Boston University
College of Engineering
Department of Electrical and Computer Engineering

Secure Cooperative Accident Avoidance for Vehicles

MS Project

Jimmy C. Chau
Advisor: Thomas Little
Originally submitted May 7, 2011
Last revised January 26, 2016
## Contents

1 **Introduction**  
   1.1 Motivation .................................................. 1  
   1.2 Vehicular networking technologies ...................... 1  
   1.3 Challenges .................................................. 1  
   1.4 This project ................................................ 2  
      1.4.1 Scope .................................................... 2  
      1.4.2 Organization ............................................ 3  

2 **Usage scenarios** ........................................... 3  
   2.1 Entities ..................................................... 3  
   2.2 Use cases ................................................... 5  
      2.2.1 Sudden braking ........................................... 5  
      2.2.2 Informative versus safety-critical approaches ....... 5  
      2.2.3 Cars approaching an intersection ................... 6  

3 **Design objectives** ........................................ 7  
   3.1 Robustness ................................................ 7  
      3.1.1 Against inaccessible infrastructure ................. 7  
      3.1.2 Against uncooperative vehicles ..................... 7  
   3.2 Speed ....................................................... 8  
   3.3 Data validation ........................................... 8  
   3.4 Dealing with adversaries .................................. 9  
   3.5 Sender verification ........................................ 9  
   3.6 Privacy .................................................... 9  
      3.6.1 Degrees of privacy .................................... 9
### 3.6.2 Protecting non-publicly-observable information

### 3.6.3 Unlinkability

### 4 Threat model

#### 4.1 Security goals

#### 4.2 Assumed trust

#### 4.3 Possible adversaries

#### 4.4 Adversarial powers

### 5 Existing Vehicular Ad-hoc Network (VANET) security solutions

#### 5.1 Data verification

##### 5.1.1 Position verification via time-of-flight

##### 5.1.2 Position and mobility verification using plausibility models

##### 5.1.3 Consensus

##### 5.1.4 Position verification by region-specific pseudonyms

#### 5.2 Sender verification

##### 5.2.1 Unsuitable schemes

##### 5.2.2 Using group signatures

#### 5.3 Notable weaknesses

##### 5.3.1 Executing Sybil attacks across multiple regions

### 6 Remaining problems

#### 6.1 Selectively dropping or jamming messages

#### 6.2 Initial deployment

##### 6.2.1 Non-participating vehicles

##### 6.2.2 Unlinkability as sole cooperative vehicle

##### 6.2.3 Significant probability of failure
7 New security solutions for cooperative collision avoidance

7.1 Improving data verification with sensing and directional line-of-sight communications

7.2 Additional benefits of directional line-of-sight communications

7.3 Short-term linkability

7.3.1 Short-term linkability for Temporary Anonymous Certified Keys (TACKs)

7.3.2 Short-term linkability for Message-Linkable Group Signatures (MLGSs)

7.3.3 Other methods to achieve short-term linkability

8 New protocol for detecting potential collisions

8.1 Considerations

8.2 Relative versus absolute positions

8.2.1 In favor of relative position

8.2.2 In favor of absolute position

8.3 Messages

8.3.1 Movement reporting message

8.3.2 Observation messages

8.3.3 Relay messages

9 Conclusion

10 Acronyms and abbreviations

11 Appendix

11.1 References for the poster
1 Introduction

1.1 Motivation

Automobile accidents have been a common occurrence; for example, in 2007, over ten million motor vehicle accidents occurred in the United States of America alone [4]. Many of these accidents have resulted in deaths [1]. Fortunately, many existing and emerging technologies can help to reduce the likelihood of future accidents.

In particular, vehicular networking technologies, when combined with Electronic Control Units (ECUs), may enable cars to automatically avoid certain types of accidents. ECUs already provide many life-saving features to drivers such as anti-lock brakes, stability control, and airbag activation. The ECUs responsible for these functions gather information from sensors and other ECUs in the car to automatically react if necessary, without intervention from the driver, to prevent or to mitigate damage from accidents.

The addition of vehicular networking can vastly expand the information available to ECUs within each car, allowing them to respond to a greater variety of more complex situations. Whereas today’s ECUs can only use information from sensors installed in the same car, future ECUs may also incorporate information from nearby cars by sharing information over a vehicular network.

1.2 Vehicular networking technologies

Vehicular networking is an emerging field in which a variety of solutions have been proposed to establish computer networks between vehicles. Solutions range from forming ad-hoc networks to incorporating vehicles into established networks, such as the Internet; from using undirected non-line-of-sight communications to using highly directional line-of-sight communications, such as Visible-Light Communication (VLC). The applications for vehicular networks are similarly varied. Other examples include providing in-car entertainment and providing information about traffic conditions.

This project assumes that vehicular networks form in an ad-hoc fashion where connections to a wider area network, such as the Internet, are only occasionally available.

1.3 Challenges

In order to maximize the potential for such a system, the ECUs orchestrating these functions must be allowed to act without seeking the driver’s approval; like airbags, new safety mechanisms will be much less useful if they are delayed by a human’s reaction time. At times, the car may even need to override the driver’s instructions to avoid an accident. Such an override is analogous to
how a car with anti-lock brakes may refuse to apply full pressure to a brake even if the driver fully depresses the brake pedal.

However, allowing ECUs to override the driver or to act without explicit approval raises new safety issues, especially if the ECU does so based on information provided by other vehicles. Since this information from other vehicles is not necessarily trustworthy, it may be erroneous, causing the ECU to make mistakes. For example, an adversary might maliciously transmit incorrect information to force dangerous decisions onto nearby vehicles. Alternatively, other vehicles may naturally malfunction, even without tampering, causing them to transmit incorrect information. To prevent these risks from outweighing the benefits, each vehicle needs to validate or verify the information that it receives.

Cooperation over vehicular networks also poses privacy issues. Since the participating cars must exchange information with each other in order to cooperate, cooperation may expose otherwise private information. For example, an adversary may collect position updates shared over the vehicular network, allowing the adversary to virtually follow many cars over long periods of time. By combining these position updates into routes, the adversary may be able to deduce sensitive information, such as the home and workplace of the vehicles’ occupants.

Security measures imposed to validate shared information may also inadvertently diminish privacy. For example, if vehicles are required to digitally sign their messages in order to prevent vehicles from impersonating each other, the signatures may assist adversaries in uniquely identifying each vehicle, reducing the anonymity of each vehicle since two messages signed with the same key likely originate from the same sender. If an adversary sees two messages from the same sender in two locations, the adversary can conclude that the vehicle traveled from one location to another, which again reveals the vehicle’s route.

Although the adversary might still be able to gather this information without vehicular networking, vehicular networking can make this task much less expensive by removing the need to physically follow each vehicle, thus making such attacks more likely.

1.4 This project

1.4.1 Scope

Vehicular networking can offer many additional benefits. For example, they can deliver news, traffic, and entertainment to traveling vehicles. Cooperation over vehicular networks can also offer benefits aside from accident avoidance by facilitating computer-control of vehicles, they can greatly improve the efficiency of traveling by car. However, these applications of vehicular networking are outside the scope of this project.

Specifically, this project will investigate privacy and security issues surrounding communi-
cations for cooperative collision avoidance: using cooperation over vehicular networks to enable cars to automatically avoid, or reduce the severity of, crashing into each other. Although the solutions presented by this project may also be applicable to other types of accidents, such as collisions into stationary objects or into pedestrians, this project will exclude these other applications in order to more thoroughly investigate collision avoidance between vehicles.

