Observations and Recommendations on ***Connected Vehicle Security***

cloud security alliance®

The permanent and official location for *Cloud Security Alliance Internet of Things Working group* is **https://cloudsecurityalliance.org/group/internet-of-things/**.

Acknowledgments

Lead Authors

Brian Russell
Aaron Guzman
Paul Lanois
Drew Van Duren

Contributors

Joe Kish
Ashton Mozano
Arvind Tiwari
Shahid Sharif
Ron F. del Rosario
Alexandre Caramelo Pinto
Meena Krishnan
Velmurugan Manoharan
K S Abhiraj
Abel Sussman
Rohan Patil
Syed Mohamed A
John Yeoh
Larry Hughes

Table of Contents

1. Introduction

Looking at the Bigger Picture

2. Background and the Evolution of Vehicle Connectivity

The CAN Bus

Diagnostic Tools

Infotainment Connectivity

Door Locks - Remote Keyless Entry

Connected Vehicles (V2V, V2I, V2X)

Vehicles as Components of the Internet of Things

Continued Vehicular Integration With Mobile Applications

Integration With Smart Homes, Smart Roads, Smart Cities, and Smart Businesses

The Evolution of Mobility as a Service

The Impact of Cloud Connectivity

3. Areas of Concern to Connected Vehicles

An Example

4. Recommendations for Securing the Connected Vehicle Environment

Security by Design Processes and Standards

Cryptography: Key Management, Crypto Modules, Libraries and Protocols

Securing the Vehicle Platforms

Strong Segmentation / Boundary Defense

Default Secure Configurations

Secure Update Processes

Interface Filtering

Secure Protocol Implementations

Aftermarket Protections

API Security Guidelines

Access Control

Mobile Application Security

Data Integrity

Privacy Protections

Securing the Traffic Infrastructure

Device Management

Monitoring

Auditing and Logging

Event Correlation

Device and Software Inventory

Malware Defense

Wireless Access Controls

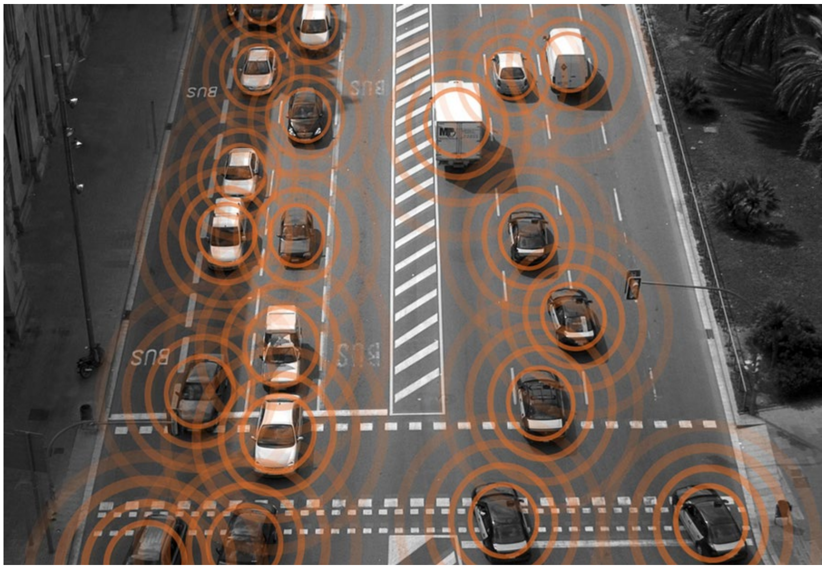
- Redundancy Controls
- Boundary Protections
- Policies and Procedures
- SCMS Security as the Foundation for Connected Vehicles
- Recommendations on Handling Gaps
- Enabling Trust Between Cryptographic Domains
- Security Design Assurance
- Indicators of Compromise (IoC)
- Standardized Methods For Disclosure
- Standards for Securing Mobile Applications in the V2X Context
- Focused Coordination Between Security, Technology and Automotive Communities
- Continued Research and Development

5. Conclusion

1. Introduction

The introduction of Connected Vehicles (CVs) has been discussed for many years. Pilot implementations currently underway are evaluating CV operations in realistic municipal environments. CVs are beginning to operate in complex environments composed of both legacy and modernized traffic infrastructure. Security systems, tools and guidance are needed to aid in protecting CVs and the supporting infrastructure. Recent headlines, such as the **infamous Jeep hack** and **Tesla hack**, demonstrate just how critical it is to ensure the security of CVs. Hackers' ability to hijack control of CVs is now proven very real.

The authorities have taken notice of the risk. On March 17, 2016, **a joint public service announcement** by the FBI, Department of Transportation (DoT) and the National Highway Traffic and Safety Administration (NHTSA), warned of the threat of Internet-based attacks on cars and trucks. While the FBI noted that the vulnerabilities identified so far have been addressed, it is important that CV consumers and manufacturers be continually aware of the inevitability of future vulnerabilities. Going forward, we can probably expect to see the same level of regulation as for critical infrastructures. Some lawmakers, such as the **state of Michigan**, are already considering laying out the foundations of legislation and sentencing guidelines for the crime of car hacking.

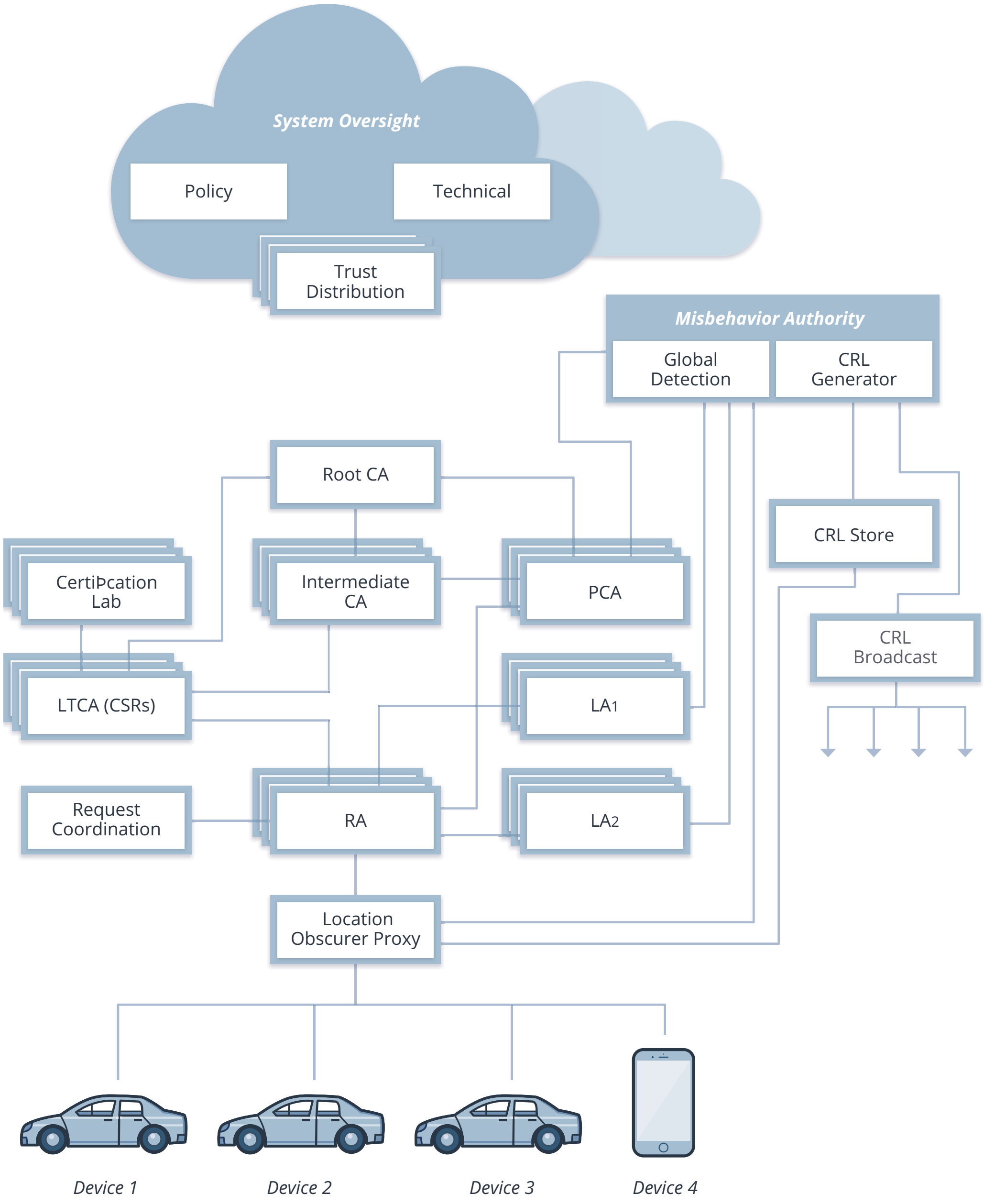


The Department of Transportation Federal Highway Administration (FHWA) has developed a **Connected Vehicle Reference Implementation Architecture (CVRIA)**. CVRIA defines four architectural views — Enterprise, Functional, Physical, and Communications — with security integrated throughout each view. Review of the CVRIA provides a solid understanding of the applications, connectivity and components associated with the overarching CV ecosystem.

