Secure Crash Reporting in Vehicular Ad hoc Networks

Sumair Ur Rahman and Urs Hengartner
David R. Cheriton School of Computer Science
University of Waterloo
Waterloo ON, N2L 3G1, Canada
{surrahman,uhengart}@cs.uwaterloo.ca

March 26, 2007

Abstract

We present AutoCore, an automated crash reporting application that uses VANETs (Vehicular Ad hoc NETworks) to provide authenticated digital video and telemetry data. This data is recorded by vehicles either involved in or at the scene of a crash and can be used by investigators to reconstruct the events that lead up to the crash. To secure this application, we present a security infrastructure that extends the state of the art in VANET security. In particular, the contributions of this infrastructure include (a) the concept of Road-worthiness Certificates, (b) use of these certificates in a practical scheme for the distribution of cryptographic vehicle credentials issued by regional transportation authorities, (c) a decentralized scheme for conditionally anonymous, inter-vehicle communication, (d) efficient support for the roaming of vehicles between different transportation authority jurisdictions and (e) an evaluation of our security infrastructure using AutoCore.

1 Motivation & Introduction

Although emergency services usually respond to an automotive accident quickly, removing injured people within minutes of a crash, wreckage often remains on the scene for several hours as investigators take photographs and measurements to determine liability. For other commuters travelling on the same route, this often leads to lengthy delays.

To solve this problem, we propose a VANET (Vehicular Ad hoc NETwork) application, AutoCore, that automatically records video and telemetry data in a crash for use during an investigation. If investigators were provided with such data and could be assured of its authenticity, wreckage at crash sites could be removed quickly, reducing traffic delays. This system could also help determine liability in hit-and-run incidents.

An automated collision reporting application presents an interesting set of research challenges, including maintaining vehicle location and identity privacy, providing conditional anonymity for vehicles reporting collisions, protecting the system against various attacks and ensuring the authenticity of reported data.

Previous work that addresses VANET security and privacy has focused on identifying threats [15, 19], trusted inter-vehicle communication [7] and on the design of a security framework for VANETs [9, 18]. Several challenges, however, remain open: First, the task of distributing the cryptographic credentials used by vehicles to sign and authenticate outgoing messages has been largely ignored. Second, proposed key management schemes require a centralized database for the conditional anonymity of vehicles [18], introducing a single point of failure. Third, a concrete example of how these proposed techniques could be applied to protect a particular VANET application has been missing.

Contributions – Our contributions are (a) the AutoCore crash reporting application and a detailed analysis of the threats against it, (b) a security infrastructure to secure AutoCore that uses a decentralized scheme to provide conditionally anonymous inter-vehicle communication, (c) the concept of cryptographically-verifiable Road-worthiness Certificates issued to vehicles by authorized vehicle service centres, (d) the use of these certificates in a practical scheme for the distribution of cryptographic vehicle credentials issued by regional authorities via roadside access points, (e) efficient support for the roaming of vehicles between different regions and (f) a detailed security and cost analysis of our infrastructure.

Because our focus is on security and privacy, a detailed description of an automated crash reporting system is beyond the scope of this paper. Instead, we provide a brief overview of the system in section 3, after reviewing the state of the art in section 2. Section 4 analyzes our threat model, section 5 introduces our security infrastructure and section 6 describes how we use this security infrastructure to protect our crash reporting application. Section 7 analyzes our security infrastructure and section 8 discusses implementation issues. We review related work in section 9 and briefly discuss some directions for future work when concluding in section 10.

2 State of the Art

Automakers are working on pushing the safety envelope with proactive safety systems. These next-generation safety systems require vehicles to form cooperative groups, allowing them to exchange information and build awareness of their environments [16, 6, 22]. VANETs are thus a logical foundation for such safety systems.

To date, most industrial and academic research efforts in vehicular safety communications over VANETs have focused on the design of suitable MAC layer protocols, with the most prominent such protocol being WAVE (Wireless Access for the Vehicular Environment, often also referred to as DSRC or Dedicated Short Range Communications). Examples of vehicular safety applications studied so far include collision avoidance, cooperative driving and traffic optimization [16].

Designed as a short-to-medium range wireless protocol, WAVE is capable of supporting data rates of up to 27Mbps over a range of 1km and has been standardized as 802.11p [10] by the IEEE 1609 working group. In the US, the FCC has already allocated a 75MHz wide radio spectrum for WAVE at the 5.9Ghz band, with regulatory bodies in the EU and Japan pursuing similar initiatives.

Vehicular safety communications to support safety applications primarily consist of independent *geocast* [1] messages produced by vehicles and roadside infrastructure. Jiang et al. [11] group these safety messages as: *Routine Safety Messages* sent by vehicles and infrastructure on a regular basis, usually two or three times a second, and *Event Safety Messages* triggered by changes in vehicle behaviour such as sudden braking or infrastructure status, such as a vehicle running a traffic light. Messages generated by our collision reporting application fall into the latter category.

With VANETs largely being an emerging research field, little work has been done so far to address the security and privacy issues that arise from vehicles constantly sharing information about their movements and whereabouts with other vehicles and roadside infrastructure. One key challenge is the conditional anonymity of vehicles: a VANET security scheme should make it impossible for a global observer (e.g., law enforcement authorities, insurance companies, etc.) to track vehicles through the messages they transmit, while simultaneously allowing a vehicle to be reliably identified

[1] *Broadcast messages that contain information which is only relevant to recipients in a limited geographic region*

through these same messages when liability needs to be determined in the event of a crash and the ensuing investigation.

3 AutoCore

In this section, we describe AutoCore, an automated collision reporting application. We begin by listing the concerned entities, then provide a brief overview of the system and end by walking through a typical usage scenario.

3.1 Concerned Entities

In the design of AutoCore, we consider the needs and roles of the following five groups:

3.1.1 Drivers

Drivers likely do not want to have to actively submit crash evidence. AutoCore has to be able to do this automatically, without any driver involvement. The system should not compromise driver location or identity privacy. (See Zimmer [23] for a detailed discussion of privacy issues in VANETs.)

3.1.2 Governmental Transportation Authorities

Primarily concerned with maintaining roads and roadside infrastructure (traffic lights, stop signs, etc.) and issuing license plates, governmental transportation authorities (GTAs) want minimal involvement beyond their traditional role as a driver licensing and vehicle registration authority in the collision reporting process. We assume that GTAs are responsible for issuing vehicles and infrastructure with the cryptographic credentials used to secure inter-vehicle and vehicle-to-infrastructure communication (described in section 5).

3.1.3 Courts of Law

To protect the privacy of drivers, drivers and governments alike want a trusted legal entity, ideally a Court of Law, to have control over the release of the identities of vehicles involved in or at the scene of a collision. We assume that the required legal procedures are similar to those already in place to tap a phone line or view bank account records (e.g., a subpoena).

3.1.4 Law Enforcement Authorities

These authorities need to investigate and determine liability in the event of an accident. Given only the time and location of an incident, law enforcement authorities require easy access to authenticated evidence (video and vehicle telemetry). We assume that the requirement of a court order to obtain such evidence is not unrealistic.

3.1.5 Roadside Access Point Operators

These operators could either be GTAs, law enforcement authorities or commercial service providers. As part of our system model, we assume roadside access points (RAPs) will be deployed at various public locations accessible by vehicles (fuel stations, parking lots, etc.), with each RAP having

Internet connectivity and potentially serving more than one purpose (e.g., delivery of in-car entertainment content or electronic license plate renewal services). We also assume that communication between RAPs and vehicles will be governed by the local GTA, that is, all RAPs deployed within a GTA's jurisdiction will have to be inspected and certified by the GTA with cryptographically-verifiable credentials (section 5).