Furthermore, although this report may use specific maneuvers for illustrative purposes, the research presented does not aim to devise maneuvers to escape dangerous situations. Instead, this project assumes that given enough correct information about the vehicle’s environment, the vehicle’s computers or ECU will be able to determine the appropriate response.

1.4.2 Organization

This report explores cooperative collision avoidance in 9 additional sections. Section 2 enumerates the participants in this system and illustrates some scenarios in which cooperative collision avoidance can be used. Section 3 examines these use cases to identify requirements and other desirable traits for systems; these include requirements for robustness, speed, data validation, sender verification, dealing with adversaries, and privacy.

Section 4 describes the abilities of adversaries and the trust relationships that this project assumes. Section 5 evaluates existing security solutions for VANETs against the threat model, goals, and usage scenarios described in sections 2, 3, and 4. Section 5 also describes potential attacks against these existing solutions.

Section 6 explores issues that remain unresolved by the existing solutions such as problems arising from the initially sparse deployment of these cooperative systems. Section 7 introduces new concepts and techniques for secure vehicular networks. Section 8 presents and evaluates a new protocol for exchanging information to detect potential collisions. And section 9 lists some remaining issues to be investigated in future research.

2 Usage scenarios

2.1 Entities

Figure 1 illustrates the participants and interactions in a system with networked vehicles. Well-behaved cooperating vehicles, illustrated as $V_1$ and $V_2$ form the core of this system; by actively exchanging information such as position and speed, they are able to coordinate and cooperate with each other.

Roadside Units (RSUs), labeled $RSU_1$ and $RSU_2$, are stationary nodes on the vehicular
network. Where they are available, they generally provide some supporting services to the vehicular network. For example, they may provide Internet access, relay messages between vehicles, or assign credentials to well-behaving vehicles. However, the coverage of RSUs may be incomplete and their services may be unavailable in some locations.

Other supporting infrastructure include Certificate Authorities (labeled CA1 and CA2) and car factories (labeled F1 and F2). These entities are typically entrusted to manage the long-term credentials for each vehicle; CA1 may also manage the credentials for RSUs. Potential CAs include each state’s department of motor vehicles, the country’s department of transportation, or Internet CAs.

Although they do not technically participate in the cooperative vehicular network, older cars without networking capabilities also interact with the system. Especially in the beginning, the design of the system must account for their presence since most cars will initially not support cooperation. Failure to detect and account for these older vehicles may cause cooperating cars to unknowingly collide with older vehicles.

Malicious or malfunctioning entities (depicted as M1 and M2) may also exist in the system. For example, drivers may deliberately or unintentionally modify their vehicles to incorrectly report information in an attempt to manipulate traffic in their favor. In addition, adversaries, such as pranksters, may attempt to place virtual vehicles where they don’t actually exist.

Adversaries may also attempt to setup rogue infrastructure. For example, they may try to impersonate a legitimate RSU in order to collect information on passing vehicles. Generally, unless the entity is assumed to be trusted, any entity in the system can be adversarial. Furthermore, unless security measures prevent it, adversaries can impersonate any entity in this system.
Figure 2: The blue car, sensing a deer about to cross the road, can preemptively warn the truck behind itself. This preemptive warning can enable the truck to avoid rear-ending the car in case the car needs to suddenly brake to avoid the deer.

2.2 Use cases

2.2.1 Sudden braking

Vehicular networks can help vehicles avoid collisions in many scenarios. For example, it can help prevent one vehicle from rear-ending each other as illustrated in Figure 2.

In such scenarios, accidents can be prevented in a variety of ways using vehicular networking. These approaches range from being simply informative to being fully automatic. In the former category, upon detecting the deer, the blue car can warn the truck behind it, which simply passes the warning on to the truck’s driver, relying on the driver to take the appropriate action. This approach is the most benign, the simplest to implement, and the least likely to inadvertently cause an accident. However, although this advance warning is still useful, better automated approaches can be much more effective.

In the latter category, the truck can, in addition to warning the driver, preemptively slow down; if the car suddenly brakes, it can notify the truck, which will automatically brake or change lanes in response, eliminating any delays caused by the driver’s non-zero reaction time. However, with this approach, an erroneous automated response can be catastrophic.

2.2.2 Informative versus safety-critical approaches

This latter approach places stricter requirements on the security of the system; a VANET security system that is sufficient for the latter approach should also be sufficient for the former approach.

Many existing VANET security solutions have been designed to only provide adequate pro-
Figure 3: Vehicles $V_1$ and $V_2$ approaching the intersection may collide at the intersection; do to the presence of buildings, they cannot see each other until they reach the intersection. Through vehicular networking, they can notify each other of their presence by either relaying the message through $V_3$, through RSUs, or by using non-line-of-sight communications.

tection for informational applications of vehicular networking, such as relaying traffic conditions, where failures only result in a degradation of service and not in a catastrophic accident. This level of security may be sufficient for the former approach, where the system merely warns the driver about potentially dangerous situations.

However, due to the much higher stakes of collision avoidance, this project assumes the latter category, which places much stricter requirements on the security of the system; a security system that is sufficient for the latter approach should also be sufficient for the former approach.

2.2.3 Cars approaching an intersection

Although it is easier to understand, the example in section 2.2.1 does not fully illustrate the potential uses of vehicular networking in preventing collisions between vehicles. Arguably, the truck in the example can achieve similar results by using radar, Light Detection and Ranging (LIDAR), or sonar sensors to detect slowing and braking by the car ahead of it.

Figure 3, which depicts two cars approaching an intersection, better illustrates the advantages of vehicular networking. In this scenario, the two cars speeding towards the intersection will not be within line-of-sight of each other until they reach the intersection, where they will collide unless they adjust their speeds. Since they are not within line-of-sight, they cannot detect each other using radar, LIDAR or sonar.
However, they can still communicate with each other using non-line-of-sight communications, such as Radio-Frequency (RF) communications. They can also use line-of-sight communications, such as VLC, with the assistance of another car or a RSU to relay messages. By exchanging their positions and speeds with each other over the vehicular network, they may be able to detect each other’s presence early enough to avoid a crash.

Using vehicular networking, they may also be able to negotiate with each other to determine which car should slow down or speed up to avoid the collision.

3 Design objectives

3.1 Robustness

Since this project assumes that failures can result in catastrophic accidents, the cooperative collision system must be robust to avoid failures.

3.1.1 Against inaccessible infrastructure

For this reason, the system must be able to operate in the absence of an Internet connection, which may not always be available. Similarly, RSUs may not always be available; especially during the initial deployment of the system when few RSUs have been setup.

3.1.2 Against uncooperative vehicles

As described in section 2.1, not all vehicles will be able to cooperate. Especially when the cooperative collision avoidance system is initially deployed, most vehicles will not cooperate. For these reasons, the cooperative collision avoidance system ought not rely solely on communications to detect potential collisions. Otherwise, when one cooperative vehicle automatically maneuvers to avoid one collision, it may unknowingly crash into another uncooperative vehicle as illustrated in figure 4.

To detect vehicles that cannot communicate over the vehicular network, each vehicle with cooperative collision avoidance must employ additional methods. This project assumes that each cooperating vehicle will use some form of line-of-sight sensing, such as radar, [LIDAR] or sonar, to detect vehicles that do not cooperate.

