNB can service to measure the feasibility of our approach. For simplicity, we currently do not consider external traffic other than the CV2X application. We consider such robust realistic situations for the future.

Simulation Model and Configuration. For a given value of resource blocks, we consider an increasing number of UEs(n) from 2 to 65. Starting from an ideal 1 response BSM per UE to the worst(each UE sending a BSM to each other), we have response BSMs/UE(r_n) varying from 1 to $n-1$ for each n. Using the standard LTE configurations, we use the resource block values 25, 50, 100 for 5, 10 and 20 MHz(bandwidths of individual LTE bands) respectively. The smaller resource block values could be looked at as dedicated resource allocations within the larger 100RB scenario to understand the behavior if some bandwidth is explicitly reserved for CV2X. We modify the native UDP echo server of ns3 to mimic our CV2X application for measuring latency with varying BSMs/UE. The UDP echo client on UEs send a BSM every 100 ms, to which the UDP echo server responds with varying number of BSMs/UE. Additionally, to add a mobility scenario, we move the UE's randomly around a central eNB at 20 m/s using the ns3's 2D random direction mobility model. The eNB is configured to use an isometric 2×2 multiple-input multiple-output (MIMO) antenna. We utilize default ns3 configurations for the rest of the network parameters. Since, we are more concerned about the latency at the application level, we do not consider lower level protocol layers, but use the ns3's Flow monitor module to get the higher level network layer latency and packet loss.

Simulation Analysis. Even though the periodicity requirement of the application is 100 ms, we set a safe cutoff threshold of 60 ms for our analysis. Recall that our UE proof-of-concept does not consider any latency/overhead for actions from the vehicle. Also, we only measure the hardware overhead for response upto 10 BSMs/UE. The communication overhead of data between the UE processor and the modem would considerably increase with size of the response from server. Therefore, we consider a safe latency threshold of 60ms to determine the feasible number of UEs the eNB can support. As seen from the Fig. 7, for 100 RBs, a 2×2 MIMO antenna setup on the eNB can service up to 45–50 UEs under our latency threshold. We also only see packet loss when the number of UEs are above 50. In the case of 25 RBs and 50 RBs, we do see a local maxima for reduction in latency, but the latency overshoots our latency threshold bar for other arbitrary values of BSMs per UE. Therefore, from Fig. 8a and Fig. 8b, a safe value of number of UE that can be supported are 35 and 45 for 25 RBs and 50 RBs respectively.



(a) (b)

Fig. 8. Average latency and packet loss for (a) 25 RBs (b) 50 RBs (in increasing number of UEs starting from 2 and then in multiples of 5)

Given, that the commercial networks utilize more robust software and hardware like 4×4 MIMO antennas, proprietary scheduling algorithms and dedicated resources, the number of feasible UEs that could be supported would be slightly higher than the presented values.

8 Future Work

As mentioned, though we try to analyze the various topologies and COTS smartphone capabilities, there is still scope of further investigation. Considering the network, we assume no external traffic or influences for the simulation. Additionally, the hardware evaluation with a single UE does not provide a holistic discussion of the approach. So, we intend to further investigate the possibility of using LTE via enhanced simulations and a larger scale hardware test bed. Subsequently, our secure CV2X application on the UE is abstracted from the underlying cellular modem. So, we would like to explore the possibility of analysing our approach on the 5G or device-to-device (D2D) frameworks. A more interesting tangent we wish to also explore is to setup the above approach to connect with an actual vehicle over the OBD-II port, enabling us to analyze and investigate our approach for vehicle compatibility and driving applications. This would enable the end-to-end discussion of our whole idea.

9 Conclusion

In this paper we investigated a secure framework to enable CV2X using commercial of-the-shelf smartphones by leveraging device hardware security extensions. In particular, we look at using the smartphones as CV2X radios without degrading their performance for regular operation. Further, we explored possible threats to using such a framework on smartphones and provide mitigation approaches to thwart the same. Additionally, we also consider the whole LTE network in our framework and investigate the possible solutions of using a dedicated server to act as an intermediary between the UEs. Simulation results, backed by hardware measurements indicate that CV2X can be securely implemented using COTS smartphones.