One of the primary capabilities enabled by the CV architecture is the ability of vehicles to communicate with proximal vehicles ("V2V"), with infrastructure ("V2I"), and with applications ("V2X"). Communication is accomplished through a wireless messaging protocol known as Dedicated Short Range Communication (**DSRC**). DSRC messages are digitally signed to guard against tampering and spoofing.

The digital signatures are enabled by certificates provisioned to each component from an infrastructure known as the Security Credential Management System (**SCMS**). The SCMS is a proof-of-concept Public Key Infrastructure (PKI) tailored to provision certificates to vehicles and infrastructure. SCMS implements robust privacy controls that guard against both message manipulation and casual tracking of vehicles (and by extension, their owners) by unauthorized parties (the “outsider threat”). It also protects against rogue parties that operate components of the SCMS itself (the “insider threat”). The SCMS employs components such as Location Obscurer Proxies (LOPs) that shield vehicle identities from PKI components and vehicle operators. Vehicles employ rotating certificates taken from a pool, and then use them to digitally sign messages. The SCMS design is depicted in **Figure 1** (**reference**)

Figure 1



Work is also being done to support secure vehicle operations. In July 2016, the Auto Information Sharing and Analysis Center (ISAC) published **a report** titled “Automobile Security Best Practices.” The report provides a well-thought-out set of recommendations for securing vehicle operation platforms.

Other industry work focuses on helping Original Equipment Manufacturers (OEMs) and suppliers understand the threats associated with vehicles. Industry groups such as I AM The Cavalry have released guidance to this effect, for example, the **Five Star Automotive Cyber Safety Program**.

Looking at the Bigger Picture

When we consider the future of automobile technology, it is important to take a “big picture” view of the various aspects of vehicles and infrastructure components to better understand their interrelationships, dependencies and threats to the traffic ecosystem. In the future:

- CVs will operate while communicating with both legacy and modernized traffic infrastructures and their sensors.
- Traffic Management applications and vehicles will interact with cloud services using a mixed set of transport protocols (RF/ WiFi, etc).
- OEM and 3rd party applications will be installed on vehicle platforms and traffic infrastructure components to provide enhanced capabilities.
- CVs will integrate with the IoT ecosystem to support vehicle integration with smart homes and smart businesses.

As in other industries, innovation will abound as methods and capacities for connectivity rise. We anticipate full integration of CVs with the IoT, which presents all new security challenges.

Next we analyze the evolution of vehicle connectivity towards fully connected and autonomous systems. We then provide recommendations for enterprise-wide security controls to safeguard the driving public. Finally, we evaluate the security gaps that need attention. Our intent is to provide a comprehensive perspective on vehicle security design, which must be flexible enough to adapt to future challenges, and be cognizant of unanticipated threats that future disruptive technologies may bring.

2. Background and the Evolution of Vehicle Connectivity

Automobile connectivity today is evolving on a number of fronts. Platforms designed in the pre-connected era are now being connected in multiple ways. This has led to the ability of security researchers to gain access to sensitive vehicle functions in order to it perform activities not intended by the driver. Sensitive functions can be compromised via direct access (e.g., USB and the On Board Diagnostic (OBD-II) port, including with 3rd party dongles), or remote access (e.g., infotainment systems/consoles, Bluetooth, WiFi, NFC and cellular).

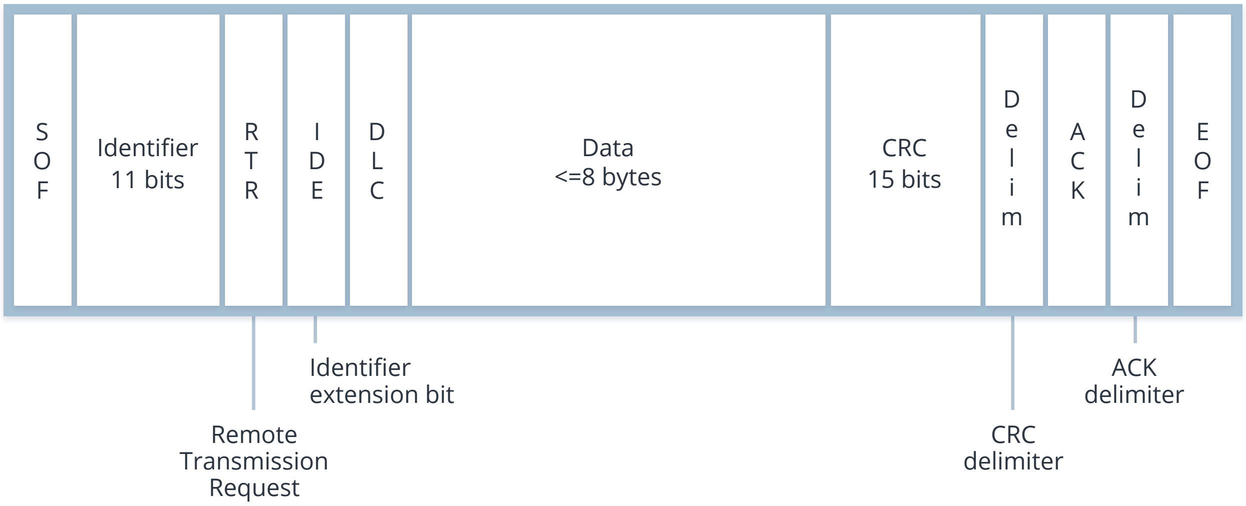
We begin this discussion of vehicle connectivity by describing the Controller Area Network (CAN) bus, which is a communication platform still used by most vehicles today.

The CAN Bus

One of the primary internal communication mechanism in vehicles is the CAN Bus. It is used to support communications between Electronic Control Units (ECUs) within the vehicle. The CAN bus was designed as a closed network, and therefore implements no security features such as message encryption or authentication. An unauthorized party that gains access to the bus can block legitimate messages and transmit illegitimate ones. Both actions can cause unwanted effects within the vehicle.

CAN frames include an Identifier, Control, Data field and a Cyclical Redundancy Check (CRC). Their simple structure is displayed in **Figure 2**.

Figure 2



Although some vehicles have only one CAN bus, others have multiple ones to segment safety-critical vs. noncritical functions. Automobile designers often find a need to interconnect multiple CAN busses, and do so using one-way gateways (built on micro controllers) to enforce separation. Some manufacturers interconnect CAN busses with infotainment systems, for example, to allow raising or lowering sound volume based on the traveling speed of the vehicle. It is important to secure even a simple use cases like this one.

Diagnostic Tools

Since 1996, automobile manufacturers have built in support for retrieval of diagnostic codes and other information from OBD-II ports. Although these ports typically provide read-only access to information needed to diagnose a problem with the vehicle, some manufacturers do allow commands to be sent over the CAN bus, though typically not for safety-critical functions.

Diagnostic access to the CAN bus is specified in ISO 14229-1:2013. Interestingly, the abstract for ISO 14229-1:2013 states:

> *ISO 14229-1:2013 does not apply to non-diagnostic message transmission on the vehicle's communication data link between two ECUs.* ***However, it does not restrict an in-vehicle on-board tester (client) implementation in an ECU in order to utilize the diagnostic services on the vehicle's communication data link to perform bidirectional diagnostic data exchange.***

There are tools available today that allow for malicious reading and manipulating of data through the CAN bus. The table below details a few of these tools.