3.2 System Overview

The AutoCore system consists of control software, secure storage and a software interface to on-board positioning, imaging and telemetry sensors. To support the system, we assume the presence of a Tamper-Proof Device (TPD), a positioning system such as Differential GPS, cameras (now being fitted to luxury vehicles, such as the Lexus LS460 and the Mercedes-Benz S-Class to support automated parking and enhanced driver night vision, respectively) and a WAVE-like communication interface for inter-vehicle and vehicle-to-infrastructure communication. The latter is assumed to be reliable enough to ensure all vehicles within transmission range receive messages broadcast by AutoCore. Vehicles fitted with the system continuously record video and store this data with the corresponding vehicle position and telemetry data in the TPD. This storage takes the form of a ring buffer, where old data gets overwritten by more recent data.

We call vehicles that are directly involved in a collision *Primaries*, whereas those in the vicinity and within camera range are called *Witnesses*. All Primaries and Witnesses are equipped with the AutoCore system and are capable of sending the following two types of messages over their communication channels:

- *Collision Beacons* – These messages inform nearby vehicles that a collision event has occurred and include the current time and the source vehicle's current GPS coordinates.
- *Witness Beacons* – These messages inform nearby vehicles that the source vehicle is a witness to a collision event and include the current time and the source vehicle's current GPS coordinates.

In the event of a collision, Primaries transmit Collision Beacons, which trigger the generation of collision reports (by both Primaries and Witnesses, as described below) that contain video and vehicle telemetry data recorded during the collision for delivery to local law enforcement authorities. Since this data is recorded continuously and stored in a ring buffer, AutoCore has access to data from before it finishes processing a Collision Beacon, which ensures that processing and message propagation delays do not reduce the amount of useful data delivered in a report.

Collision reports are delivered to law enforcement authorities as follows: The authorities may either physically remove the TPD containing the report from Primaries to obtain the data they hold or use handheld devices to authenticate with the Primaries and obtain the data wirelessly through a WAVE link. Reports generated by Witnesses may either be obtained via handhelds or through delivery to RAPs that provide collision report delivery services. These RAPs would then forward all received reports to law enforcement authorities over the Internet.

RAPs providing report delivery services and handhelds issued to law enforcement authorities are termed Collision Report Collectors (CRCs). Both types of CRCs are certified by the local GTA and hold similar cryptographic information.

When a report is delivered to a CRC or to law enforcement authorities, the recipient issues the sender with a cryptographically verifiable receipt (explained in section 6.3). These receipts are used to form an audit trail that proves reports were in fact delivered to law enforcement authorities.

Timestamp
Location
Collision Beacons
Witness Beacons
Video Data
Host Vehicle Telemetry Data
Host Vehicle Anonymous Credential Certificate
Host Vehicle Signature

Figure 1: AutoCore collision report

Collision reports are stored in vehicles until they are delivered. (We assume sufficient storage space.)

Figure 1 shows the format of a collision report. A report also includes all received application beacons. The items shaded in dark gray are encrypted before delivery (see section 6.3). The report is signed and contains a certificate required for validation of this signature (see section 5.6).

When investigating a collision event, law enforcement authorities collate all received reports based on time and location of the event in question and use the provided video and telemetry evidence to reconstruct the collision and determine liability. We describe how reports are decrypted and verified in section 6.3.

3.3 Usage Scenario

We now walk through a usage scenario involving several vehicles fitted with the system travelling in opposite directions along a highway.

Vehicle A speeds up and attempts a late lane change, colliding with vehicle B. Both vehicles, equipped with AutoCore, are now termed Primaries. At the moment of impact, onboard sensors, such as those used to trigger airbags, inform AutoCore of a collision and both vehicles broadcast Collision Beacons. Vehicles C and D travelling behind the two Primaries as well as vehicle E traveling on the opposite side of the highway, but ahead of A and B, are equipped with AutoCore. These vehicles hear the Collision Beacons sent by one or both of the two Primaries. Vehicles C, D and E are now termed Witnesses. These Witnesses respond by broadcasting Witness Beacons, informing all involved vehicles (Primaries and Witnesses) of their presence, which then record these Witness Beacons as well as the original Collision Beacons as part of their collision reports.

After the collision, vehicle A is severely damaged. Law enforcement officials obtain its Collision Report by removing its TPD. The reports produced by vehicles B and C (the latter stops after the collision) are obtained by law enforcement officials using handheld CRCs. Vehicles D and E deliver their reports to roadside CRCs at fuel stations or parking lots automatically.

4 Threat Model

In this section, we identify threats to privacy and system security. Because our application operates on VANETs, we show how general threats identified against these networks [15, 19] manifest themselves in our application. We also discuss several application-specific threats, such as leaked collision reports.

4.1 Privacy Threats

Our privacy goal is to provide conditional anonymity. That is, vehicles broadcasting VANET application messages cannot be tracked or identified even in a pervasive deployment of VANET roadside infrastructure (e.g., smart traffic lights, dividers or signs), while accident investigators remain able to identify Primaries and Witnesses in case of an accident. We now present threats to this goal.

4.1.1 Vehicle Positioning

As illustrated in previous work [15], an attacker may attempt to track the movements of a vehicle by listening to the messages that it broadcasts. Although AutoCore broadcasts messages only when it encounters an emergency event (not very often), it would likely share cryptographic information (signing keys, electronic license plates, etc.) with other safety applications housed in the same TPD. These applications might have more frequent messaging requirements (up to 3 times a second [11, 18, 16]). Our threat model must thus accommodate frequent messaging.

4.1.2 Vehicle Identification

Although drivers already give up a significant amount of privacy by driving cars with license plates on them, what they expect is to be identified only when they are within visual range (i.e., while their license plates can be seen). The ability to track the movements of a vehicle without seeing it is considered a violation of privacy [23]. An example of this would be roadside infrastructure, such as traffic lights, being able to identify vehicles from the messages that these vehicles broadcast and reporting back to a central observer.

4.1.3 Leaked Collision Report Data

Unprotected collision reports may be viewed by unauthorized entities. An insurance company, for example, might look through these reports to single out drivers who have been at the scenes of multiple accidents and charge them higher premia. It would be considered a privacy violation if law enforcement authorities, other drivers, roadside access points or the GTA could freely view these reports.

4.2 Security Threats

In this section, we consider Denial of Service attacks that affect the availability of the AutoCore system, threats to the authenticity of the data generated by AutoCore, the ability of attackers to subvert AutoCore, attacks that would affect the integrity of the AutoCore software and threats against infrastructure used by vehicles equipped with AutoCore.

4.2.1 Denial of Service

Several types of Denial of Service (DoS) attacks are possible on AutoCore. An adversary could overwhelm vehicles by flooding them with false application beacons, rendering the communication channel, AutoCore and any other dependent applications useless. A similar type of DoS attack might target RAPs used by vehicles to refresh *Anonymous Credentials* (see section 5.6) or to deliver collision reports.

A second class of DoS attack might attempt to overwhelm a GTA's certificate refresh services (see section 5.6) by patching into the link between RAPs and the GTA, flooding it with garbage data. A similar attack might be launched against servers used by law enforcement authorities to collect collision reports from vehicles via RAPs (described in section 3.3).

In addition to the above, the most basic type of attack mentioned in almost all VANET security literature is signal jamming. In such an attack, an adversary would simply jam the communication channel used by vehicles and RAPs, rendering any dependent applications useless and preventing critical information from reaching vehicles and RAPs. We rely on earlier work [18] for addressing this particular DoS attack.

4.2.2 Message Suppression

In this type of attack, a driver either physically disables his inter-vehicle communication system or modifies the application to prevent it from either sending or responding to application beacons. The goal of such an attacker would be to prevent registration and insurance authorities from learning about collisions involving his vehicle or to avoid delivering collision reports to roadside access points.