References

1. Shavit, M., Gryc, A., Miucic, R.: Firmware update over the air (FOTA) for automotive industry. (SAE Technical Paper) (2007)
2. Nie, S., Liu, L., Du, Y.: Free-fall: hacking tesla from wireless to can bus. Briefing, Black Hat USA **25**, 1–16 (2017)
3. Nie, S., Liu, L., Du, Y., Zhang, W.: Over-the-air: How we remotely compromised the gateway, BCM, and autopilot ECUs of Tesla cars. Briefing, Black Hat USA (2018)
4. Arm, L.: ARM Security Technology-Building a Secure System using TrustZone Technology. (PRD-GENC-C. ARM Ltd., Apr. (cit. on p.) (2009)
5. McCord, K.: Automotive Diagnostic Systems: Understanding OBD I and OBD II. (CarTech Inc.) (2011)
6. Pinto, S., Gomes, T., Pereira, J., Cabral, J., Tavares, A.: IIoTEED: an enhanced, trusted execution environment for industrial IoT edge devices. IEEE Internet Comput. **21**, 40–47 (2017)
7. Ahmad, Z., Francis, L., Ahmed, T., Lobodzinski, C., Audsin, D., Jiang, P.: Enhancing the security of mobile applications by using TEE and (U) SIM. In: 2013 IEEE 10th International Conference on Ubiquitous Intelligence and Computing and 2013 IEEE 10th International Conference on Autonomic And Trusted Computing, pp. 575–582 (2013)
8. Wang, Y., Gao, W., Hei, X., Mungwarama, I., Ren, J.: Independent credible: secure communication architecture of android devices based on TrustZone. In: 2020 International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics), pp. 85–92 (2020)
9. Ahlawat, A., Du, W.: TruzCall: secure VoIP calling on android using ARM TrustZone. In: 2020 Sixth International Conference on Mobile and Secure Services (MobiSecServ), pp. 1–12 (2020)
10. Liu, S., Xiang, W., Punithan, M.: An empirical study on performance of DSRC and LTE-4G for vehicular communications. In: 2018 IEEE 88th Vehicular Technology Conference (VTC-Fall), pp. 1–5 (2018)
11. Hassebo, A., Obaidat, M., Ali, M.: Commercial 4G LTE cellular networks for supporting emerging IoT applications. In: 2018 Advances in Science and Engineering Technology International Conferences (ASET), pp. 1–6 (2018)
12. Amjad, Z., Sikora, A., Hilt, B., Lauffenburger, J.: Low latency V2X applications and network requirements: performance evaluation. In: 2018 IEEE Intelligent Vehicles Symposium (IV), pp. 220–225 (2018)
13. Pyykönen, P., Lumiaho, A., Kutila, M., Scholliers, J., Kakes, G.: V2X-supported automated driving in modern 4G networks. In: 2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP), pp. 271–275 (2020)
14. ARM Arm Security Technology Building a Secure System using TrustZone Technology. ARMDeveloper. https://developer.arm.com/documentation/PRD29-GENC-009492/c?lang=en
15. Linaro Open Portable Trusted Execution Environment. OPTEE Documentation. https://optee.readthedocs.io/en/latest/index.html
16. GlobalPlatorm Introduction to trusted execution environments (2018). https://globalplatform.org/resource-publication/introduction-to-trusted-execution-environments/. Accessed 30 Mar 2022

17. Paradisi, A., Yacoub, M.D., Figueiredo, F.L., Tronco, T.R. (eds.): Long Term Evolution. TIT, Springer, Cham (2016). https://doi.org/10.1007/978-3-319-23823-4
18. European Telecommunications Standards Institute. Service requirements for V2X services ETSI TS 122 185 V14.3.0. (2017)
19. Society of Automotive Engineers. V2X Communications Message Set Dictionary J2735_202007 (2020)
20. Bajaj, R., Ranaweera, S., Agrawal, D.: GPS: location-tracking technology. Computer **35**, 92–94 (2002)
21. Van Diggelen, F.: A-gps: Assisted gps, gnss, and sbas. Artech house (2009)
22. Nardini, G., Virdis, A., Campolo, C., Molinaro, A., Stea, G.: Cellular-V2X communications for platooning: design and evaluation. Sensors **18**, 1527 (2018)
23. Miao, L., Virtusio, J., Hua, K.: Pc5-based cellular-v2x evolution and deployment. Sensors **21**, 843 (2021)
24. Krawczyk, H., Bellare, M., Canetti, R.: HMAC: keyed-hashing for message authentication. (RFc 2104) (1997)
25. Dobraunig, C., Eichlseder, M., Mendel, F., Schläffer, M.: Ascon v1.2: lightweight authenticated encryption and hashing. J. Cryptol. **34**, 1–42 (2021)
26. Camarillo, G., Garcia-Martin, M.: The 3G IP Multimedia Subsystem (IMS): Merging the Internet and the Cellular Worlds. John Wiley & Sons, Hoboken (2007)
27. Li, C., et al.: Mobile edge computing platform deployment in 4G LTE networks: a middlebox approach. In: USENIX Workshop on Hot Topics in Edge Computing (HotEdge 2018) (2018)
28. Giust, F., et al.: MEC deployments in 4G and evolution towards 5G. ETSI White Paper **24**, 1–24 (2018)
29. Pine64 Pinephone. https://www.pine64.org/pinephone/
30. Oehler, M., Glenn, R.: HMAC-MD5 IP authentication with replay prevention (1997)
31. Lentz, M., Sen, R., Druschel, P., Bhattacharjee, B.: Secloak: arm trustzone-based mobile peripheral control. In: Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services, pp. 1–13 (2018)
32. Amjad, Z., Sikora, A., Lauffenburger, J., Hilt, B.: Latency reduction in narrowband 4G lte networks. In: 2018 15th International Symposium on Wireless Communication Systems (ISWCS), pp. 1–5 (2018)
33. Riley, G., Henderson, T.: The ns-3 network simulator. In: Modeling and Tools for Network Simulation, pp. 15–34 (2010)
34. Nin, J.: NS3-MEC: MEC model for NS-3. GitHub. https://github.com/mmajanen/ns3-MEC