CAN Hacking Tools (CHT)	A device with support for remote access (using bluetooth/GSM) that can interface with and control a vehicle
CANSpy	Auditing tool for cars. Supports packet interception. Requires physical vehicle access and connects via the OBD-II port.
CANtact	Open source CAN to USB hardware interface for analyzing vehicle CAN bus.
SocketCAN/CAN-utils	Set of open-source CAN Drivers & networking stack.

(To showcase the large quantities of additional tools available for interfacing to the CAN bus, reference **Ben Ferris' list on Peerlyst**.)

It is important to illustrate the real-world impacts of manipulating the CAN bus. Consider the research case in 2015 involving a dongle designed to be plugged into the OBD-II port, for use by insurance companies and trucking fleets. It was found that specially-crafted SMS messages could be sent to the dongle and passed on to the CAN bus of a Chevrolet Corvette. **Vulnerability Note VU # 209512** issued by US CERT states:

> *These devices are plugged into a vehicle's on-board diagnostics port (OBD-II), usually located under the wheel. The device itself contains a GPS receiver, cellular chip, and on board microprocessors which communicates with the vehicle's CAN bus to gather info (speed, braking, etc) The device then communicates via the cell network to the service provider to share data on the vehicle's operation…A remote, unauthenticated attacker may be able to execute arbitrary code on the device. In addition, a remote, unauthenticated attacker may be able to cause the vehicle damage or passengers injuries if the device is compromised.*

It is clear from this and other vehicle security research that enabling connectivity to the once-closed CAN bus can result in harmful effects if security engineering principles are not properly applied.

Infotainment Connectivity

One of the first stages of pervasive connectivity with vehicles has been through the infotainment system. Manufacturers include these systems to provide feature-rich services and content to their customers. Often these services are enabled through subscriptions. **Researchers have shown** that it is possible to gain access to an infotainment system and use that access as a jumping-off point to more sensitive vehicle functions.

A CV's Infotainment System is an easy target for exploitation due to the connectivity it requires from various web services (i.e; Weather, Traffic, Streaming Audio, etc.). All it takes is a single vulnerability (e.g., misconfigured server, unencrypted API call) for an attacker to exploit the rest of the system.

Door Locks - Remote Keyless Entry

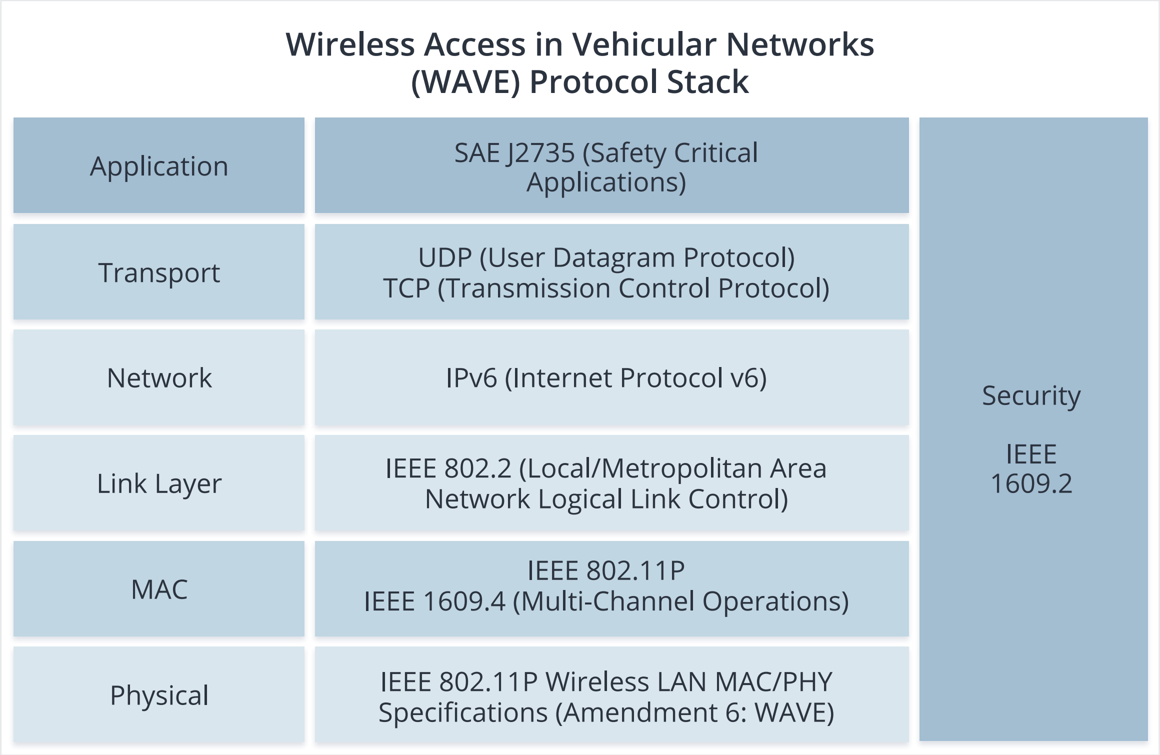
Door locks offer additional connectivity options by using protocols such as Bluetooth and NFC along with key fobs and even smart phone applications.

There have been recent media reports on the **vulnerabilities of the Remote Keyless Entry**, whereby a thief uses a device to "amplify" the signal generated by a keyless remote, or plants a device near the vehicle that intercepts the door-opening code for later playback when the owner of the vehicle is away.

Connected Vehicles (V2V, V2I, V2X)

The Wireless Access in Vehicular Networks (WAVE) technology stack defines various standards for vehicular communications. Figure 3 provides a view into those standards.

Figure 3



Wireless Access in Vehicular Networks (WAVE) Protocol Stack

Application	SAE J2735 (Safety Critical Applications)	Security IEEE 1609.2
Transport	UDP (User Datagram Protocol) TCP (Transmission Control Protocol)	
Network	IPv6 (Internet Protocol v6)	
Link Layer	IEEE 802.2 (Local/Metropolitan Area Network Logical Link Control)	
MAC	IEEE 802.11P IEEE 1609.4 (Multi-Channel Operations)	
Physical	IEEE 802.11P Wireless LAN MAC/PHY Specifications (Amendment 6: WAVE)	

CVs communicate with each other, and with infrastructure, using DSRC. DSRC supports rapid transmissions of messages between vehicles (V2V), infrastructure (V2I), and applications (V2X). Messages can be both safety-critical and non-safety-critical. For example, a Basic Safety Message (BSM), which is defined in SAE J2735, can be consumed by surrounding vehicles to make autonomous decisions, or to inform the driver of changes to the state of the traffic environment. BSMs include a diverse set of data elements that can be harvested by CVs to operate more safely and efficiently.

Vehicles broadcast a core set of data elements at a rate of 10x per second. These data elements include vehicle position, size, speed, heading acceleration and brake system status. Additional data elements are also available, as per the table below.

Examples of Data Elements Broadcasted by Vehicles		
GPS Position	Steering wheel angle	Stability control status
Latitude	Acceleration set	Brake boost applied
Longitude	Longitudinal acceleration	Auxiliary brake status

Elevation	Vertical acceleration	Vehicle Size
Positional accuracy	Taw rate	Vehicle width
Motion	Brake system status	Vehicle length
Transmission and speed	Brake applied status	Vehicle temperature
Transmission state	Brake status not available	Vehicle weight
Speed	Traction control state	Camera imaging
Heading	Anti-lock brake status	Radar imaging

**suggested but not limited to these examples*

A wide range of applications can be supported with V2V constructs. For example, vehicles can be warned of emergency breaking occurring many car lengths ahead, or of a disabled car blocking an unlit roadway at night.

Additional applications can be supported through Vehicle-to-Infrastructure (V2I) communication that incorporate connected roadside units (RSUs). These include but are not limited to environmental applications that provide motorists with warnings and notifications, safety applications that identify red-light or stop-sign violations as well as work-zone notifications. Vehicles can also consume broadcast messages that provide information on speed limits, signal phase and timing, and the presence of traffic conditions ahead.

RSUs provide the connectivity for infrastructure equipment. These devices provide DSRC communication on one interface and backhaul communications to Traffic Management Centers (TMCs) on another interface (i.e., via Ethernet connectivity to the traffic signal control cabinet). As with any infrastructure, RSUs must be managed. TMCs support the ability to manage them remotely, through mechanisms such as Secure Shell (SSH) or National Transportation Communications for ITS Protocol (NTCIP). NTCIP is a collection of protocols that includes Simple Network Management Protocol (SNMP) and the Simple Transportation Management Protocol (STMP). NTCIP allows for functions such as status reporting, control, and upload of configurations and download of event logs.