4.2.3 Message Fabrication/Alteration

A prankster might fabricate or replay altered messages to force on-scene vehicles into recording collision data. These fictitious emergency/collision events would result in bogus data being collected by vehicles and making its way up to law enforcement authorities, wasting valuable communication, processing and storage resources. An attacker might also use this technique to mask the occurrence of a collision by diverting limited application resources to the fabricated collision.

4.2.4 Key/Certificate Replication

In this attack, an adversary would seek to undermine the system by replicating a single vehicle's identity across several vehicles. The goal of such an attacker would be to confuse the authorities and possibly prevent identification of vehicles in hit-and-run incidents (assuming the attack were carried out on a large enough scale).

4.2.5 Rogue Roadside Access Points

Because AutoCore makes use of RAPs as CRCs for the delivery of collision reports, a rogue RAP could compromise the authenticity of this data. Similar threats are possible through handheld devices that an attacker could configure to masquerade as a handheld CRC issued to law enforcement authorities. Such a rogue CRC could simply collect reports and delete them, preventing law enforcement authorities from obtaining necessary evidence for an investigation.

With RAPs also being used for the delivery of fresh credentials to vehicles (we describe this process in sections 5.5 and 5.6), rogue RAPs could compromise this process, potentially leaking valid vehicle credentials to an attacker.

5 Security Infrastructure

In this section, we begin by discussing the use of a TPD to store secret data and protect the integrity of AutoCore software in vehicles. We then describe the certificate authorities required to support our security infrastructure, introduce each of the four types of cryptographic elements used by our security infrastructure: *Vehicle Identifiers*, *Road-worthiness Certificates*, *Electronic License Plates* and *Anonymous Credentials*. We end by describing an efficient scheme for the roaming of vehicles outside their home GTA's jurisdiction.

5.1 Tamper-Proof Device

Protection of sensitive data stored in vehicles, such as collision reports produced by AutoCore and the cryptographic keys described in the following subsections mandates the use of a Tamper-Proof Device (TPD). We assume that the TPD is similar to a TPM (Trusted Protection Module), as defined by the Trusted Computing Group [20] and already deployed in many computers. Namely, the device is passive, that is, it can generate key pairs and perform signing operations, but does not run software. A TPD guarantees that the generated private keys never leave the device (or only in encrypted form). TPDs contain sensors that can detect tampering and erase all the sensitive information protected by the device. With the help of the private keys embedded in the TPD, software using the TPD can authenticate a vehicle to infrastructure (roadside access points and handhelds issued to law enforcement officials) and prove that the TPD has not been tampered with, as described in sections 5.5 and 5.6.

TPMs provide only protection against software-based attacks. Since VANETs are used for safety applications, we require TPDs to also resist hardware-based attacks. Furthermore, for software that uses a TPM, the TPM provides mechanisms to authenticate the state of this software at the software's load time. For TPDs, we assume that these mechanisms have been extended to ensure that the software is in a predefined state throughout its runtime and that sensitive information protected by the TPM become inaccessible as soon as this software is being tampered with. For example, secure co-processors provide this functionality by running software within a tamperproof box. However, secure co-processors tend to be expensive and slower than current desktop computers. The exact design of a TPD is therefore topic of future research.

The TPD uses secure storage for storing collision reports and signing keys. This storage is either embedded in the device or external. In the latter case, stored data must be encrypted, its integrity ensured and the TPD must defend against replay attacks.

In the rest of this paper, we have the term "TPD" cover both the actual TPD and any software that makes usage of the TPD and that is protected by the TPD, as explained above.

5.2 Certificate Authorities

We envision the presence of several certificate authorities to support our security infrastructure:

5.2.1 Vehicle Manufacturers

Currently, vehicle manufacturers issue a unique Vehicle Identifier Number (VIN) to all vehicles that they produce. These numbers are stamped onto the frame of a vehicle, effectively binding VINs to vehicles for their operational lifetime. Similarly, manufacturers could issue vehicles with *Vehicle Identifiers* (see section 5.3) that can be cryptographically verified and are bound to the vehicle for its operational lifetime.

5.2.2 Governmental Transportation Authorities (GTAs)

Just as these authorities register vehicles and issue physical license plates, GTAs will issue *Electronic Licence Plates* (see section 5.5) to vehicles registered in their region of jurisdiction. In addition, GTAs will issue vehicles that operate in their jurisdictions and that hold valid Electronic Licence Plates (not necessarily issued by the same GTA) with *Anonymous Credentials* (see section 5.6) allowing the vehicles to communicate with other vehicles in the region. We assume that GTAs have certificates for all vehicle manufacturers registered in their jurisdiction.

5.3 Vehicle Identifiers

Vehicle Identifiers are used to uniquely identify vehicles. A vehicle identifier consists of a signing key pair (VID_{Pu}, VID_{Pr}) and a corresponding certificate VID_{Cert}, which contains information uniquely identifying the vehicle (e.g., its VIN number) and that binds this information to the vehicle's public key VID_{Pu}. Such a public key is equivalent to today's VIN numbers. The pair (VID_{Pu}, VID_{Pr}) is created by a vehicle's TPD and VID_{Cert} is issued and installed in a vehicle's TPD by its manufacturer during production. Vehicle Identifiers are valid for the lifetime of a vehicle.

5.4 Road-worthiness Certificates

A Road-worthiness Certificate $RoadWorthy_{Cert}$ is issued to a vehicle by its manufacturer or authorized inspection authorities. Such a certificate proves that the vehicle has been inspected and approved for road-worthiness (safety checks, emissions, etc.). The certificate lists the vehicle's VID_{Pu} and is valid for the period of time for which the vehicle has been deemed to be road-worthy. We assume that the vehicle's home GTA holds certificates for each inspection authority, which allows the GTA to validate Road-worthiness Certificates. These certificates are used by a vehicle to renew its Electronic License Plate.

5.5 Electronic License Plates

Electronic License Plates (ELPs) serve the same purpose as physical license plates. ELPs consist of a signing key pair (ELP_{Pu}, ELP_{Pr}) and a corresponding certificate ELP_{Cert}, which binds the vehicle's VID_{Pu}, contained in VID_{Cert}, to its public key ELP_{Pu} under a digital signature produced by the vehicle's home GTA. The certificate is valid for the duration of the vehicle's registration (about a year).