Since such sensing is necessary, this project will also assume the availability of such sensing capabilities.
Figure 4: Cooperating vehicles must be able to detect uncooperative vehicles in order to make safe decisions. In this scenario, vehicle $V_2$ suddenly brakes to avoid the deer. Vehicle $V_1$ has two options: to brake or to change lanes. In scenario A, braking may cause the gray uncooperative vehicle to rear-end $V_1$; in scenario B, changing lanes will cause $V_1$ to hit the gray uncooperative vehicle. Without sensors to detect uncooperative vehicles, $V_1$ cannot tell which scenario applies.

3.2 Speed

One of the main advantages provided by automatic collision avoidance is the due to the ECU's ability to react much faster to dangerous situations than a human driver can. To maintain this advantage, the communications required for cooperation should be short to avoid delays due to transmission times. Fast algorithms should also be used where possible to avoid processing delays that may arise from operations such as generating cryptographic signatures or modeling the movement of other vehicles.

3.3 Data validation

Due to the dangers of using executing maneuvers derived from incorrect information and due to the possible presence of adversaries, the cooperative collision avoidance system must be able to validate the data it receives.

Due to the delays that would inevitably arise from needing to communicate with a centralized server and the likelihood of losing access to supporting infrastructure, this validation cannot be centralized and cannot rely on infrastructure such as RSU.

As described in section 5.1 many techniques have been published for validating information from vehicular networks. These techniques verify received information using characteristics of the method of communication such as propagation speed or transmission range, models of plausible behavior, consensus, and history.
3.4 Dealing with adversaries

In order to discourage malicious behavior over the vehicular network, and to limit the damage that successful adversaries can cause, the cooperative collision avoidance system must be able to identify the adversary (traceability), prevent the adversary from participating as a legitimate user (revocation), and if appropriate, assist law enforcement in apprehending and prosecuting the perpetrator(s).

Furthermore, to ensure that adversaries cannot feign innocence after being caught, all messages sent over the vehicular network should offer non-repudiation.

3.5 Sender verification

Many data validation techniques rely on the honest majority assumption [7, 9, 15]: that most of the participating vehicles are honest and well-behaved. However, using consensus to validate information is counter-productive if the majority of the participants are malicious. If an adversary can successfully impersonate many vehicles, the adversary can present many malicious vehicles to its target, effectively defeating the honest majority assumption.

The methods used to deal with adversaries also require the ability to identify adversaries and to prove that the adversary is indeed adversarial. Allowing adversaries to successfully assume the identity of another legitimate vehicle will thwart these protections against adversaries.

For these reasons, the sender of every exchanged message that can possibly damage another participant must be verifiable. Techniques for accomplishing this typically use Message Authentication Codes (MACs) or cryptographic signatures. MACs tend to be more computationally efficient, but cannot offer non-repudiation, which is required for dealing with adversaries.

3.6 Privacy

In order to achieve the widespread adoption necessary for cooperative collision avoidance to reach its full potential, users must be assured that the risks arising from such systems do not outweigh the benefits; these include threats to the users’ privacy.

3.6.1 Degrees of privacy

Although absolute privacy is not practically feasible, the cooperative collision avoidance system can still provide some privacy protections. The extent of these privacy protections can range from being non-existent to being too restrictive to be useful. More moderate protections include ensuring
that only publicly observable information is exposed and that long-term tracking of any particular vehicle remains as difficult.

3.6.2 Protecting non-publicly-observable information

Even without cooperative technologies, driving on public roads inherently exposes information to nearby observers. For example, casual observers will be able to determine the external appearance, location, and approximate velocity of passing vehicles. A more attentive observer may be able to determine the vehicle’s make and model; license-plate number; the number and appearance of the occupants; part of the path traveled; and through signals required by law, part of the future path. A determined observer can learn even more: the entire route from start to end, the month of the last inspection, and other potentially sensitive information about the occupants. These are arguably examples of acceptable losses of privacy; after all, many people still drive despite these threats.

Generalizing these examples, exposure of publicly observable information arguably may be an acceptable compromise on privacy. One way to ensure that cooperative accident-avoidance preserves this privacy is to ensure that only information that is already in this set is ever given out.

To demonstrate the feasibility of this approach in collision-avoidance applications, the idea that all information that is related to collision-avoidance is already in the set of publicly observable information is first proven. This proof broadly treats all information that can be observed or inferred with an accuracy greater than blind guessing as publicly observable.

Then, a corollary that non-publicly-observable information is irrelevant to collision-avoidance is presented. Finally, guidelines are presented on how to fix collision-avoidance systems that transmit non-publicly-observable information.

The proof

Definition 1. \( \mathbb{A} \) is the set of all possible values for attribute \( A \). For example, if \( A \) is a car’s maximum braking deceleration, then \( \mathbb{A} \) is the set of all non-negative real numbers (with units of acceleration).

Definition 2. Test \( T \) run by observer \( O \) is an algorithm or a device that takes information provided by \( O \) and outputs either 0 or 1. \( T \) may be probabilistic.

Definition 3. Attribute \( A \), whose value is \( a \), is observable to observer \( O \) if and only if (iff) there exists \( a_1 \subset \mathbb{A} \) and test \( T \), which \( O \) can use, that can distinguish whether \( a \) is in \( a_1 \) or not. That is iff

\[
Pr_{a \leftarrow \mathbb{A}}[T \rightarrow 1|a \in a_1] \neq Pr_{a \leftarrow \mathbb{A}}[T \rightarrow 1|a \notin a_1]
\]

Definition 4. Attribute \( A_i \) of car \( C_i \) is publicly observable iff there exist observer \( C_j \) where \( i \neq j \) and \( A_i \) is observable to \( C_j \).
We use this definition for publicly observable because \( C_i \) is presumably unable to prevent \( C_j \)'s observation.

**Definition 5.** Attribute \( A_i \) of car \( C_i \) affects car \( C_j \) iff there exists a attribute \( A_j \) of \( C_j \) which depends on \( A_i \). That is iff there exists \( a_{i,1} \subset A_i \) and \( a_{j,1} \subset A_j \) such that

\[
Pr_{a_i \leftarrow A_i, a_j} [a_j \in a_{j,1} | a_i \in a_{i,1}] \neq Pr_{a_i \leftarrow A_i, a_j} [a_j \notin a_{j,1} | a_i \notin a_{i,1}]
\]

where \( a_j \) is the actual value of \( A_j \).

Note that \( T \) may take some time to generate an output, so though an attribute may be observable, the observation may not be immediately available. Similarly, note that \( A_j \) may be an attribute at a later time than \( A_i \); for example, if \( A_i \) was \( C_i \)'s speed, \( A_j \) may be whether \( C_j \) crashes two seconds later.

**Theorem 1.** If attribute \( A_i \) of car \( C_i \) affects car \( C_j \), where \( i \neq j \), then attribute \( A_i \) is publicly observable.

From Definition 5 let \( A_j \) be the affected property in \( C_j \).

**Assumption 1.** Assume that the value \( a_j \) of the affected attribute \( A_j \) in \( C_j \) is can be known by an observer other than \( C_i \); that is, there exists a test \( T_j \), such that

\[
T_j \rightarrow \begin{cases} 1 & : a_j \in a_{j,1} \\ 0 & : a_j \notin a_{j,1} \end{cases}
\]

where \( a_{j,1} \) is the same as defined in Definition 5.