Mobile applications are also a large component of the CV ecosystem. Pedestrians may use apps loaded onto their smartphones or purpose-built dongles to communicate with infrastructure equipment (e.g., traffic lights) as well as vehicles. Pedestrian-to-vehicle communication (one instance of V2X) will support abilities that include detection of pedestrians as they enter crosswalks or cross at non-designated intersections. New innovative use cases will likely emerge as vehicles progressively communicate with smartphones.

A key take-away is that vehicles will quickly become reliant on messages received from other vehicles, infrastructure and mobile applications. It is therefore crucial to be able to trust that these messages will be delivered as expected, have not been tampered with, and have not been sent by unauthorized entities. The provenance of the data must be trustworthy, including data flows obtained from or processed through cloud-based applications.

Within the vehicular environment, V2V, V2I and V2X messaging is managed by a device known as On Board Equipment (OBE). OBEs integrate with the CAN bus to provide information such as vehicle speed and brake system status to participating entities. This bring us back full circle to needing to protect the internal components of a vehicle in order to maintain confidence that V2V, VSI and V2X messages are legitimate. One can imagine a situation where an attacker feeds false data to a compromised OBE in order to broadcast incorrect information.

Vehicles as Components of the Internet of Things

CVs ar coming online early in the age of the Internet of Things (IoT). We are already seeing the integration of vehicles with mobile applications and robust connectivity to the cloud. Vehicles can be considered complex IoT endpoints that are capable of establishing communications with any other IoT endpoints. This communication will typically occur via the cloud. As the IoT and CVs mature, we will notice:

- Continued vehicular integration with mobile applications
- Creation of Software-as-a-Service (SaaS) models that support vehicle diagnostics, management and maintenance
- New uses cases that integrate CVs with Smart Homes, Smart Cities and Smart Businesses
- An Evolution of Mobility-as-a-Service leading with the rise of autonomous vehicles

Continued Vehicular Integration With Mobile Applications

When considering vehicular integration with mobile applications, there are three primary examples:

1. Mobile applications that connect directly to a vehicle to perform some function (e.g., turn on air conditioner, start car)
2. Mobile applications that synchronize with a vehicle's infotainment system that reaches back to service providers
3. Mobile applications that identify a vehicle as a component of the larger ecosystem (e.g., a parking app) and potentially ties-in with service/subscription-based billing

A good example is the Starlink app from Subaru. This app enables door lock/unlock operations and access to the horn and lights. The app also serves as a gateway to entertainment services that can be piped to the vehicle's infotainment system.

Also consider infrastructure apps such as Park 4u and ParkMe from an integration perspective. Once a driver is able to locate a parking space in a given range of miles, V2I signals could be fed to a navigation app such as Waze that provides optimal routing to the space. Complex deconfliction algorithms would be needed to address cases where multiple vehicles are navigating to the same space. Such complexity can allow for disruption if the workflow is not secured.

Integration With Smart Homes, Smart Roads, Smart Cities, and Smart Businesses

Aftermarket developers will find ways to merge vehicle platforms with other IoT platforms. Within business, fleet management systems will consider vehicle platforms as IoT endpoints, providing connectivity to capture telemetry, identify maintenance needs and provide software updates. In the consumer realm, IoT product manufacturers will continue to look for ways to integrate cars with the connected home experience. Each new integration point reflects another potential attack vector into the vehicle, or into any automated processes that bind a vehicle or its state to other functions.

From a consumer perspective, opening up CV platforms to the cloud also introduces the ability to merge vehicle information with smart homes, using the cloud as transport. For example, a vehicle may download information reported by your smart refrigerator to provide a real-time alert to pick up a gallon of milk as you near a grocery store, or even place an order at your favorite take-out restaurant on the way home.

VCs can also help smart roads and smart cities understand traffic flows, road conditions, traffic conditions, etc., by being part of a crowd that provides real-time information to city administrators. The information can be used to inform a host of decisions from road repairs to traffic management.

We already have real-world examples of IoT capabilities built into vehicles. For example, there is an **automatic connected car adapter for NEST**, and some Mercedes vehicles can link directly to NEST. Product such as Amazon Echo and the Alexa app have tie-ins to vehicles that allow a consumer to:

- Query a vehicle for information related to the last trip taken
- Track the current location of a vehicle
- Set the environmental controls within a vehicle
- Start a vehicle remotely

The tie-in to a product like the Amazon Echo means that a consumer can, in some circumstances, start his or her car with a voice command to a smart home product. Locking down this command and control capability across all of the disparate components involved — smart home product, mobile app (x2), communications channel, vehicle platform — **should be a focused and coordinated effort on behalf of the automobile and tech industries, with an emphasis on open standards.**

Other interesting examples of potential CV / IoT integration include autonomous hotel check-ins, notifications when running late, dinner reservations, movie ticket ordering and payments for gas/parking. There are even APIs available to connect electric vehicles to the Smart Grid. Each of these is an autonomous service that is enabled through subscription or usage-based billing.

Privacy concerns should also not be dismissed. The ability to tie in location-tracking for a vehicle with smart home IoT products can provide a path for a determined attacker to stalk a victim.

The Evolution of Mobility as a Service

Mobility-as-a-Service has already evolved from the days of having to phone a taxi company to get a ride. With the introduction of mobile services such as Uber and Lyft, riders can simply use their mobile phones to schedule on-demand pickups. Today's driver also has a mobile phone that he or she uses to identify pickup location and navigate the course.

Soon, these companies will begin to leverage the capabilities of autonomous vehicle platforms. Uber is already beginning an experiment in Pittsburgh with Level 3 automation, where a driver in the "cockpit" is ready to take control, if needed, at any time. These experiments seem aimed at getting the public used to the experience of being picked up and driven by an autonomous car.

As we see the influx of autonomous car services throughout the world, additional points of integration will have been added to the vehicle platforms. Similar to the model of flying drones in Beyond Line of Sight configurations (e.g., setting waypoints via a Ground Control Station), we will see the ability to command a vehicle to start then drive to a set location. **Safeguarding the Command and Control channels in this configuration is important, and security researchers and technology organizations should be working together today to identify probable weaknesses, and to strategize on appropriate security controls.**

The Impact of Cloud Connectivity

The cloud removes geographic distance barriers and enables unique new features - many of which will not be considered until after the deployment of CV infrastructure and capabilities. For instance, researchers are working today to leverage the compute capabilities of a group of vehicles to create on-demand vehicular clouds that support a variety of use cases.

Further, the move towards 5G communications offers exciting new connectivity options for vehicles. There is potential for future cellular technologies such as 5G, when paired with cloud computing, to either replace or augment DSRC communications capabilities to support direct interaction with the cloud.

Examples of Systems, Solutions, and Services Linked in the Cloud*

Content providers	Repair shops
Service providers	Network operator
App developers	Fleet company
OEMs	Dealer
Energy companies	Insurance companies
Support center	Smart homes

**suggested but not limited to these examples*

IoT system architects can make use of the cloud to link disparate services, solutions and devices within and even across industries. These connections support data acquisition from myriad endpoints to deliver new value and enable fully threaded autonomous operations. There are exciting new SaaS-based applications and use cases being developed as more organizations understand the pieces of the puzzle and begin to brainstorm new ways of doing things. The same will occur in the CV domain, where companies such as Ericsson are already developing CV cloud solutions that can be deployed on standard cloud infrastructures.

An interesting example of cloud connectivity is **happening in Eindhoven**, where connected cars are participating in a pilot to report acceleration and location data via the cloud for analysis by the traffic authority.

Cloud connectivity enables interesting new capabilities for auto OEMs. The potential to collect and analyze data from vehicle operations can support analytics for product updates and future vehicle releases.

3. Areas of Concern to Connected Vehicles

When we consider risks to CVs and the ecosystem that supports them, we must take into account the various points of connection that have been discussed so far in this paper. Given that there are many forms of connectivity (DSRC, Cloud, WiFi, Cellular, Bluetooth, NFC, etc.), we should consider any vehicle that exposes these communication capabilities as being "connected" in some regard. **Adding additional connectivity means additional security risk.**

Within a system-of-systems (SoS) such as the CV ecosystem, there are many points of interconnectedness. A compromise of any one of these points potentially offers attackers the ability to move laterally throughout the entire ecosystem to compromise other points. For example, someone who has compromised the internal security of a vehicle's CAN bus may be able trick an OBE into sending fake data to other vehicles. Likewise, an attacker may be able to compromise equipment within a transportation center to send false traffic warnings or squelch the transmission of legitimate warnings. New connectivity options, such as cloud interfaces and wifi hotspots/access points from manufacturers, add to the complexity and therefore security of the overarching CV systems. They must be analysed and well understood using techniques such as threat modeling.