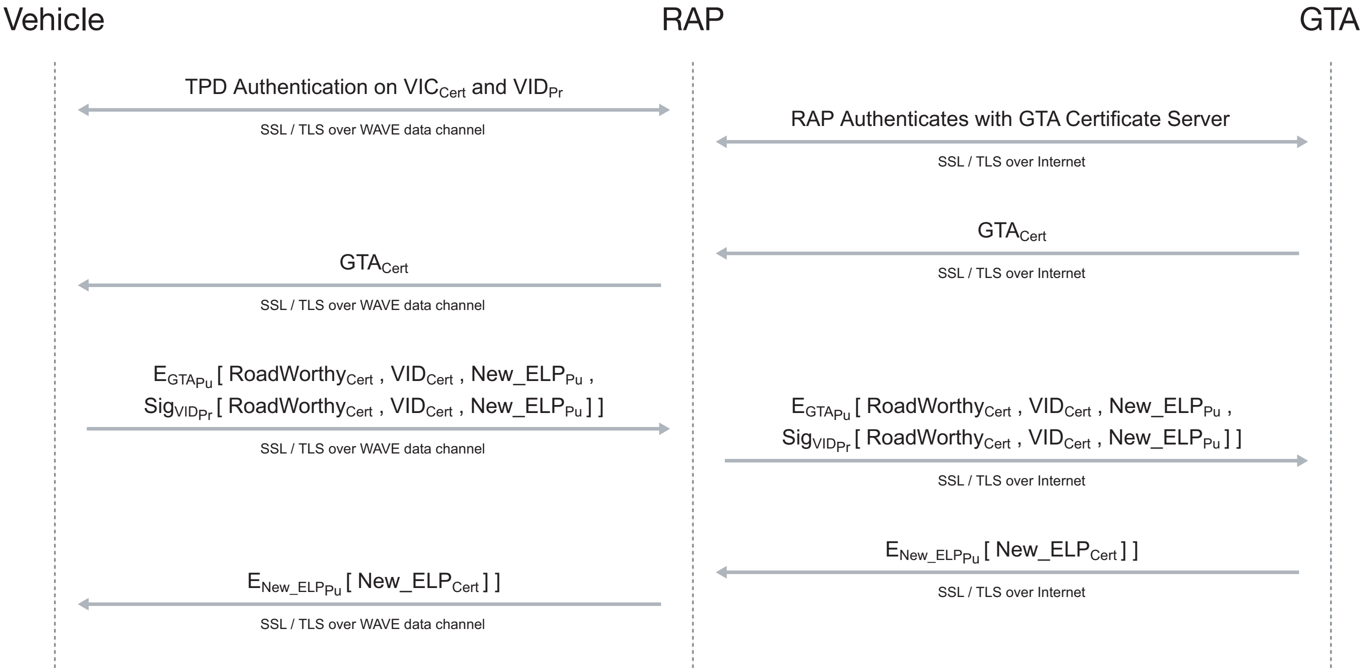
Vehicle
RAP
GTA
TPD Authentication on VIC$_{Cert}$ and VID$_{Pr}$
SSL / TLS over WAVE data channel
RAP Authenticates with GTA Certificate Server
SSL / TLS over Internet
GTA$_{Cert}$
SSL / TLS over Internet
GTA$_{Cert}$
SSL / TLS over WAVE data channel
$E_{GTA_{Pu}}$ [RoadWorthy$_{Cert}$, VID$_{Cert}$, New_ELP$_{Pu}$, Sig$_{VID_{Pr}}$ [RoadWorthy$_{Cert}$, VID$_{Cert}$, New_ELP$_{Pu}$]]
SSL / TLS over WAVE data channel
$E_{GTA_{Pu}}$ [RoadWorthy$_{Cert}$, VID$_{Cert}$, New_ELP$_{Pu}$, Sig$_{VID_{Pr}}$ [RoadWorthy$_{Cert}$, VID$_{Cert}$, New_ELP$_{Pu}$]]
SSL / TLS over Internet
$E_{New\_ELP_{Pu}}$ [New_ELP$_{Cert}$]]
SSL / TLS over Internet
$E_{New\_ELP_{Pu}}$ [New_ELP$_{Cert}$]]
SSL / TLS over WAVE data channel

Figure 2: Protocol for the renewal of ELPs

Vehicles initially acquire or renew their ELPs through roadside access points (RAPs) using the Road-worthiness Certificates described in section 5.4. Figure 2 shows the renewal protocol. The protocol works as follows:

When a vehicle reaches a RAP that advertises ELP renewal services for its home GTA, the vehicle's TPD authenticates with the RAP using its VID_{Cert} and VID_{Pr}. For example, the authentication can use SSL/TLS with client authentication. The purpose of client authentication is to demonstrate to the RAP that the vehicle has not been tampered with; any tampering with the TPD would have erased VID_{Pr}, preventing the vehicle from authenticating. After successful authentication, the RAP will be willing to act as a relay between the vehicle and the local GTA.

To guard against compromised RAPs and attackers masquerading as legitimate RAPs, each RAP has a signing key pair (INF_{Pu}, INF_{Pr}) and a corresponding certificate INF_{Cert}, issued by the local GTA. The TPD ensures the validity of this certificate when authenticating with the RAP.

The vehicle first receives the public key of the GTA, signed by a publicly known CA, such as VeriSign. We assume that the TPD has some CA certificates embedded in it, similar to a Web browser, and information that allows it to identify certificates belonging to GTAs.

The TPD then generates a signing key pair (New_ELP_{Pu}, New_ELP_{Pr}) and sends New_ELP_{Pu}, along with $RoadWorthy_{Cert}$ and VID_{Cert} to the RAP for forwarding to the GTA. The message is signed with VID_{Pr}. For privacy reasons, the message is encrypted with the public key of the GTA. Once the GTA has decrypted the ciphertext and verified the certificates, the vehicle's signature and potentially other, external conditions, such as payment of fees or traffic tickets, it issues New_ELP_{Cert} covering New_ELP_{Pu} to the vehicle via the RAP. Similar to the vehicle-RAP connection, the RAP-GTA connection is also secured with SSL/TLS.

5.6 Anonymous Credentials

Anonymous Credentials consist of a signing key pair ($AnonCred_{Pu}$, $AnonCred_{Pr}$) and a certificate $AnonCred_{Cert}$ covering $AnonCred_{Pu}$. The certificate is issued by a GTA (not necessarily a vehicle's home GTA) and contains no public information that could be used by an unauthorized observer to identify the vehicle. Vehicles will possess a set of Anonymous Credentials and use the signing key $AnonCred_{Pr}$ of a credential to sign outgoing AutoCore messages. The corresponding certificate $AnonCred_{Cert}$ accompanies such a message. To avoid tracking of a vehicle based on $AnonCred_{Cert}$, the vehicle changes credentials often using a variable-frequency key changing algorithm [18].

A certificate $AnonCred_{Cert}$ consists of

$AnonCred_{Pu}|InvisibleIdentity|\text{GTA\_GUID}|Sig_{GTA_{Pr}}[AnonCred_{Pu}|InvisibleIdentity|\text{GTA\_GUID}]$.

The $InvisibleIdentity$ field in the certificate consists of

$E_{Court_{Pu}}\,[\,E_{GTA_{Pu}}\,[\,\text{VID}\,]\,]$.

GTA_GUID is a globally unique identifier assigned to the issuing GTA and VID is a unique identifier (per GTA) assigned to each ELP (similar to a license plate number). The two identifiers are included in an ELP. $AnonCred_{Cert}$ includes GTA_GUID in plaintext so that the identity of the GTA (and the court) that can decrypt the $InvisibleIdentity$ field can be determined.

An $InvisibleIdentity$ is invisible because it is first encrypted using the issuing GTA's public key and then encrypted again using the public key of a trusted legal entity (in our case, a local Court of Law). This double encryption ensures that a vehicle's identity is hidden and can be revealed only when both the local court of law and the GTA co-operate. Note that we need a randomized encryption scheme for producing the $InvisibleIdentity$ field. This way, a vehicle's $InvisibleIdentity$

field will be different in each of its Anonymous Credentials, preventing tracking of the vehicle based on this field.