Note that this assumption will hold if observer \( C_j \) can determine (know) the values of their own attributes.

Since the focus of this paper is collision avoidance, one affected attribute \( A_j \) is likely whether a collision occurred or not. Since collisions tend to be obvious to every nearby observer, including the car(s) involved, the assumption that someone other than \( C_i \) can determine whether a collision occurred is plausible.

**Proof of Theorem 1.** Let the observer be \( O \), where \( O \) is not \( C_i \). Given that \( A_i \) affects car \( C_j \), from Definition 5

\[
Pr[a_j \in a_{j,1} | a_i \in a_{i,1}] \neq Pr[a_j \in a_{j,1} | a_i \notin a_{i,1}]
\]

Applying Assumption 1, substitute \( a_j \in a_{j,1} \) in the equation above with \( T_j \rightarrow 1 \).

\[
Pr[T_j \rightarrow 1 | a_i \in a_{i,1}] \neq Pr[T_j \rightarrow 1 | a_i \notin a_{i,1}]
\]

Since this equation satisfies Definition 3, where \( T_j \) is the test for the attribute \( A_i \), \( A_i \) is observable. Furthermore, from Definition 4 since observer \( O \) is not \( C_i \), to whom attribute \( A_i \) belongs, \( A_i \) is publicly observable. \( \square \)
Given that all relevant information can either be directly observed or indirectly inferred, the usefulness of cooperative systems, like this cooperative collision-avoidance system, comes into question: if this information is publicly known anyway, what is the purpose of deliberately sharing it?

One key benefit becomes apparent when considering the limitations imposed by causality: information inferred from an event can only be obtained after the event begins; however, information cannot be used to alter an event before being known. For example, if an analysis of a rear-ending reveals that the car in the back was following too closely, this information cannot then be used to prevent the same rear-ending.

Another reason is that information that is observable to a powerful adversary may not necessarily be observed by the ordinary neighboring vehicle with whom cooperation is desired. Deliberately sharing information makes cooperation easier.

**The corollary** From Theorem 1.

**Corollary 1.** If $A_i$ of car $C_i$ is not publicly observable, then $\forall j \neq i$, attribute $A_i$ does not affect car $C_j$.

**Guidelines** From Corollary 1 we know that if a collision-avoidance system transmits an attribute that is not publicly observable (as defined in Definition 4), such a transmission is not necessary to avoid collisions and increases the threat to privacy; such an attribute can be omitted.

However, be careful to adhere to Definitions 3 and 4 when determining whether an attribute is publicly observable. Even if test $T$ is extremely difficult to use or if the test only has a slight dependence on the attribute, the attribute remains observable. Note that while attributes that affect whether cars collide are necessarily publicly observable, the converse is not necessarily true.

### 3.6.3 Unlinkability

Although all attributes that may lead to a collision are already publicly observable to very powerful adversaries, additional privacy protections against less powerful adversaries may also be desirable. Furthermore, even against power adversaries, participants may wish to make attacks more difficult in order to deter them.

For example, consider an adversarial employer who wants to monitor the travels of every employee during their off hours. If the adversary were powerful, it can simply follow each member and not need to rely on its targets’ cooperation capabilities, so assume that the adversary does not follow each vehicle. Instead, imagine that the adversarial employer monitors multiple select locations in the city where the workplace is located and wants to record whenever any of the
targets pass those locations; as examples, these locations may be the offices of regulators that are investigating the company. (For a reasonable number of locations, the adversary may be able to inexpensively setup receivers at those locations for the purpose of eavesdropping on transmitted messages.)

One possible defense against this is to ensure that all messages sent by every participant is anonymous and unlinkable in the long term: that is, adversaries can not tell whether two messages that are separated in time by a certain duration were sent by the same sender.

In this example, long-term unlinkability ensures that even if the adversary encounters a targeted vehicle at a later time, the adversary can neither tell whether the vehicle belongs to the targeted group nor tell whether vehicles observed at different locations and at different times are the same as ones observed before.

4 Threat model

For the purpose of analyzing solutions, the following threat model will be used.

4.1 Security goals

Although the design objectives have been presented in section 3, security-specific goals are summarized here for clarity and completeness.

1. Only accept valid data from legitimate senders.

   Legitimate senders are either cooperating vehicles or supporting infrastructure that have been authorized by one or more trusted authorities to participate in the cooperative collision avoidance system. Sybil nodes\footnote{Sybil nodes are fake nodes or participants created in a Sybil attack, in which the attacker attempts to create many fake identities. As an example, a successful Sybil attack against a voting system will allow the attacker to cast multiple votes, each on behalf of a different fake entity.} and former participants with revoked credentials are excluded from the set of legitimate senders.

   Note that valid data are not necessarily correct data. Ideally, each participant in the system will only accept correct data, but this goal does not appear to be practically feasible in a useful way. Instead, valid data are data that pass all available and applicable validation checks, which can include, among others, checks for timeliness, plausibility, and consistency.

2. Be able to positively identify adversaries (traceability and non-repudiation).
As long as security goal 1 is achieved, the ability to positively identify adversaries that are in the set of legitimate senders is sufficient since adversaries that are not legitimate senders will be ignored.

3. Be able to eliminate adversaries from the system (revocation).

4. Maintain the anonymity of behaving vehicles (long-term unlinkability).

4.2 Assumed trust

- The majority of legitimate senders are honest and well-behaved. This is the honest majority assumption [10].

- The Certificate Authority or Certificate Authorities properly issue credentials, revoking credentials, and preserving or revoking anonymity. Note that this assumption is not that the CAs will never issue credentials to adversarial users since that will require the CAs to know a priori which users are adversarial. Instead, this assumption is that the CAs are not adversarial themselves and that adversaries are not allowed to become CAs.

- Every participant knows the time and their clocks are sufficiently synchronized. This assumption is necessary to enable the use of timestamps to improve security. It assumes that the method of synchronizing each participant’s clock is not vulnerable to attack.

- Local devices and locally sensed information are trustworthy. Although the internal systems of vehicles may be vulnerable to tampering and other attacks [8, 12], defenses against these threats are outside the scope of this project. Instead, this project assumes that drivers can trust their own cars.

- Some solutions also assume that each participating vehicle will have a trusted tamperproof module [13, 14, 6, 9, 11]. This project does not make this assumption.

4.3 Possible adversaries

- Cooperative vehicles with valid credentials may be adversarial. Although the the honest majority assumption states that most of these vehicles will behave honestly, some of these may be adversarial. These adversaries can potentially include previously honest vehicles that have not-yet-revoked credentials.
• RSUs can also be adversarial. Examples of these rogue RSUs are malfunctioning RSUs and RSUs that have been subject to tampering.

• External adversaries, or adversaries without valid credentials to participate in the system can be adversarial. This group includes stationary roadside attackers and attackers with revoked credentials.

4.4 Adversarial powers

• The adversary can modify its local data. For example, if the adversary is a vehicle with still-valid credentials, traveling at 30 meters per second, it may trick itself into believing that it is actually traveling at 1 meter per second. This may cause the adversarial vehicle to report the wrong speed to nearby vehicles.

• The adversary may have multiple vehicles or RSUs under its control. However, the honest majority assumption prevents adversaries from controlling the majority of vehicles.

• Adversaries may be mobile. This means that among other attacks, the adversary may simply tail a targeted vehicle.