There are a number of motivations for bad actors to compromise CV components and technologies. These range from curious hackers attempting to demonstrate weaknesses, to malicious entities attempting to cause harm, on both small and large scales. Widespread outages of traffic systems have financial implications, cause confusion, and even grind society to a halt for a short time.

The table on the following page provides a view into example attacks that could be performed by adversaries, and their respective results. The sensitivity of the transportation industry is such that any of these actions, even when intended to merely show off or harass, could result in crashes and fatalities. Results can be magnified when large commercial CVs are involved.

Attack	Result
Exploit an unauthenticated API (e.g., remote features) used in a CV Exploit a vulnerability in a mobile application used to connect to a CV	Hijacks control of the CV's non-safety-critical operations
Reverse engineer a CVs firmware, and modify its MCU to bypass security controls or change functionality Locally or remotely exploit a vulnerability in a CV's Self-Driving Vehicle (SDV) code Infiltrate a CV's supply chain to install malware in a CV, and possibly inject it into an entire ecosystem Physically interface with a CV (e.g., via its USB port) to install malware Identify a method to circumvent a CV's safety features	Hijacks control of the CV's safety-critical operations
Monitor a CV's messaging traffic for an extended period of time Knowledge of vehicle location, regular routes taken, and duration of stay.	Unauthorized tracking of the CV
Infect a CV with ransomware to restrict/limit use Implement Denial of Service against traffic infrastructure	Unauthorized disablement of the CV
Exploit a CV's weak cryptographic features Exploit unchanged/weak passwords used somewhere in a CV's software	Unauthorized disablement of the CV
Spoof a CV's sensors (e.g., LIDAR, GPS)	Cause the CV to react unexpectedly
Steal or crash an autonomous CV	Complete loss or control of the CV
Coordinated attack on vehicles and infrastructure E.g., Denial of Service, Internet and wireless domains (jamming)	Suspension of critical and/or non-critical CV systems and controls.

Physical or remote input of self-replicating code targeting either CVs or infrastructure equipment resulting in the creation of botnets comprised of RSUs, etc that can be used to stage DDoS attacks	Suspension of critical and/or non-critical CV systems and controls.
Overload a circuit board used in a CV (e.g., use USBKill on a USB port)	Damage the CV's physical components

An Example

Let's explore the use case related to taking over non-safety-critical functions within vehicle. We've seen researchers compromise functionality within a Nissan Leaf based on unauthenticated remote transactions. The researchers explain that the smartphone app was sending the Leaf an anonymous HTTP GET request, with no user identification information. The only identifier passed was the Vehicle Identifier Number (VIN) of the target. There was no way to authenticate that the app was authorized to access that particular vehicle's control/data. By trial and error, the researchers were able to access other Leafs as well (**video**).

Researchers have also identified security vulnerabilities within vehicles. For example, in July 2015, **Wired magazine revealed** that two hackers were able to remotely disrupt the driving of a 2014 Jeep Cherokee driven by one of the magazine's writers, even turning off the car's transmission. Following this announcement, **1.4 million cars and trucks were recalled** by Fiat Chrysler in June 2014, followed by the **recall of an additional 8,000 Jeeps** (2015 Renegades) due to address remote hacking concerns. Fiat Chrysler is not the only car manufacturer having experienced hacks of their CV ecosystem: in August 2015, researchers managed **to take control of a Tesla Model S** and turned it off at low speed. Fortunately, Tesla quickly and remotely delivered software updates to fix the issue. In September 2016, **a remote attack** was discovered through the Tesla Bug Bounty Program.

As CV technology becomes more prevalent, it is likely to increasingly attract the attention of hackers and become a larger concern.

4. Recommendations for Securing the Connected Vehicle Environment

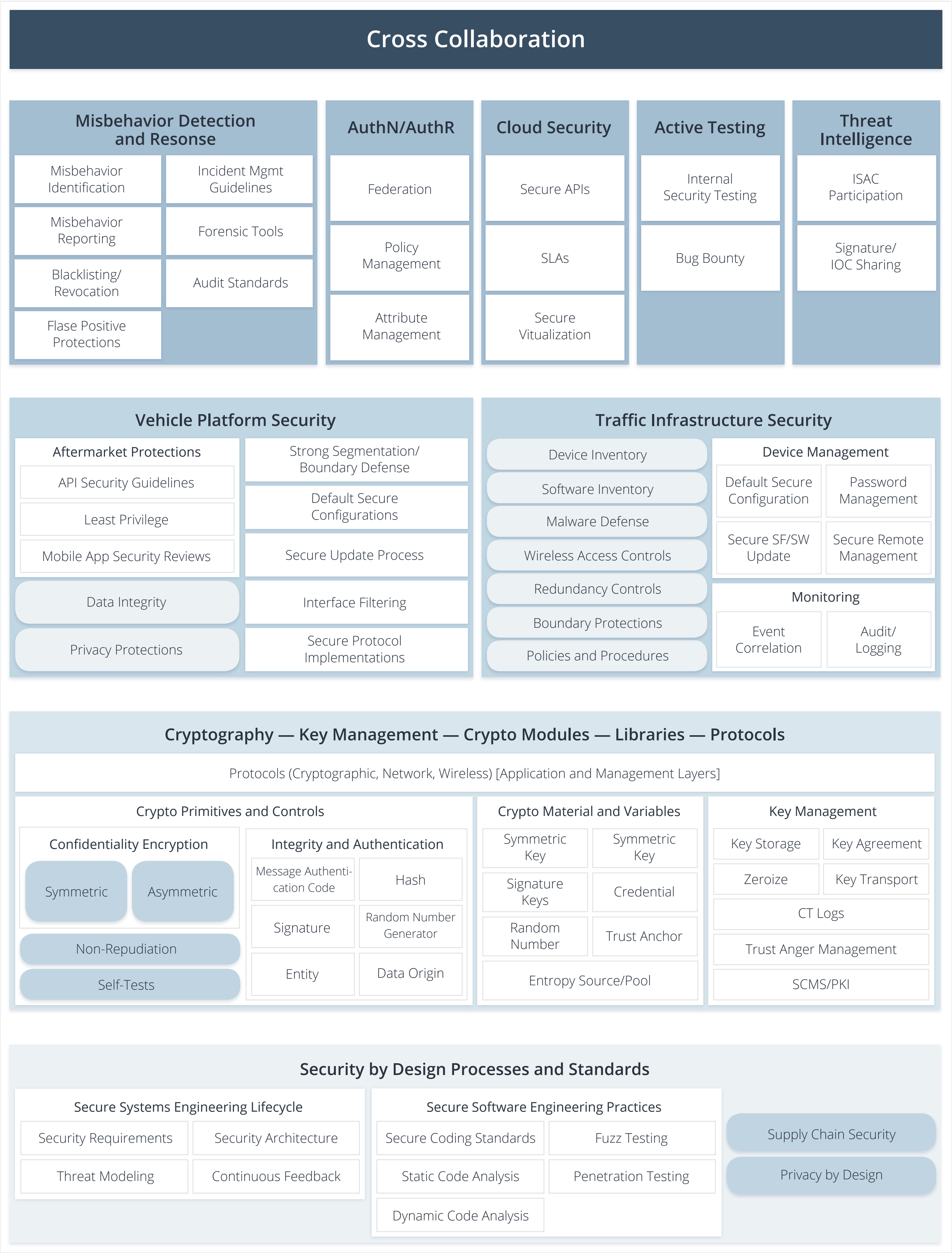
Securing the overall ecosystem of CVs that operate within an IoT cloud-connected environment requires coordinated planning and execution across multiple stakeholder communities (e.g., OEMs, suppliers, aftermarket developers, traffic infrastructure developers, traffic management center operators). This should be achieved through a methodical examination of the security posture across all transportation participants and components that interact with CVs and CV infrastructure. For example, has an automobile manufacturer accounted for the ramifications of inserting a 3rd party device into the OBD-II port of the vehicle? Can a compromised 3rd party device result in spoofed V2V messages being transmitted?