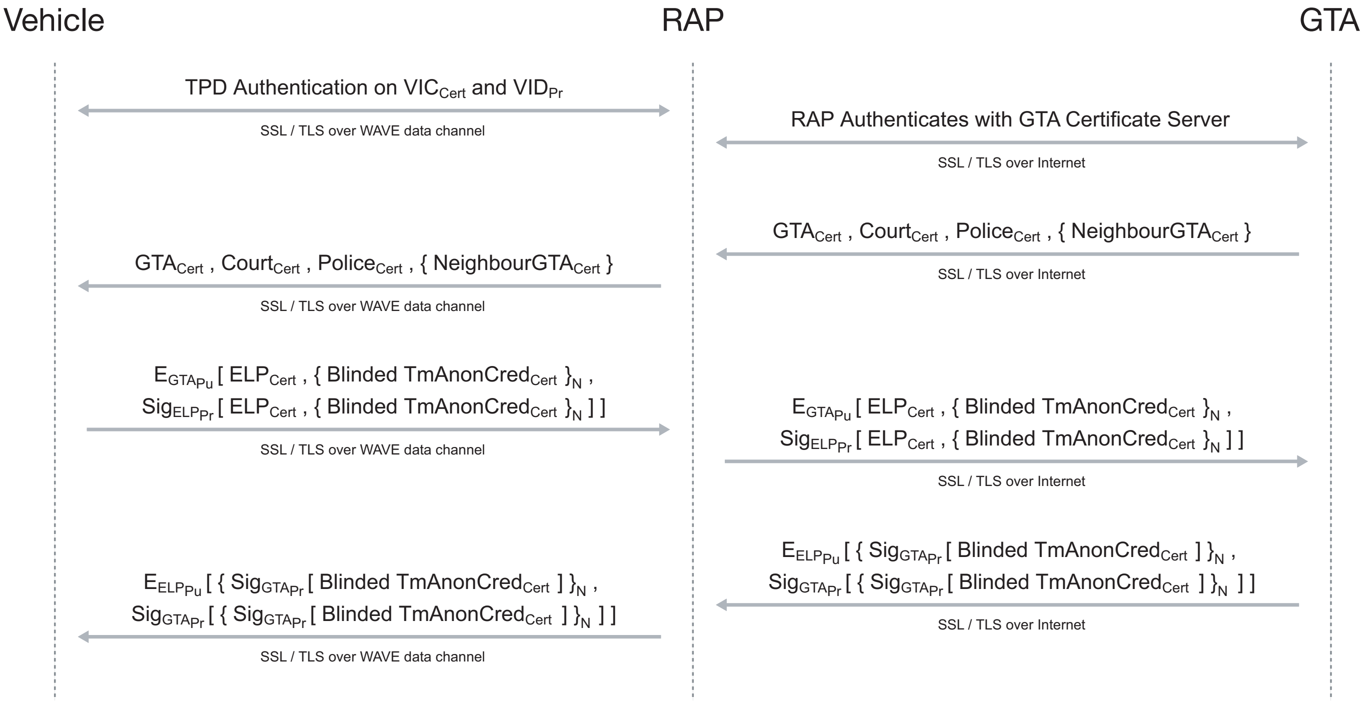
Vehicle
RAP
GTA
TPD Authentication on VIC_{Cert} and VID_{Pr}
SSL / TLS over WAVE data channel
RAP Authenticates with GTA Certificate Server
SSL / TLS over Internet
GTA_{Cert} , $Court_{Cert}$, $Police_{Cert}$, { $NeighbourGTA_{Cert}$ }
SSL / TLS over Internet
GTA_{Cert} , $Court_{Cert}$, $Police_{Cert}$, { $NeighbourGTA_{Cert}$ }
SSL / TLS over WAVE data channel
E_{GTAPu} [ELP_{Cert} , { Blinded $TmAnonCred_{Cert}$ }$_N$, Sig_{ELPPr} [ELP_{Cert} , { Blinded $TmAnonCred_{Cert}$ }$_N$]]
SSL / TLS over WAVE data channel
E_{GTAPu} [ELP_{Cert} , { Blinded $TmAnonCred_{Cert}$ }$_N$, Sig_{ELPPr} [ELP_{Cert} , { Blinded $TmAnonCred_{Cert}$ }$_N$]]
SSL / TLS over Internet
E_{ELPPu} [{ Sig_{GTAPr} [Blinded $TmAnonCred_{Cert}$] }$_N$, Sig_{GTAPr} [{ Sig_{GTAPr} [Blinded $TmAnonCred_{Cert}$] }$_N$]]
SSL / TLS over Internet
E_{ELPPu} [{ Sig_{GTAPr} [Blinded $TmAnonCred_{Cert}$] }$_N$, Sig_{GTAPr} [{ Sig_{GTAPr} [Blinded $TmAnonCred_{Cert}$] }$_N$]]
SSL / TLS over WAVE data channel

Figure 3: Protocol for the renewal of Anonymous Credentials.

To ensure the conditional anonymity of vehicles (as described in section 2), we use a blind signature scheme [5] for the certification of Anonymous Credentials by GTAs. Our scheme has the advantage that a GTA cannot learn a vehicle's $AnonCred_{Pu}$'s while being ensured that the vehicle's identity can be recovered from $InvisibleIdentity$ (if approved by a court). We present the protocol in figure 3. We now briefly outline the details of this protocol.

When a vehicle encounters a RAP that advertises Anonymous Credentials refresh services, it authenticates with the RAP using the same process described earlier for the renewal of ELPs. In the next step, the vehicle gets the certificates for the local GTA, the local court and the local law enforcement authorities. The certificates are all signed by a publicly known CA. The vehicle also gets certificates for neighbouring GTAs of the local GTA. These certificates will be used when the vehicle travels between GTA jurisdictions (see section 5.7). The vehicle then executes the following protocol:

1. First, the vehicle computes the number of Anonymous Credentials that it will require given the amount of credentials that it already holds and its distance-to-empty (remaining driving range given current fuel). This number, N, can be computed using the variable-frequency key-changing algorithm mentioned earlier.

2. The vehicle's TPD then generates N key pairs ($AnonCred_{Pu}$, $AnonCred_{Pr}$) and produces certificate templates for each of these ($TmAnonCred_{Cert}$). The templates contain all the information contained in a certificate, except the GTA's signature. The templates are then blinded and sent to the GTA, along with the number of certificates required by the vehicle N and its electronic license plate ELP_{Cert}. This information is encrypted with GTA_{Pu} and signed with ELP_{Pr}.

3. The GTA decrypts the ciphertext, validates the signature and ensures that ELP_{Cert} has not expired. If successful, the GTA signs each of the blinded certificate templates and returns them to

the vehicle.

4. The vehicle then unblinds the signatures and combines them with the original templates, completing the refresh process.

In the protocol, the GTA signs blinded certificate templates. Therefore, the GTA cannot verify whether these templates have the required structure, as shown above. Furthermore, it cannot check whether the value encrypted in the $InvisibleIdentity$ field is correct. Certificate templates are generated by the TPD. Since the vehicle's request message is signed with its ELP_{Pr}, the GTA can assume that the TPD has not been tampered with and that the template and the embedded ciphertext are correct. To deal with malfunctioning TPDs that, for example, include a wrong ID in the $InvisibleIdentify$ field, we could use a cut-and-choose protocol, where the vehicle sends $M > N$ blinded templates to the GTA and where the GTA asks the vehicle to unblind $M - N$ randomly chosen templates before signing the remaining ones. Note that the unblinding process also requires the vehicle to submit the random values used in the generation of $InvisibleIdentity$ to the GTA so that the GTA can repeat the two encryption operations. The disadvantage of the cut-and-choose protocol is that it increases the load on a GTA, especially when $M \gg N$.

5.7 Travel between GTA Jurisdictions

Vehicles communicating under our security scheme, as described so far, are only capable of authenticating (and thus reacting to) messages generated by vehicles and infrastructure from their home GTAs. Therefore, we need to extend our scheme to allow vehicles to communicate with vehicles and infrastructure certified by other GTAs, enabling communication while traveling outside the home GTA's jurisdiction.

Before proposing our solution, we introduce some terminology. Vehicles operating outside their home GTA's jurisdiction are termed *Visitors*, while vehicles/infrastructure operating/deployed within their home GTA's jurisdiction are termed *Locals*. A *Foreign GTA* is the GTA responsible for the region in which a *Visitor* is operating, while a *Home GTA* is the GTA responsible for the region the Visitor is registered in.

In our solution, we allow Visitors to acquire Anonymous Credentials from a Foreign GTA. This approach requires that the Foreign GTA maintains a list of trusted GTAs, one of them being the Visitor's Home GTA. This way, the Foreign GTA can verify ELPs certified by the Home GTA. Since a Visitor will ultimately encounter a RAP that provides Anonymous Credentials refresh services, this approach guarantees that the Visitor will ultimately be able to communicate with other vehicles and infrastructure in the region serviced by the Foreign GTA.