• Adversaries may temporarily jam communications. However, highly directional receivers may limit the locations where an adversary can successfully jam a communication.

• Adversaries may refuse to transmit or relay messages. They may also do so selectively, relaying some messages but not others.

• Adversaries may eavesdrop on communications. They may also have significantly more sensitive receivers, allowing them to have a longer listening range than the typical vehicle. The directionality of transmissions may limit the locations where an adversary can successfully eavesdrop on a particular message.

• Adversaries may fabricate messages and inject them into the vehicular network. However, validation mechanisms, if used, might enable other vehicles to ignore these injected messages.

• Adversaries may attempt to modify messages in transit. Some opportunities to modify messages include when the adversary relays a message or when the adversary is within range to interfere with the transmission. However, validation mechanisms, if used, may allow the recipient detect the modification.
• The adversary may (selectively) behave honestly.

• The adversary may observe detectable information and does not need to solely rely on the vehicular network for information.

For example, the adversary may use radar to detect the location of vehicles instead of using the cooperative collision avoidance system.

5 Existing VANET security solutions

Many solutions to certain security and privacy problems arising from vehicular networking have already been proposed. Although none of these solutions have been designed for cooperative collision avoidance, some can be adapted for use in cooperative collision avoidance and they offer valuable insight into potential solutions and attacks.

These solutions can be roughly broken into two categories: data verification and sender verification. Sender verification checks to ensure that the claimed sender is indeed the actual sender, that the actual sender is authorized to send the received message, and that the message has not been altered or tampered within transit. Sender verification is typically done by checking MACs or digital signatures.

Data verification attempts to verify that the content of received messages is correct. For example, if the message contains the location of the sender, data verification will attempt to check that the sender is indeed at the stated location. Data verification often relies on sender verification to function properly.

5.1 Data verification

Since the cooperative vehicle will automatically react to emergency situations, as explained in section 2.2.2, the vehicle must make decisions based on the information available to it. If this information is incorrect, the cooperative vehicle may unintentionally place itself and its occupants in greater danger. For this reason, verifying information received from external sources, which are not necessarily trustworthy, is critical.

5.1.1 Position verification via time-of-flight

Figure 5 illustrates one method of securely determining the positions of neighboring vehicles as described in [7]. In this protocol, vehicle \( V \) seeks to determine the position of neighboring vehicles.

\footnote{The description of the protocol presented in this report has been simplified. In its simplified form, it may not adequately defend against attacks, but it illustrates the basics. For a complete description, see [7].}
Figure 5: A simplified version of the protocol described in [7] for securely determining the position of neighboring vehicles is illustrated. Here, $V$ is the vehicle attempting to determine the positions of neighboring vehicles. $X$ is one neighboring vehicle; others may be present, but are not shown.

and vehicle $X$ is one neighboring vehicle; other neighboring vehicles may be present.

$V$ begins by broadcasting an anonymous polling message with a fresh public key $K'_V$ at time $t_V$.

Upon receiving the polling message, each neighboring vehicle notes when the message was received: $t_{VX}$. After waiting a random duration, vehicle $X$ broadcasts a reply with a message encrypted for $V$ that contains $t_{VX}$; $X$’s public key, $K_X$; and $X$’s signature for $t_{VX}$.

This step gives $V$ enough information to calculate the time-of-flight from $V$ to $X$ without revealing this information to other nearby vehicles. This time-of-flight can be used to determine the distance between the sender and receiver. It can also be used as an additional check on the position that $X$ transmits to $V$ in a later step.

After a random wait, $V$ broadcasts a “reveal” message, proving that $V$ sent the original poll. Upon receiving this “reveal” message, $X$ replies to $V$ with a signed then encrypted message containing its location, $p_X$; $t_X$, which is when it sent its original reply; and $t_X$, which is when it received the broadcasted original reply of all other neighboring nodes.

At the end of this exchange, $V$, has the claimed location of every neighboring vehicle and enough information to determine the time-of-flight between every pair of vehicles. $V$ uses the time-of-flight information to verify the position information.

5.1.2 Position and mobility verification using plausibility models

Another method of verifying received messages is by using models of or rules on what events or situations are plausible. In this approach, information is deemed untrustworthy if it depicts
Figure 6: The “appearance margin” and the “minimum distance moved” plausibility models are illustrated with vehicle $V$ performing the checks. “Appearance margin” verification checks to ensure that newly appearing vehicles first appear in the appearance margin: the region near $V$’s transmission range. “Minimum distance moved” verification defends against stationary adversaries that have a limited transmission range, $d_{MDM}/2$; in this approach, $V$ refuses to trust the newly appearing vehicle unless it remains within range as $V$ travels a distance longer than $d_{MDM}$.

implausible situations.

Many checks for plausibility have been presented [14, 3]. These include the appearance margin check and the minimum distance moved check illustrated in Figure 6. Other verification methods include the following:

- Checking that the following characteristics are within certain predefined limits:
  - the reported speed of the vehicle,
  - the frequency at which messages are received,
  - that the relative position is within the communication range, and
  - that the timestamp is appropriately recent.

- Check that vehicles do not overlap.

- Attempt to predict every neighboring vehicle’s movement using Kalman filters. Then compare the reported vehicle movements against the predicted movements to determine plausibility.

5.1.3 Consensus

Information can also be verified by consensus [15, 9]. This approach applies the honest majority assumption to conclude that the consensus is correct.
5.1.4 Position verification by region-specific pseudonyms

Methods for verifying the position of a sender by checking that the position matches the region of the sender’s region-specific pseudonym\[^{3}\] have also been presented \[^{13,2}\]. These methods rely on the assumption that vehicles can only obtain valid pseudonyms for their current region.

However, this assumption is not necessarily true. An attack which enables the adversary to obtain credentials for a different region is presented in section \[^{5.2.2}\].

5.2 Sender verification

Data verification schemes rely heavily on sender verification. For example, the position verification scheme in section \[^{5.1.1}\] would be much easier to defeat if one adversarial vehicle can fake the responses of all neighboring vehicles since the adversary can guarantee that the times of flight between all neighboring vehicles are consistent with their faked positions. Furthermore, attacking consensus-based verification schemes will become trivial if the adversary can perform a Sybil attack to become a majority of the vehicles.

Sender verification is also essential for tracing adversaries and proving that the found adversaries are indeed adversarial. If sender verification is not used or if it is inadequate, the adversary will not need to identify itself when performing attacks; worse, adversarial users can frame honest users through spoofing when transmitting malicious messages.

5.2.1 Unsuitable schemes

Many secure communication protocols exist that allow the intended recipient of a message to verify that the message did indeed originate from the claimed sender. These protocols are commonly used on the Internet and typically use CAs, certificates, public-key cryptographic signatures, and symmetric-key Message Authentication Codes (MACs).

For example, a pair of users can sign messages in a Diffie-Hellman exchange\[^{4}\] to establish a symmetric key\[^{5}\]. The certificate issued to each user by a CA allow both users to verify that the

\[^{3}\]Pseudonyms are temporary credentials, which are typically used in vehicular networking to mask the identity of the user.

\[^{4}\]The Diffie-Hellman exchange is a cryptographic protocol that allows two users to generate a shared secret by openly exchanging messages that do not need to be secret. In this case, the shared secret becomes a symmetric key.

\[^{5}\]Both signatures and MACs can be used to demonstrate that the sender of a message knows some secret. For signatures, this secret is a private key, which is typically known to only the sender; the corresponding public key, which is used to check the signature, can be stored in a certificate, which securely binds the public key to a particular identity. For MACs, this secret is a symmetric key, which is known to both the sender and the receiver. Using MACs tends to be computationally more efficient than using signatures, so many protocols use MACs instead of signatures to protect most of the messages transmitted.
exchange is done with the intended user and not an adversary. Using the shared symmetric key, each of the two users can tag messages with MACs. They can also verify the MAC using the symmetric key, allowing them to check that received messages were sent by the other party.