Comprehensive and discrete security guidance is needed to help guide organizations that are operating at various layers of the technology stack. This includes guidance on:

- Security of the CV platforms, which includes:
    - protection of control systems (CANBus, OBD-II port access, OBE)
    - protection of infotainment systems
    - protection against use of insecure third-party devices and applications
    - recommendations about software security engineering
    - hardware security controls (e.g., to protect MCUs)
    - interface security
    - configuration security
    - software maintenance
- Security of the ***smartphone applications*** that interact with (or are installed within) vehicles and CV communication systems
- Security of ***roadside equipment and infrastructure***, to include monitoring for security events, strong authentication/authorization for both local and remote access and cryptographic key management methods
- Security of ***messaging and communication protocols*** that devices rely upon for interaction both today (e.g., DSRC, cloud) and in the near future (e.g., 5G)
- Security of ***applications that support CV capabilities***, to include assurance of the “quality” of the software

Figure 4 provides a view into some of the controls that should be considered by various stakeholders across the traffic ecosystem. These controls are in no way comprehensive, but they do represent a starting point for organizations to begin investigating how to secure this dynamic, evolving ecosystem.

Figure 4



Cross Collaboration

Misbehavior Detection and Resonse

- Misbehavior Identification
- Incident Mgmt Guidelines
- Misbehavior Reporting
- Forensic Tools
- Blacklisting/ Revocation
- Audit Standards
- Flase Positive Protections

AuthN/AuthR

- Federation
- Policy Management
- Attribute Management

Cloud Security

- Secure APIs
- SLAs
- Secure Vitualization

Active Testing

- Internal Security Testing
- Bug Bounty

Threat Intelligence

- ISAC Participation
- Signature/ IOC Sharing

Vehicle Platform Security

Aftermarket Protections

- API Security Guidelines
- Least Privilege
- Mobile App Security Reviews

- Strong Segmentation/ Boundary Defense
- Default Secure Configurations
- Secure Update Process
- Data Integrity
- Interface Filtering
- Privacy Protections
- Secure Protocol Implementations

Traffic Infrastructure Security

- Device Inventory
- Software Inventory
- Malware Defense
- Wireless Access Controls
- Redundancy Controls
- Boundary Protections
- Policies and Procedures

Device Management

- Default Secure Configuration
- Password Management
- Secure SF/SW Update
- Secure Remote Management

Monitoring

- Event Correlation
- Audit/ Logging

Cryptography — Key Management — Crypto Modules — Libraries — Protocols

Protocols (Cryptographic, Network, Wireless) [Application and Management Layers]

Crypto Primitives and Controls

Confidentiality Encryption

- Symmetric
- Asymmetric

- Non-Repudiation
- Self-Tests

Integrity and Authentication

- Message Authentication Code
- Hash
- Signature
- Random Number Generator
- Entity
- Data Origin

Crypto Material and Variables

- Symmetric Key
- Symmetric Key
- Signature Keys
- Credential
- Random Number
- Trust Anchor
- Entropy Source/Pool

Key Management

- Key Storage
- Key Agreement
- Zeroize
- Key Transport
- CT Logs
- Trust Anger Management
- SCMS/PKI

Security by Design Processes and Standards

Secure Systems Engineering Lifecycle

- Security Requirements
- Security Architecture
- Threat Modeling
- Continuous Feedback

Secure Software Engineering Practices

- Secure Coding Standards
- Fuzz Testing
- Static Code Analysis
- Penetration Testing
- Dynamic Code Analysis

- Supply Chain Security
- Privacy by Design

Security by Design Processes and Standards

Processes form the foundation for security engineering. All stakeholders developing components within the CV ecosystem should implement ones based on a secure systems engineering lifecycle. The lifecycle should incorporate threat modeling activities that lay out the unique threats to a particular product. Developers should focus on answering the question "What if?" (for example, "What if an interface is used in ways that it was not intended to be used?"). The lifecycle should also include monitoring the effectiveness of and adherence to the lifecycle.

Development teams must do their part to mitigate vulnerabilities in software. Secure software engineering practices include defining secure coding standards, implementing static and dynamic code analysis, and bringing penetration testers on board to identify vulnerabilities prior to fielding. Understanding the ramifications of a potentially compromised supply chain is also important, especially when using open source libraries.

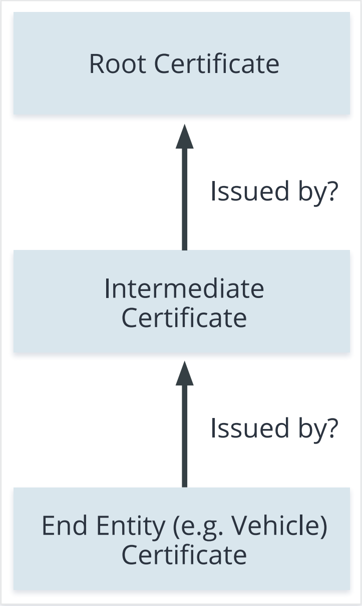
Cryptography: Key Management, Crypto Modules, Libraries and Protocols

Figure 5 emphasizes the need to properly implement cryptographic controls, as they serve to support many of the higher layer capabilities within the CV ecosystem. This is true both at a system level and individual device level. Incorrect / improper implementation of cryptographic tools and libraries opens wide avenues of attack.

Cryptographic protocols make use of cryptographic primitives and variables. Cryptography requires the availability of strong entropy sources which requires vetted random number generators (RNGs). Any component that makes use of crypto within the CV ecosystem should have access to a strong RNG. RNGs and entropy sources are evaluated as part of **NIST Federal Information Processing Standards** (FIPS) 140-2 validation.

Proper protections for cryptographic keys includes secure storage and zeroization capability. Other factors to consider include support for key agreement, key transport, and management of certificate trust anchors used in authenticated transactions.

When a certificate (e.g., 1609.2 or X.509) is used for a digital signature, the relying party must perform a root chain validation to verify that the certificate was issued by a trusted party.



Within the CV ecosystem, the Security Credential Management System (SCMS) plays a critical role in this regard. There is also significant use of traditional X.509 certificates for protocols such as Transport Layer Security (TLS).

Proper configuration should also be considered by all implementers of cryptographic protections. Recommendations from NIST or the European Union Agency for Network and Information Security (ENISA) guidelines on cryptographic protocols must be enforced. For example, when using TLS, ensure that only the desired cipher suites and TLS protocol versions are enabled, and deprecated deprecated protocols like TLS 1.0 are disabled. Enabling weaker cryptographic algorithms may allow a malicious peer to downgrade the security of the shared channel.

Additionally, the proliferation of many distributed cryptographic nodes suggests that migration to hardware roots of trust, such as Hardware Security Modules (HSMs), that are hardware-secured and tamper protective be encouraged throughout the transportation infrastructure.

Securing the Vehicle Platforms

There are unique security controls to implement at the platform level, whether that platform is a CV, an RSU or a TMC.

We focus here on securing the vehicle platforms themselves. With the addition of new communication capabilities and enhanced software features, CVs will be the focus of much research to identify novel methods for taking advantage of security flaws in the software or hardware of the the vehicle platform. These recommendations provide a solid starting point for securing the safety-critical functions of a vehicle even while operating in a highly exposed ecosystem.

Strong Segmentation / Boundary Defense

Today, many vehicles implement multiple CAN busses to separate safety-critical features (e.g., brakes, lane detection) from non-safety-critical features (e.g., door controls, lights). Separation should be maintained between CAN busses, and also between external components (e.g., infotainment system) and the CAN busses. Automobile manufacturers should incorporate separation/gateway security devices to enforce this separation, ensure their secure configuration, and perform comprehensive security testing to identify and fix any flaws.

Default Secure Configurations

Vehicle platforms include many software configurations. Operating systems are implemented within some vehicles. Some vehicles expose Wireless Access Points (WAPs) for consumer convenience. Configuration options should be understood and made as restrictive as possible.

Secure Update Processes

The need to update vehicle software is growing in urgency. There are many challenges that must be addressed, for example, the ability to rollback a software update that results in error. Capabilities like these must also apply to vehicle and infrastructure updates. There are a number of things to consider, including:

- Which field testing/certifications must an update pass before being made generally available
- What rollback functionality must be put in place
- How can updates be applied with minimal inconvenience to consumers
- How to prevent updates from be applied during vehicle operation
- Update processes must account for security throughout their entire lifecycle, from development of the update all the way through the distribution of the update

Manufacturers and OEMs that implement their own cryptographic update capabilities must also implement substantial security measures to prevent compromise, for example, protecting the root certificate chain and infrastructure.