In the interval before the Visitor encounters this RAP, the Visitor will not be able to communicate with vehicles or services in the Foreign GTA's region. We propose the following solution for this problem: As described in section 5.6, vehicles receive a set of $NeighbourGTA_{Cert}$ certificates when they obtain Anonymous Credentials. This set includes GTA certificates for each neighbouring Foreign GTA, $ForeignGTA_{Cert}$'s, signed by the Home GTA, and a set of certificates for the Home GTA, $HomeGTA_{Cert}$'s, each of them signed by a neighbouring Foreign GTA. Similarly, each infrastructure element, like a RAP or a handheld CRC, is given the same set of certificates when it is certified by its Home GTA. We assume cooperation between neighbouring GTAs to achieve this.

With this technique, Visitors are able to authenticate messages produced by Locals by verifying the certificate attached to such a message against the set of $ForeignGTA_{Cert}$'s that the Visitors hold.

To allow Locals to authenticate messages produced by Visitors, Visitors include one of their $HomeGTA_{Cert}$'s in their messages, in particular, the certificate issued by the Foreign GTA. Locals validate $HomeGTA_{Cert}$ and add the public key of the Visitor's Home GTA to their list of trusted

GTAs.

Including $HomeGTA_{Cert}$ in messages increases overall message size (see section 8.1 for actual message sizes). In case the communication channel is congested, a Visitor could add $HomeGTA_{Cert}$ only to a subset of its sent messages. We leave a more thorough exploration of this technique to future work. Note that, as soon as a Visitor obtains Anonymous Credentials from the Foreign GTA, it no longer has to transmit $HomeGTA_{Cert}$.

6 Securing AutoCore

In this section, we discuss how the security infrastructure introduced earlier is used to protect AutoCore communications, as described in section 3.2.

6.1 Securing Inter-Vehicle Communications

In securing inter-vehicle communications, we are primarily concerned with ensuring the authenticity of AutoCore messages and guaranteeing non-repudiation of these messages while protecting vehicle location and identity privacy. We achieve these goals through the use of Anonymous Credentials.

Collision and Witness Beacons produced by AutoCore (and other safety messages) follow the format shown below:

$M, T, Sig_{AnonCred_{Pr}}[M|T], AnonCred_{Cert}$.

M is the message, T is a timestamp included to ensure message freshness and $AnonCred_{Cert}$ is the Anonymous Credential certificate signed by the local GTA that corresponds to $AnonCred_{Pr}$ used to sign the message. The signature guarantees message authenticity and non-repudiation. If necessary, a court and a GTA can jointly determine the identity of the sending vehicle by decrypting the $InvisibleIdentity$ field in $AnonCred_{Cert}$. We discuss this process in section 7.4.

6.2 Securing Vehicle to Infrastructure Communications

Communication between vehicles and infrastructure is required for the delivery of data. Examples of such data include the cryptographic information held by vehicles, as described in section 5, and collision reports produced by AutoCore. Vehicle to infrastructure communication is secured using standard mutual authentication and secure data transfer protocols, such as SSL/TLS with client authentication.

As mentioned in section 5.5, we assume that each RAP has a signing key pair (INF_{Pu}, INF_{Pr}) and a corresponding certificate INF_{Cert}. We make the same assumption for other kinds of roadside infrastructure, such as handhelds issued to law enforcement officials. This way, a vehicle can detect fake infrastructure.

6.3 Securing AutoCore Collision Reports

As mentioned in section 4, it is necessary to cryptographically protect AutoCore collision reports in order to guarantee their integrity and prevent the abuse of information contained in them.

To describe how reports are secured, we refer back to the AutoCore collision report format shown in figure 1. When vehicles have finished recording video evidence and telemetry data, they place this data in a report with all Collision and Witness Beacons received for the corresponding collision event. Each report's header contains the event's timestamp and location, as recorded by the vehicle generating the report. The reports is then signed with the reporting vehicle's current private key $AnonCred_{Pr}$, with the corresponding certificate $AnonCred_{Cert}$ included for verification purposes.

These items combine to produce the complete report shown in figure 1. The items shaded in dark gray are encrypted using the public key $Police_{Pu}$ of the local law enforcement authority when a vehicle encounters a CRC to deliver its report to. Vehicles obtain this public key in $Police_{Cert}$ when obtaining Anonymous Credentials. Reports that cannot be retrieved from a vehicle through a CRC, perhaps because the vehicle is badly damaged, are obtained directly from the vehicle in cleartext form, as described in section 3.2.

When law enforcement authorities obtain these reports, they decrypt them using $Police_{Pr}$, if necessary, and verify the submitting vehicle's signature to ensure that the report is authentic. Next, they can revoke the conditional anonymity of the Beacon messages based on the process discussed in section 7.4. Note that the police should decrypt a collision report only if certain standards have been met (e.g., the accident or hit-and-run incident have been reported). In particular, for privacy reasons, the police should not pro-actively decrypt collision reports.

During each delivery step, the CRC or law enforcement authority receiving a collision report returns a signature, created with INF_{Pr} or $Police_{Pr}$ and covering the received report, to the delivering vehicle or RAP. This establishes an audit trail.

7 Security Analysis

Our security scheme is designed to meet the following requirements:

- *Authentication* – Since vehicles should only react to legitimate AutoCore messages, it is necessary to authenticate these messages.

- *Data Consistency* – To guard against false AutoCore messages produced by legitimate vehicles (e.g., in the case of malfunctioning equipment or software), it should be possible to verify the consistency of AutoCore messages.

- *Non-repudiation* – It should be possible to reliably identify vehicles from the AutoCore messages they transmit such that drivers cannot deny their vehicles were the source of these messages, allowing investigators to determine liability in the event of a collision.

- *Privacy* – The privacy of drivers against unauthorized observers should be guaranteed, preventing the misuse of AutoCore messages to track the movements of vehicles.

- *Real-time Constraints* – Due to the short contact times between vehicles (e.g., in the case of vehicles crossing in opposite directions on a highway) and the short duration of crash events (typically no more than a few seconds), strict time constraints must be placed on the delivery and processing of AutoCore messages.

In the following subsections, we review the above requirements in the context of our crash reporting application and describe how our security infrastructure satisfies each of them.

7.1 Authentication

All messages produced by vehicles are signed with their current $AnonCred_{Pr}$ and authenticated using the corresponding $AnonCred_{Cert}$. While this guarantees only that a message comes from a vehicle that was trustworthy when it was issued $AnonCred_{Cert}$, it does prevent outsiders (not authorized to communicate with vehicles by the GTA) from sending authenticated messages. When a TPD is being tampered with, it will erase all its $AnonCred_{Pr}$'s and its other private keys and will no longer be able to sign messages or renew its credentials.

7.2 Data Consistency

Data consistency depends on the application in question. In AutoCore, data consistency is provided by having AutoCore correlate Collision Beacons with sudden braking or a sudden change in direction by the host vehicle. In addition, if more than one vehicle equipped with AutoCore is involved in a collision, it is possible to correlate Collision Beacons sent by these vehicles by checking the timestamps and locations included in the messages.

7.3 Non-repudiation

We achieve non-repudiation as follows:

- A vehicle cannot claim to be a different vehicle, because it signs messages with its own private keys. Furthermore, ELPs are unique and only one vehicle holds the corresponding ELP_{Pr} in its TPD.
- A vehicle cannot deny having sent messages because a message is signed with $AnonCred_{Pr}$, which belongs to the vehicle and was generated by the vehicle in the first place. Timestamps included in each message guard against message replay attacks.