However, this scheme is not suitable for mainly two reasons. First, although MACs are traceable in this scheme, they do not offer non-repudiation; the recipient cannot prove to another party that the other participant in the Diffie-Hellman exchange is indeed the other sender. Since both the sender and the receiver have the symmetric key for the MAC and since the same symmetric key used to verify a MAC can be used to generate a MAC, a third party cannot conclude which of the claimed sender or receiver actually generated the message and MAC.

Non-repudiation can be achieved by using signatures instead of MAC to authenticate messages. If a certificate belonging to the claimed sender verifies the signature, then the claimed sender is responsible for endorsing the signed message.

The second reason why this scheme is not suitable for cooperative collision avoidance is that this scheme gives honest users neither anonymity nor unlinkability. Anonymity is not available since the certificate identifies the user. Unlinkability is not available since the same certificate is used for long durations; an adversary can tell that messages authenticated using the same certificate came from the same sender.

Many VANET-specific approaches to sender verification have been presented. However, many of these solutions are unsuitable for cooperative collision avoidance because they are either susceptible to Sybil attacks [2, 5, 9] or give one or more entities the ability to easily perform long-term tracking of vehicles [2, 6].

5.2.2 Using group signatures

Two schemes that claim to prevent Sybil attacks (security goal in section 4.1) and maintain unlinkability (security goal 4) were found [13, 15]. Both schemes use group signatures, which is used to provide revocable and conditional anonymity and unlinkability.

Group signatures schemes allow group members to sign messages on behalf of a group. Recipients of these signed messages can verify that it was signed by a member of the group, but only the group manager is able to identify who generated the signature. Both [13] and [15] use modified group signature schemes, which allow the recipient to determine whether two of the same messages or requests were generated by the same group user key.

Upon receiving a message with a valid MAC, if the recipient did not generate the message and if the recipient did not disclose the key needed to generate the MAC, the recipient can conclude that the other user with which the receiver performed the Diffie-Hellman exchange either generated the message or authorized someone else to generate the message by sharing the key. In both cases, the other participant of the Diffie-Hellman exchange is known to be at least partially responsible for generating the message. Using the other participant’s certificate, which was presented to verify signatures during the Diffie-Hellman exchange, the recipient can trace the message back to the other participant’s identity.
Temporary Anonymous Certified Keys  The TACK system, presented in [13], consists of RSUs, which act as Regional Certificate Authorities (RAs); a central CA; and the users, which are vehicles on the vehicular network. Each user is assigned a long-term group user key by the central CA.

Using the group user key, the user can request a temporary pseudonym from the local RA by first generating a new public and private key pair. After generating a new key pair, the user sends the new public key to the RA in a request signed with the group user key. The modified group signature scheme allows the RA to determine if two requests within the same time period were generated using the same group user key.

After checking that no previous request has been made in the same time period and after checking a Revocation List (RL) to ensure that the group user key has not been revoked, the RA stores the request, and returns to the user a short-term certificate with the new public key signed by the RA. This short-term certificate expires; after the expiration, the user can request a new short-term certificate. The short-term certificate also specifies the region in which the certificate is valid.

To communicate with other vehicles, the user signs its messages using the private key of its temporary pseudonym. Other vehicles can verify signature against the short-term certificate; checking the signature on the certificate against the RA’s public key; and checking the RA’s certificate, which grants the RA authority, against the central CA’s public key.

Assuming that the CA is trusted to only authorize well-behaved RAs and to only issue one group user key to each user, Sybil protection against individual adversaries is provided since individual adversaries can not obtain more than one valid temporary pseudonym in any given region.

Since the user switches its temporary pseudonym when the short-term certificate expires or when the user enters a new region, other vehicles and external adversaries cannot link the users message over long periods of time. Furthermore, since requests for temporary pseudonyms are only linkable if they are made within the same time period, the user is also unlinkable to the RA in the long-term. Presumably, the trusted central CA can link messages sent by the user if it is able to listen to the requests made to the RAs, but this too can be prevented by encrypting the temporary pseudonym requests and replies.

To trace adversaries, the recipient of the malicious signed message, the RA and the central CA need to cooperate. The signed message proves that the owner of the temporary pseudonym sent the message; the request stored by the RA proves the owner of the temporary pseudonym generated a particular group signature; and the CA can revoke the anonymity of a group signature.

---

7 However, as explained in section 5.2.2, Sybil attacks become possible if multiple adversaries collaborate maliciously.

8 Encrypting the TACK requests and replies is not mentioned in [13], but it may be done by using the RA’s public key to encrypt request to the RSU. Replies to the user can be encrypted using the user’s temporary public key, which is included in the user’s request.
Figure 7: Sender authentication schemes in which Regional Certificate Authorities operate independently in multiple regions are susceptible to Sybil attacks performed by multiple collaborating adversaries. In this figure, \( m \) adversaries exist and are spread out across \( n \) regions. By using other adversaries as proxies, each adversary request pseudonyms from every RA. Adversaries can then exchange pseudonyms to collect multiple pseudonyms for their current regions, thus successfully performing a Sybil attack.

**Message-Linkable Group Signatures** As described in [15], Message-Linkable Group Signatures (MLGSs) are another way to apply group signatures. This group signature scheme is modified so that signatures can only be linked if they are for the same message and signed by the same group user key. As with plain group signatures, the group manager (or CA) can trace the user that generated a particular signature and revoke the user’s anonymity. Unlike TACKs, Regional Certificate Authorities are not necessary, which may make deployment significantly easier. Temporary pseudonyms are not used in MLGSs.

MLGS offers Sybil protection as long as the message remains the same. To perform data verification check that require a consensus, participants can sign messages with which they agree, where different messages will be voted upon separately.

### 5.3 Notable weaknesses

Among the existing proposed solutions, multiple exploitable vulnerabilities were found. These vulnerabilities offer valuable insights for the development of future solutions.

#### 5.3.1 Executing Sybil attacks across multiple regions

Multiple sender verification schemes, including TACKs, use independently operating Regional Certificate Authorities (RA) to assign pseudonyms to vehicles [13 2]. Since these RAs do not
check if a vehicle that is requesting a pseudonym already has a valid pseudonym from another region, each vehicle can obtain pseudonyms from multiple RA, so that they are all simultaneously valid. The ability to obtain a new pseudonym despite already having a valid one in another region is an intentional feature of these sender verification schemes; a vehicle crossing between regions will need a new pseudonym for the new region regardless of whether the pseudonym in the old region has expired.

Limited communication range may prevent an adversary from requesting certificates across too many regions, but collaborating adversaries in different regions can serve as proxies for each other, using wide-area networks to relay messages across regions. Still, well-behaved users will ignore messages signed by pseudonyms from other regions so an adversary with pseudonyms from multiple regions can not successfully perform a Sybil attack yet.

However, if collaborating adversaries exchange these pseudonyms over a wide-area network, they can successfully perform Sybil attacks as shown in Figure 7.