Interface Filtering

Safeguarding interfaces is an important factor in keeping CVs secure. One attack vector used to remotely gain access to a CVs non-safety-critical features is unprotected interfaces. Security controls should include filtering of all interface traffic to do things like prevent malformed messages from being transmitted to the CV software. Defining and bounding the types of data that can be included in messages is also important.

Secure Protocol Implementations

Government organizations should provide guidance about secure implementations of communication technologies used by CVs. Many of today's CVs accomplish communication with things through bluetooth and Wi-Fi hotspots / access points. Overarching security architectures must protect communication points from attackers aiming to infiltrate CV components and ancillary components within infrastructure.

For example, the Bluetooth Low Energy (BLE) protocol offers flexible options for pairing devices, including a mode that requires mutual authentication. It is important that designs include secure pairing mechanisms such as this that do not include easy-to-guess PINs and passwords.

Aftermarket Protections

We have seen aftermarket devices that interface with VCs in various manners, including the OBD-II port. Some of these devices provide capabilities to consumers that likely were not considered at the time of vehicle manufacture. Manufacturers should responsibly envision add-on technology, and implement platform security controls to guard VCs' sensitive systems.

For example, 3rd party developers (e.g., insurance companies) have instituted dongles that interface with OBD-II ports. TomTom's GoLive interfaces over Bluetooth. These types of interfaces can provide unauthorized entry points into the overarching CV SoS. Following are just a few critical concepts that must be considered during design.

API Security Guidelines

Vehicle manufacturers should work together to provide security best practices for API interfaces. These should include recommendations for mobile application developers and others (e.g., cloud providers) that seek to interface with vehicle software.

Means for authenticating transactions, encrypting and integrity checking should all be clearly defined.

Access Control

The concept of "least privilege" should be utilized throughout the design. Applications that interface with vehicle systems should not be allowed to have more privileges that are absolutely essential for the functionality required. For example, applications should not be allowed to access data or transmit data from/to the vehicle unless required.

Mobile Application Security

Mobile applications are now being developed specifically for the automobile market. In the consumer IoT realm, the security of mobile apps that connect to devices is just as important as the security of the device itself. Attackers can potentially identify weaknesses in mobile applications and use them to exploit the vehicles.

Developers of CV mobile apps should implement certificate pinning to prevent man-in-the-middle (MITM) attacks over untrusted networks.

Data Integrity

Data integrity controls are relevant to various points within the vehicle. From a CV perspective, one of the primary security goals is maintaining the integrity of data that is communicated to other vehicles and to the surrounding infrastructure. Controls must be in place that ensure data (e.g., vehicle readings) cannot be spoofed or manipulated prior to entering the OBE responsible for transmitting the data via DSRC.

Privacy Protections

One of the primary goals of the SCMS is to enable privacy protections for consumers. Within the CV ecosystem, privacy is primarily focused on reducing the ability to illicitly track a vehicle's driver. The SCMS certificate handling design directly mitigates this through the use of a pool of certificates that digitally sign all message traffic originating from an OBE.

The SCMS architectural design also accounts for safeguarding against monitoring of driver habits and vehicle locations by using Pseudonym Certificate Authorities and specialized devices like Location Obscurer Proxies (LOPs).

Securing the Traffic Infrastructure

The overarching traffic infrastructure consists of many services, components and capabilities that support the goal of smooth traffic operations. For example, applications and equipment that provide GPS signals, RSUs that support toll payments and monitoring of traffic speeds, and radio equipment that implements DSRC communications between infrastructure and vehicles.

Securing the traffic infrastructure must be viewed from two distinct perspectives. First, it is important to ensure that infrastructure components cannot be used as launching-off points for malicious actors to gain access to vehicle platforms. Second, traffic infrastructure presents an enticing target for bad actors. For example, significant damage can be inflicted by changing traffic signals sequences unexpectedly or inappropriately.

The traffic infrastructure itself presents a viable target for attackers. The infrastructure is spread across wide-ranging geographies and includes control systems that were not designed for pervasive and persistent connectivity. Even so, new industry solutions are emerging that provide cloud-based capabilities, and this trend will continue well into the future.

In many ways, securing the traffic infrastructure is very similar to securing Industrial Control Systems (ICS). There are core business processes that must be secured when there are numerous pathways (e.g., actuators and sensors) into the systems that support those processes.

The traffic infrastructure within the V2I environment will consist of both modernized and legacy components. Vehicles will operate within this mixed environment, interfacing with many infrastructure components on a regular basis. Given that traffic infrastructure ensures the orderly flow of vehicles around a city, it is important that the infrastructure itself also be secured utilizing security best practices.

In addition to traditional traffic management features, drivers now also benefit from things like crowd-sourced alerts that redirect traffic around congested areas. Such navigation data can be sourced from a number of places, including traffic management centers, mobile applications and social media feeds.

Device Management

Devices within the traffic infrastructure are many and varied, and represent one of the critical categories of components that must be secured, including applications, data and interfaces. Devices providing functions such as traffic signal control, messaging, and communications are developed by a number of software and hardware vendors. RSUs often have on-board GPS connectivity. Many RSUs are accessible via a standard RS-232 serial interface to the device, and most do not provide any form of high assurance tamper resistance.

Securing the interface between the RSU and mobile device is critical. RSUs are typically paired with a supporting mobile application used to configure and query the devices. The mobile application may also support resetting the devices into factory setup mode, which could cause disrupt operations.

RSU design must consider the larger implication of operating traffic infrastructure in the emerging CV ecosystem. For example, RSUs typically contain only a small amount of data storage, and some will shut down when that storage becomes full. The simple practice of rotating log files can prevent a large impact on traffic instructure.

Some devices include integrated web servers that allow administrators to configure the devices by accessing particular Uniform Resource Locators (URLs). Where RSUs or other traffic management devices are configured this way, the configuration of the web servers themselves must be secure.

The table below contains a number of other recommended security controls for managing traffic infrastructure devices.

Security Control	Discussion
Default secure configurations	Developers of RSUs and other transportation devices must apply configurations that reflect security best practices. Traffic management implementers should verify that those best practices have been implemented.
Password and Secure Shell (SSH) key management	Default passwords must be changed on all devices upon first use. Private SSH keys should not be shared across devices.
Secure software updates	Developers should incorporate the ability to remotely update software, and provide patches in a timely manner. Publishers should digitally sign updates using a certificate trusted by the device. Traffic management implementers should practice secure update processes.

Secure remote management	Many RSUs implement NTCIP, a composite protocol for the management of transportation equipment. This protocol includes Simple Network Management Protocol (SNMP), Simple Fixed Message Protocol (SFMP) and Simple Transportation Management Protocol (STMP).
Secure OS	Automobile/IoT OS manufacturers should incorporate security controls into their designs. Industry best practices and crowd-sourced threat intelligence should be considered.

Monitoring

Auditing and Logging

Although the capability exists for online logging of traffic infrastructure, it is not always enabled. At a minimum, transportation system implementors should be able to identify suspicious activity within RSUs. For example, a device should send a notification of failed authentication attempts. This can be accomplished by leveraging NTCIP.

Other events that could be monitored include:

- Attempted physical access
- Attempted SSH access
- Attempted privilege escalation
- Attempted access to restricted file access
- Unauthorized GET/SET requests

Monitoring of log data at set intervals can also prove useful, although this may identify suspicious activity only well after the fact.

Event Correlation

Network event correlation, like monitoring, is not actively employed within many traffic management centers. The ability to tie together activities occurring in various locations provides enhanced situational awareness that can alert authorities of potential systematic attacks on the infrastructure. In addition, correlating events in near real time across an inventory of servers and devices that include RSUs and is geographically dispersed can be very useful in understanding suspicious activity. The development of this capability should be a focus of additional research.