7.4 Privacy

In our solution, vehicles send messages in an anonymous way, which defends against vehicle identification or positioning attacks, as introduced in section 4. However, this anonymity is conditional, that is, in the case of an accident, it must be possible by authorities to revoke this anonymity. We now address these two issues in more detail.

7.4.1 Anonymity

Privacy against vehicle identification attacks is guaranteed through the absence of any public information about a vehicle in $AnonCred_{Cert}$'s sent out by the vehicle. Furthermore, a GTA that issues an $AnonCred_{Cert}$ does not see the contents of these certificates and will thus not be able to re-identify vehicles by colluding with roadside infrastructure.

Privacy against vehicle positioning attacks is guaranteed through frequently changing the Anonymous Credential used by a vehicle. Since each Anonymous Credential looks different, these credentials cannot be used for tracking a vehicle. Furthermore, a vehicle must also frequently change its MAC or IP addresses.

7.4.2 Revocation of Conditional Anonymity

In the event of a collision, law enforcement authorities may want to learn the identities of vehicles that sent messages included in collision reports. To reveal these identities, the authorities will take a report to a Court of Law. For each beacon message in the report, the Court of Law will remove the first layer of encryption from the $InvisibleIdentity$ field in $AnonCred_{Cert}$ attached to the message. The second layer of encryption will be removed by the GTA upon receipt of a valid court order, which will reveal a vehicle's identity.

7.5 Real-time Constraints

We discuss issues with real-time constraints in section 8.1, where we address the problem of choosing a suitable cryptosystem for each of the cryptographic elements introduced in section 5.

8 Implementation Issues

In this section, we examine two implementation issues, namely, what kind of cryptosystem to use in our security model and how to implement imaging in AutoCore.

8.1 Choice of Cryptosystems

Every safety message sent out by a vehicle, such as a Collision Beacon, contains, in addition to its payload, a digital signature and a certificate for the corresponding public key. To reduce overhead, we need cryptosystems with short signature and key sizes. We choose ECDSA for most of the signing key pairs. The only exception is the signing key pair used by GTAs for certifying Anonymous Credentials. As mentioned in section 5.6, these signatures are issued in a blind way. However, no blind signature scheme based on ECDSA is currently known. Instead, we use a blind signature scheme based on the BLS short signature scheme [1], referred to as BBLS ("Blind BLS") in the rest of this section. We discuss the scheme in appendix A.

For ECDSA, we choose a public key size of about 20 bytes, which results in signatures of 40 bytes. This setup provides sufficient security in the short and medium term. It is possible to use larger key sizes for improved security. For example, Raya and Hubaux [19] choose 28 bytes ECDSA keys. However, there is no immediate need since there is no long-term storage of safety messages. For BBLS, we choose a public key size of about 75 bytes, which results in signatures of size about 25 bytes. Note that BBLS public keys are not exchanged in safety messages.

Finally, we choose the Elliptic Curve Integrated Encryption Scheme (ECIES) [4] for encrypting a car's identity in the *InvisibleIdentity* field. ECIES also generates a MAC, which we leave away, since the integrity of the *InvisibleIdentity* field is assured by the certificate in which the field is embedded. Using 20 byte public keys, the size of the *InvisibleIdentity* field will be 44 bytes, that is, 4 bytes for the actual ciphertext (we assume that VID has a size of 4 bytes) and 20 bytes for each of the two random elliptic curve points (i.e., their x-coordinates) output in the two encryption steps.

Given this setup, the security overhead of a safety message is about 133 bytes. In particular, the ECDSA signature of the message is 40 bytes long. The accompanying certificate has a size of 93 bytes: 20 bytes for $AnonCred_{Pu}$, which is an ECDSA public key, 25 bytes for the BBLS signature, 4 bytes for the GTA_GUID and 44 bytes for the *InvisibleIdentity* field. For comparison, Xu et al. [21] estimate the typical payload size of a safety message to be between 100 and 400 bytes. The overhead in Raya and Hubaux's scheme is 140 bytes.

As mentioned in section 5.7, when traveling between GTA jurisdictions, a vehicle might temporarily include a certificate in which the Foreign GTA certifies the vehicle's Home GTA in some of its messages. The size of such a certificate is 60 bytes; the public key of the Home GTA needs 20 bytes and the ECDSA signature requires 40 bytes.

A vehicle generating a safety message needs to create an ECDSA signature. Similarly, a vehicle receiving a safety message needs to verify this signature and the BBLS signature of the certificate accompanying the signature. On a Pentium IV 3 GHz, it takes about 0.68 ms to generate an ECDSA signature and 1.3 ms to verify the signature. It takes about 49.7 ms to verify a BBLS signature (and 2.8 ms to create it). Raya and Hubaux [18] estimate that a vehicle has only about 2.5 ms for processing a message, which is much less than the time it takes to verify a BBLS signature. However, this signature needs to be validated only once for an Anonymous Credential used by a vehicle. Any additional messages using the same credential no longer require this overhead. For example, assume the same scenario as introduced by Raya and Hubaux [18], a highway with six lanes (three in each direction) and an inter-vehicle distance of 30 m. Vehicles transmit safety messages every 300 ms over a 300 m communication range. Here, within a second, a vehicle driving at 120 km/hour is going to see about nine new Anonymous Credentials per second from vehicles traveling in the opposite direction and six from vehicles traveling in the same direction, which leaves sufficient time to check the signature of each new Anonymous Credential.

8.2 AutoCore Imaging

For imaging data, precise requirements will likely vary depending on the jurisdiction a vehicle is operating in (i.e., the evidence requirements of local law enforcement authorities), but a very minimal setup consists of an omni-directional video camera mounted on the roof or boot of a vehicle. One such system developed by Nayar and Peri [13, 17] is capable of taking images from an omni-directional camera and generating pure perspective images. Nayar and Narasimhan [14] explore the adaptation of such computer vision systems for use in bad weather and low-light conditions.

9 Related Work

Gerlach [7] highlights key VANET security concepts and proposes a model for trusted inter-vehicle communication. Parno and Perrig [15] further examine VANET security issues, identify potential attacks and introduce a categorization scheme for adversaries. We build on these threats in section 4.

Hubaux et al. [9] focus their efforts on vehicle identity and location privacy, introducing Electronic License Plates (ELPs) that serve the same purpose as physical license plates. In Hubaux et al.'s scheme, an ELP is simply an identifier. Instead, the ELPs proposed in our work consist of signing key pairs and certificates. This way, it becomes possible to use ELPs to refresh Anonymous Credentials via RAPs. Hubaux et al. also introduce the concept of Electronic Chassis Numbers (ECNs) that can be used to uniquely identify vehicles. Our Vehicle Identifiers serve a similar purpose, but instead of simply being identifiers, they also consist of signing key pairs and certificates and can be used to bootstrap the renewal of ELPs via RAPs. Note that although Hubaux et al. mention that ELPs can be renewed upon vehicle registration, they do not present an actual renewal scheme.

Raya and Hubaux [18] build on this earlier work and introduce a security framework for VANETs, proposing the use of *Anonymous Keys* to sign messages sent by vehicles. To provide conditional anonymity, Anonymous Keys mandate the creation of a centralized database that maps these keys to a vehicle's identity. This approach has the drawback that the database becomes a single point of failure. To avoid the abuse of this database, Raya and Hubaux suggest encrypting this database with shared secrets split between authorities. Raya and Hubaux do not elaborate on the certification and distribution process of Anonymous Keys. In particular, there is the danger that, while these keys are being certified, a malicious certificate issuer, like a GTA (or an intruder), can store mappings between public keys and identities in a second database, which is not protected with a secret-sharing system. In our approach, as explained in section 5.6, the certifier never sees the public key in the Anonymous Credential that is being certified. Furthermore, our approach includes the information necessary for revoking anonymity directly in Anonymous Credentials, thereby eliminating the need for a centralized database. Finally, we describe an actual scheme for the generation and distribution of Anonymous Credentials.