For example, assuming that each adversary $M_1, M_2, ..., M_m$ is within range of one unique RA, each adversary can get $m$ pseudonyms from $m$ different regions. Then, each adversary can trade pseudonyms with each other: adversary $M_1$ gets the pseudonyms for $M_2, M_3, ..., M_m$ for $M_1$’s correct region; $M_2$ gets the pseudonyms for $M_1, M_3, M_4, ..., M_m$; and so on. After the exchange, each of the $m$ adversaries will have $m$ unique pseudonyms for its correct region. Thus, they have successfully performed a Sybil attack, allowing each adversary to multiply its influence $m$-fold. In total, this attack allows the adversaries to gain $m(m - 1)$ extra pseudonyms.

With this many Sybil nodes, the adversaries can easily overwhelm honest vehicles in any scheme that relies on consensus. For this reason, although the use of pseudonyms, as done in TACKs, is more computationally efficient than using group signatures for all messages, as done with MLGSs [5], MLGS is preferred over TACKs.

6 Remaining problems

Even after applying the solutions presented in section 5, several problems and challenges remain.

6.1 Selectively dropping or jamming messages

As shown in figure 8, the adversary’s ability to selectively drop messages can result in incorrect decisions. Since the adversary has the ability to temporarily jam communications, it can selectively jam messages if it can correctly guess when the messages will be transmitted and if it is within range of the receiver.

This threat can be mitigated by using highly directional and line-of-sight communications,
Figure 8: If in a voting or consensus system, the adversary is able to selectively drop packets, the adversary is able to change the outcome of the vote even if each vote is protected by a signature.

Figure 9: Shown are some of the advantages of directional communications in vehicular networks. For example, adversary $M_1$ cannot jam $V_4$’s communication to $V_1$ because $M_1$ is not in the necessary position relative to $V_1$. Well-behaving vehicles may also route around adversaries that drop messages.
such as VLC and possibly Dedicated Short-Range Communication (DSRC). As illustrated in figure 9, highly directional receivers for line-of-sight communications can greatly restrict the locations from which an adversary can jam communications, thus limiting the adversary’s ability to selectively drop messages.

Although an adversary in the necessary position can still drop messages, by either jamming communications or refusing to relay messages, this remaining threat can be further mitigated by using multiple routes to send messages redundantly, as illustrated in figure 9; if at least one of these routes are not interrupted by the adversary, the message can arrive to its intended recipient successfully.

6.2 Initial deployment

Difficulties can arise from partial deployment of the cooperative collision avoidance system. These include the low probability of encountering other cooperating vehicles; the inability to cooperate with older vehicles, which will initially be the majority; and the difficulty of being unlinkable when the ability to cooperate is a distinguishing feature.

6.2.1 Non-participating vehicles

As mentioned in section 3.1.2, some vehicles will not cooperate on the vehicular network. Unfortunately, none of the approaches discussed in section 5.1 take this possibility into account. Since communication-based methods cannot detect vehicles that do not communicate, additional sensors are necessary.

Section 7 investigates approaches to apply information from these sensors to improve the security and reliability of the cooperative collision avoidance system.

6.2.2 Unlinkability as sole cooperative vehicle

When the cooperative collision avoidance system is first deployed, the distribution of cooperative vehicles will be very sparse. Unfortunately, this makes the ability to cooperate become a distinguishing feature. Having this distinguishing feature, the vehicle will likely stand-out, making it difficult for the vehicle to be unlinkable.

One possible solution is to disable the cooperation capability until the number of deployed cooperating vehicles reaches a certain threshold. However, this solution will prevent the vehicle from benefiting from cooperative collision avoidance until the threshold is reached.
6.2.3 Significant probability of failure

Although the data verification techniques presented make accepting incorrect information much less likely, a significant probability of failure remains. For example, if honest and adversarial vehicles were randomly distributed along all roads, adversarial vehicles may locally outnumber honest users despite having the honest majority assumption remain globally true.

This scenario is especially true if the distribution of cooperating vehicles is sparse; in this case, any location with an adversary is likely to have at least as many adversaries as there are honest vehicles.

Locally violating the honest majority assumption weakens many of the data verification schemes. Since the information involved may be used to make life-or-death decisions, this lingering probability of failure may be unacceptable.

7 New security solutions for cooperative collision avoidance

Fortunately, other techniques can help to ensure the integrity of the gathered information.

7.1 Improving data verification with sensing and directional line-of-sight communications

As established in section [6.2.1] sensors (such as radar, LIDAR, or sonar) are required. However, none of the data verification schemes presented so far use information from these sensors to verify information from other vehicles.

Assuming that these sensors can reliably determine the relative positions of nearby vehicles within the line of sight, these sensors can be used to both gather and verify information on the positions, velocities, and acceleration of nearby communication neighbors. Specifically, if directional and line-of-sight communications is used, the sensors should be able to reliably verify information about the immediate movement of all neighbors that are one communication hop away.

Directional line-of-sight communication can also be used to screen information about nearby vehicles. For example, if a message that has not been relayed arrives from one direction, but the message claims the transmitter is in a different relative direction, then the message is most likely wrong. As another example, if a vehicle claims to be in a position that should be within line of sight, but the arriving message was relayed through another vehicle, the claimed position is suspect.¹

¹However, there are reasons why this situation may legitimately occur. For example, if the receiver facing the
These techniques are especially useful when not enough cooperating vehicles are nearby to reasonably ensure that the local majority is honest and well-behaved.

7.2 Additional benefits of directional line-of-sight communications

Including directional line-of-sight communications also has other benefits. Not only does it limit the locations where adversaries can jam communications, it also limits the locations from where adversaries can launch other attacks, such as injecting messages or eavesdropping on communications. These limitations on the adversary increase the costs of performing attacks, which may reduce their likelihood.

Still, even with directional line-of-sight communications, undirected non-line-of-sight communications remain useful for cooperative collision avoidance. Not only can non-line-of-sight communications serve as a backup when line-of-sight communications are obstructed, they are necessary for some data verification schemes, such as the position verification via time-of-flight scheme described in section 5.1.1.

7.3 Short-term linkability

Although long-term unlinkability is required for privacy, short-term linkability is highly desirable. Without short-term linkability each position update will be indistinguishable from the arrival of a new vehicle; in order for a car to update the existing information that it has on a neighboring vehicle, it must determine which vehicle to update the information for.

Not only will the lack of short-term linkability break data verification schemes such as the appearance margin and the minimum distance moved checks, it will either result in vehicles virtually disappearing whenever an update is lost or in perceived duplicates of the same vehicle.

By design, MLGSs do not offer any linkability across different messages; although TACKs offers some short-term linkability, the short-term linkability of TACKs breaks across pseudonym changes. Fortunately, short-term linkability can be added to both schemes without destroying long-term unlinkability.

7.3.1 Short-term linkability for TACKs

For TACKs, each message needs to identify the previous pseudonym. This can be done by simply including a hash of the old pseudonym’s public key. However, since this link is formed by honest transmitter is jammed by an adversary, the honest transmitter may need to transmit the message along a less direct route.
performing a public function on a publicly known information, an adversary can also perform this operation to hijack the link, causing nearby vehicles to believe that the adversary is hijacked user with an updated pseudonym.