Device and Software Inventory

One of the most important aspects of cyber security is keeping track of all authorized devices. Traffic management centers should keep an inventory of all RSUs and keep metadata about those devices up-to-date. Metadata can include:

- Software Version
- Firmware Version
- Location
- Responsible Party
- Trusted Applications installed

Malware Defense

A substantial lesson learned should be taken from the IoT botnet DDoS attacks that occurred in October 2016. These attacks show what can occur when self-replicating malware is let loose on large populations of homogeneous device types that have configuration vulnerabilities (e.g., shared passwords). Malware defense from the perspective of Road Side Units and Connected Cars is heavily geared towards establishing sufficient security controls to guard against malware infection in the first place. This must also be paired with mechanisms that allow for quick remediation of flawed code in the case a mass infection occurs. Monitoring of vehicle activity with malware detection techniques would thus help to detect anomalous behavior and filter out potential malicious events.

Wireless Access Controls

Traffic infrastructure will communicate using a diverse set of protocols. These include DSRC for V2I communications with vehicles. WiFi communications may also be enabled to communicate with peer infrastructure components and with gateways. Cellular communications from traffic infrastructure boxes will provide long-haul communications back to the Traffic Management Centers.

Organizations should implement identity-based protections (machine identity) to perform access controls prior to allowing devices to communicate on transportation networks. This should include all security-relevant endpoints to which the RSU connects.

The Cloud Security Alliance has spearheaded the creation of the Software Defined Perimeter (SDP). This should be explored for applicability towards transportation infrastructure systems to support authentication of devices prior to allowing attachment to any particular network.

Redundancy Controls

Traffic managers must take into account potentially malicious motivations when designing traffic infrastructure. For example, what happens if someone tries to perform a jamming attack on a large length of freeway infrastructure. From a CV perspective, there should be fallback mechanisms to deal with this denial of expected communications.   For example, The Traffic Management Centers could leverage the display and sensors within the vehicle itself to help mitigate the impact.

Boundary Protections

In some organizations, segmentation of IoT devices from other IT devices is mandated. Additionally, it is often prudent to segment critical computing resources from non-critical resources. Traffic infrastructure planners responsible for architecting smart traffic systems should take into consideration that equipment used in support of safety-critical communications may be better suited to being connected on segmented networks from informational, marketing, and other applications.

Policies and Procedures

Policies and procedures are important for ensuring that RSUs are consistently managed and operated in a secure manner. The policies and procedures should be detailed and enforced. Policies such as password lengths and role-based access controls should be considered by managers responsible for local/ remote maintenance. They should include removing test fixtures from devices before field deployment.

SCMS Security as the Foundation for Connected Vehicles

The Department of Transportation has designed and implemented a security system that will provide a cryptographic-based trust foundation for vehicles and RSUs within the CV ecosystem. The SCMS was designed to support enhanced privacy controls, and the detection and blacklisting of misbehaving devices (vehicles/RSUs). Misbehavior detection mechanisms are a discussion topic that requires ongoing thought about such things as what constitutes misbehavior, and how can it best be detected.

Certificates provisioned by the SCMS will enable integrity-protected and authenticated communications across safety-critical components in the V2V / V2I / V2X environments.   Certificates will enable privileged objects, such as emergency responder vehicles, to gain priority access through the traffic infrastructure.

Although some may question the viability of PKI for securing V2I communications, the SCMS provides a much needed security layer for protecting messaging and transactions. There are however some gaps in SCMS capabilities.

Recommendations on Handling Gaps

Enabling Trust Between Cryptographic Domains

Innovation will likely drive new types of interactions between vehicles, infrastructure and applications. Establishing levels of assurance for the interoperability of policy mapping (for matters such as registration and enrollment) will safeguard against proliferation of non-interoperable domains. The CV ecosystem should be treated as an SoS with policies and levels of assurance defined across the entire system. For example, establishing mutual authentication between certain components in the V2V / V2I / V2X environments will support trusted interactions between the components.

Security Design Assurance

To help promote software architectures and code that are resilient against attack, guidance and best practices are needed in the area of software assurance for vehicle software developers. (Even vehicle platforms without CV capabilities can require upwards to millions of lines of code.) This is a significant attack surface of the CV ecosystem.

The guidance and best practices should be extended to 3rd party app developers in the VC ecosystem.

Indicators of Compromise (IoC)

Significant work is needed in the area of Indicators of Compromise (IoC) for misbehavior detection. Open issues include:

- What IoCs are needed for CVs and for infrastructure?
- How are they being created, identified and shared?
- Who is responsible for misbehavior detection?
- Can APIs be made available that support new methods for communicating misbehavior? (e.g., if a manufacturer were to develop a new method of misbehavior detection)

These IoCs should also be promoted across communities, to allow for quick understanding of new attack techniques being employed in various regions.

Standardized Methods For Disclosure

Manufacturers and suppliers should embrace the security research community.

Fruits should include an easy means for disclosing vulnerabilities. In support of that, manufacturers and suppliers should provide:

- Contact email/phone numbers of the parties to disclose to
- Expected format and contents of the disclosure (i.e., description, risk, impact, proof)
- Service Level Agreement (SLA) for manufacturers and developer to respond to a disclosure

Standards for Securing Mobile Applications in the V2X Context

Mobile devices will play a large role in the V2X environment, and as such, mobile apps will make use of security credentials to enable authenticated and integrity-protected communications. Given the need to support many mobile device types, transportation application developers will be challenged to create applications on platforms that incorporate safeguarding trust stores and key material.

Mobile devices have secure storage capabilities, and those capabilities should be leveraged for safety-critical (e.g., pedestrian avoidance) applications. Two examples are Samsung Knox or Android devices fitted with ARM Trustzone (TEE). In Android, the Keystore can be used to store certificates and sensitive data, and in iOS, the Keychain.

It will be beneficial to define guidelines for levels of assurance that detail if/when SCMS certificates can be used to secure V2X communications, and for security controls on consumer mobile devices that are part of those communications.

Focused Coordination Between Security, Technology and Automotive Communities

This paper has discussed many diverse aspects of the V2V / V2I / V2X environment, and has outlined early considerations for various stakeholders in the CV ecosystem to secure their parts. The overall success related to securing the connected transportation industry will require all stakeholders to come together and share information about vulnerabilities, threats and attacks.

Consortiums such as the various Information Sharing and Analysis Centers (ISACs) are good starting points in this regard. Additional capabilities for sharing and cataloging IoCs would be a welcome advancement.

Continued Research and Development

Although the current SCMS proof-of-concept will provide a foundational set of cyber security capabilities for CVs, future transportation R&D should also be planned to identify novel methods of securing platforms. Some specific recommendations for future R&D include:

1. Evaluation of BlockChain technology as an alternative mechanism for applying integrity and authenticity protections for CVs. The SCMS provides certificates in support of safety-critical communications within the V2V / V2I / V2X environments. There are potential advantages to breaking out other systems that interact with vehicles and traffic infrastructure by levels of assurance. Blockchain, the underlying technology behind the cryptocurrency Bitcoin, is being innovated in the commercial sector to provide integrity and authentication protections for diverse transaction types such as **Guardtime's Keyless Signature Infrastructure (KSI)**. For non-safety critical communications, exploration of Blockchain technology as an augmentation to the SCMS should be considered.

2. Evaluation of Certificate Transparency (CT) logs for incorporation into the existing SCMS design. CT logs can help certificate authorities (CAs) determine if certificates have been mistakenly issued or maliciously used. Within the context of SCMS, an evaluation of the usefulness of CT logs for TLS certificates issued directly to SCMS infrastructure components should be undertaken (e.g., CAs, PCAs, LAs). This can provide an additional layer of security against rogue CAs intent on issuing SCMS certificates.
3. Continued investigation into security mechanisms to protect safety-critical messaging within vehicle platforms. This includes investigation into efficient mechanism for enabling integrity and authenticity protections within the CAN Bus and / or methods for effectively segmenting safety-critical communications from non safety-critical messaging.
4. Use of machine learning and AI for building safeguards and understanding the CAN messages for improving learning and proactively defending against cyber attacks.

5. Conclusion

CVs will operate in a complex ecosystem that connects vehicles between each other and the traffic infrastructure, and also opens up new forms of connectivity and relationships to cloud-based services and smart home, smart cities, etc. The ecosystem is becoming a very complex System-of-Systems in which security must be considered and accounted for at all levels.   The problem is made more complex by the fact that vehicles and infrastructure were not initially designed to be secure in the face of this emerging connectivity. For a safe and secure transportation system, the community must take a fresh look at the larger picture, and develop the policies, designs, and operations needed to incorporate security throughout the design. Use of disruptive technologies such as big data, machine learning and AI can help build a better, safer and more secure CV ecosystem.