In order to allow vehicles to communicate with infrastructure and other vehicles in regions governed by foreign GTAs, Raya and Hubaux [19] suggest the use of base stations deployed at borders to verify and re-certify a vehicle's Anonymous Keys. The disadvantage of this approach is that it requires that a (working) base station be deployed at every single border crossing or even just a crossing between two provinces/states, in case the provinces/states are governed by different GTAs (as in the case of Canada or the US). This assumption seems unrealistic. Moreover, when vehicles do not stop when crossing an inter-GTA border, the amount of time required to verify and to re-certify a vehicle's Anonymous Keys may be too large, with moving vehicles going out of range before the process is completed. (WAVE is currently limited to a range of 1 km [10].) Our solution avoids these problems through the use of $NeighbourGTA_{Cert}$ certificates, eliminating the need for base stations at borders.

Instead of Anonymous Credentials, we could also use an anonymous credential system, such as Idemix [3] or Brands credentials [2]. However, these systems tend to be more expensive. For example, issuing a Brands credential takes three steps, whereas our protocol requires only two.

10 Conclusion

We have introduced AutoCore, an automated crash reporting application that provides cryptographically-verifiable evidence of an automobile crash in the form of digital video and telemetry data recorded by vehicles either involved in or at the scene of the crash. To secure AutoCore, we have presented a security infrastructure that extends the state of the art in VANET security. We have analyzed this security infrastructure to demonstrate its robustness and efficiency.

Directions for potential future work include a more thorough exploration of the inter-GTA travel scheme proposed in section 5.7, a simulation of the credentials distribution schemes described in sections 5.5 and 5.6 to determine minimum required vehicle contact times with RAPs and an evaluation of our security infrastructure to assess its suitability for other VANET safety applications.

Acknowledgments

We are grateful to S. Keshav for suggesting the crash reporting application and for precious comments on a preliminary version of this paper. We also thank Ian Goldberg for suggesting and extending the BLS short signature scheme [1] and for suggesting the ECIES encryption scheme [4].

References

[1] D. Boneh, H. Shacham, and B. Lynn. Short Signatures from the Weil Pairing. *Journal of Cryptology*, 17(4):297–319, 2004.

[2] S. Brands. A Technical Overview of Digital Credentials. Technical report, Credentica, February 2002.

[3] J. Camenisch and E. Van Herreweghen. The Design and Implementation of the Idemix Anonymous Credential System. In *Proceedings of 9th ACM Conference on Computer and Communications Security (CCS 2002)*, pages 21–30, November 2002.

[4] Certicom Research. Standards for Efficient Cryptography, SEC 1: Elliptic Curve Cryptography, Version 1, September 2000.

[5] D. Chaum. Blind Signatures for Untraceable Payments. In *Proceedings of CRYPTO '82*, pages 199–203, 1982.

[6] W. Enkelmann. FleetNet - applications for inter-vehicle communication. In *Proceedings of the IEEE Intelligent Vehicles Symposium '03*, pages 162–167, 2003.

[7] M. Gerlach. VaneSe - An Approach to VANET Security. In *Proceedings of V2VCOM 2005*, July 2005.

[8] I. Goldberg. Personal Communication, March 2007.

[9] J. P. Hubaux, S. Capkun, and J. Luo. The Security and Privacy of Smart Vehicles. *IEEE Security and Privacy Magazine*, 2(3):49–55, 2004.

[10] *IEEE P1609.2 Version 1 – Standard for Wireless Access in Vehicular Environments: Security Services for Applications and Management Messages.* 2006.

[11] D. Jiang, V. Taliwal, A. Meier, W. Holfelder, and R. Herrtwich. Design of 5.9 Ghz DSRC-based Vehicular Safety Communication. *IEEE Wireless Communications Magazine*, 13(5):36–43, 2006.

[12] B. Lynn. The Pairing-Based Cryptography Library. `http://crypto.stanford.edu/pbc/`. Accessed March 2007.

[13] S. K. Nayar. Omnidirectional Video Camera. In *DARPA Image Understanding Workshop (IUW)*, pages 235–242, May 1997.

[14] S. K. Nayar and S. G. Narasimhan. Vision in Bad Weather. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 820–827, 1999.

[15] B. Parno and A. Perrig. Challenges in Securing Vehicular Networks. In *Proceedings of 4th Workshop on Hot Topics in Networks (HotNets-IV)*, 2005.

[16] Crash Avoidance Metric Partnership. *Vehicle Safety Communication Final Report.* 2006. available through the U.S. Department of Transportation.

[17] V. N. Peri and S. K. Nayar. Generation of Perspective and Panoramic Video from Omnidirectional Video. In *DARPA Image Understanding Workshop (IUW)*, pages 243–246, May 1997.

[18] M. Raya and J. P. Hubaux. The Security of Vehicular Ad Hoc Networks. In *Proceedings of 3rd ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN 2005)*, pages 11–21, November 2005.

[19] M. Raya and J. P. Hubaux. Securing Vehicular Ad Hoc Networks. *Journal of Computer Security*, 15(1):39–68, February 2007.

[20] Trusted Computing Group. `https://www.trustedcomputinggroup.org`. Accessed February 2007.

[21] Q. Xu, T. Mak, and R. Sengupta. Vehicle-to-Vehicle Safety Messaging in DSRC. In *Proceedings of 1st ACM International Workshop on Vehicular Ad Hoc Networks*, pages 19–28, 2004.

[22] X. Yang, J. Liu, F. Zhao, and N. Vaidya. A vehicle-to-vehicle communication protocol for cooperative collision warning. In *Proceedings of MobiQuitous '04*, 2004.

[23] M. Zimmer. Personal Information and the Design of Vehicle Safety Communication Technologies: An Application of Privacy as Contextual Integrity. In *AAAS Science & Technology in Society Graduate Conference*, April 2005.

A Blind Signature Scheme

In this appendix, we present the signature scheme used for creating blind signatures, as required by the protocol in section 5.6. The scheme is based on the BLS short signature scheme [1] and extended by Goldberg [8] to support blind signing by a server. The security of the signature scheme relies on the Computational Diffie-Hellman assumption.

The BLS signature scheme works as follows (simplified):

Setup – G_1 and G_2 are two groups of points on an elliptic curve, and G_T is a subgroup of $GF(p^k)$. All three groups have the same prime order n. P is a generator of G_2 and e is a bilinear pairing $G_1 \times G_2 \to G_T$. A private key x is randomly chosen from $\mathbb{Z}_n^*$. The corresponding public key V is computed as $V = xP$.

Signature generation – To sign a message M, compute $S_M = xH(M)$, where H is a hash function that maps M onto a point $\in G_1$. The signature σ is the x-coordinate of S_M.

Signature verifiation – 1) Find a point S on the curve whose x-coordinate is σ. If there is no such point, reject the signature. 2) Set $u = e(S, P)$ and $v = e(H(M), V)$. 3) If either $u = v$ or $u^{-1} = v$, accept the signature. Otherwise, reject.

To have a server generate a signature on M without the server learning M, a client proceeds as follows:

1. The client chooses a random $r \in \mathbb{Z}_n^*$, determines $r_{inv} = r^{-1}$, and blinds M by computing $B = rH(M)$. The client sends B to the server.

2. The server signs M_B by determining $S_B = xB$ and returns S_B to the client.

3. The client computes $S_M = r_{inv} * S_B$, the signature is the x-coordinate of S_M.

We performed the evaluation of this scheme, as presented in section 8.1, for an implementation based on the Pairing-Based Cryptography Library [12].