This hijacking can be prevented by instead signing each message with the private keys of the previous pseudonyms (in addition to signing the message using the private key of the current pseudonym as specified by the TACK scheme). The user can also sign every new certificate with the private key of the old pseudonym. Senders can make matching old and new pseudonyms at the receiver easier by specifically identifying the previous pseudonym; this can be done by including the previous certificate in messages.

Alternatively, the RA can identify the old pseudonym in the certificate for the new pseudonym.

These procedures ensure that linkability remains intact to recipients who continue to be in contact with the linked vehicle, but that linkability breaks for anyone who loses contact with the linked vehicle for longer than the useful life of a pseudonym.

7.3.2 Short-term linkability for MLGSs

Similarly, for MLGSs, each vehicle can periodically generate a separate, fresh public and private key pair; this period can be adjusted as necessary to achieve the desired level of privacy and tolerance for missed messages. To maintain linkability, each message includes the current public key and is signed using the previous private key. (For sender verification purposes, the whole message including the information for short-term linkability should still be signed using the MLGS scheme.) To make linking easier for the recipient, the sender can also include the previous public key or a hash of it.

7.3.3 Other methods to achieve short-term linkability

Aside from having RAs identify each vehicle’s previous pseudonym in each new certificate, the methods of providing short-term linkability in sections 7.3.1 and 7.3.2 rely on self-reporting; users can deliberately break this linkability by changing the key pairs twice without sending any messages after the first change. Although deliberately breaking short-term linkability should be considered improper behavior, users may do so in an attempt to guard their privacy against a perceived threat. Adversaries may also attempt to break short-term linkability for less benign purposes.

To guard against breaks in short-term linkability, neighboring vehicles may use the fact that positions and speeds of vehicles change gradually over time: vehicles cannot teleport and momentum prevents vehicles from instantly changing speeds. Hence, updates on the movement of a vehicle can be linked to specific known vehicles by matching positions and velocities.

Matching the current position against projected positions may be more accurate than against previous positions.
Information from sensors can further improve linking by augmenting the information available through received messages. These alternate methods can also be used to verify the linking information provided through messages to defend against adversaries that collaborate to swap the key pairs which are used to facilitate short-term linkability.

8 New protocol for detecting potential collisions

Although data verification and sender verification are important aspects of securing cooperative collision avoidance systems, what information is actually exchanged between vehicles is also an important design issue. If too much information is provided, data transmission times will increase, delaying the automatic reaction time of participating vehicles. Providing too much information may also make anonymity and long-term unlinkability impossible; as an extreme example, if the license plate number is included, long-term unlinkability will not be possible since the license plate number does not tend to change frequently.

On the other hand, providing too little information can reduce the usefulness of the communications to the point where solely relying on the sensors becomes a better solution.

The protocol presented in this report is one of two protocols envisioned for cooperative collision avoidance. The purpose of this first protocol is to help cooperating vehicles detect impending collisions and to determine communication routes for the second protocol in case it becomes necessary.

This first protocol, the “collision detection” protocol, aims to help cooperating vehicles locate each other; to share information on the velocity and acceleration (treating the Earth as a stationary reference) of each vehicle that is more accurate than what radar, LIDAR, or sonar sensors can measure; to provide this information sooner than those ranging sensors can acquire it; to extend each cooperating vehicle’s situational awareness beyond what local ranging sensor can detect, and to prepare for the second protocol in case it becomes necessary.

The second protocol executes when a possible impending collision is detected. The purpose of this second protocol, the “collision avoidance” protocol, is to determine how the cooperating vehicles involved should maneuver in order to avoid or mitigate the collision.

For the second protocol, an up-to-date map indicating where and how cars can travel is assumed to be available to each vehicle.

Given previous position $\vec{X}$, previous velocity $\vec{V}$, and the elapsed time $\Delta t$, the projected position is $\vec{X} + \vec{V} \Delta t$. If the previous acceleration and/or jerk are known, they can also be incorporated into the projected position.
8.1 Considerations

This protocol has been designed assuming that the ranging sensor(s) (such as radar, sonar, or LIDAR) are much more accurate and precise at measuring relative positions than Global Positioning System (GPS) or other technologies can determine absolute position. This protocol also assumes that each vehicle can determine its own absolute velocity and acceleration at least as accurately and precisely as neighboring vehicles can through ranging sensors. Although relative or absolute elevation is also important in determining whether potential collisions are likely to happen, they have not been considered for this protocol yet; as it is now, this protocol does not account for overpasses or underpasses, where vehicles can avoid collision by being at different elevations.

8.2 Relative versus absolute positions

8.2.1 In favor of relative position

Due to the better accuracy and precision of measurements from ranging sensors, vehicles are able to determine their relative positions to each other better than they can determine their absolute positions.

As illustrated in figure 10, if relative position is not specified, errors in absolute positions can result in errors in relative position, and more importantly, errors in relative direction. These errors in relative position arising from errors in absolute position can make a car to the west of the observer appear as a car to the east. If the observed car is traveling east, then the approaching observed car can be incorrectly perceived as a departing vehicle, potentially preventing the observing car from detecting the collision before it happens.
Figure 11: A position translation attack is shown. Due to perspective, relative positions are only correct to their intended viewer or recipient. In this attack adversary $M_2$ forwards to accomplice $M_1$ a message from $V_2$ indicating the position of $V_2$ relative to $M_2$; $M_1$ relays this message to $V_1$ as if $M_1$ was the intended recipient; if the absolute position of $V_2$ is not included in the message, this tricks $V_1$ into believing that $V_2$ is approaching the upcoming intersection (like the scenario depicted in figure 3) even though $V_2$ is actually on a different road.

The resulting relative position errors arising from absolute position errors also complicate the matching of information from the ranging sensors to information from the vehicular network. While a vehicle that the ranging sensors detect to the east may be the same one that sends an absolute position that appears to be to the west, these differences are difficult to automatically resolve. If the observing vehicle cannot resolve these differences, then the information from sensors and the information from the vehicular network will conflict rather than complement each other. This conflict will in turn reduce the utility of the collision detection protocol if the vehicles consider the ranging sensors to be more reliable.

8.2.2 In favor of absolute position

However, without absolute positions, adversaries can easily make vehicles appear to be in positions other than their actual ones. Relative positions are only correct to their intended recipient; to different vehicles the relative position of the same car will appear different due to changes in perspective.

However, if adversaries are able to relay messages or replay messages in different locations, which they are able to do by injecting a copy of the messages that they wish to replay, then the adversary can direct messages intended for the adversary to other vehicles. Since the final recipient is not the intended recipient, the relative position is incorrect to the final recipient. This attack is illustrated in figure 11.

Even worse, since the adversary does not need to modify the content of this message to
Figure 12: The content of a movement reporting message is shown. This message describes the position and instantaneous movement of the sender: $x$ specifies the relative position from the receiver, $X$ is the absolute position, $r_{\text{error}}$ is the radius of error of the absolute position measurement, $V$ is the absolute velocity, and $A$ is the absolute acceleration. Each of vectors $x$, $X$, $V$, and $A$ are expressed in north and east components.

Figure 13: The content of an observation message.

perform this attack, the original group signature remains intact. As a result, the recipient may not realize that the actual adversary is involved and instead believe that the original sender is malicious. Hence, without absolute position, adversaries are able to frame other vehicles when performing attacks.

To reduce the likelihood of such an attack succeeding, the absolute position is included with the more useful relative position. However, this position translation attack is still possible when the translation is small.

8.3 Messages

The messages used in the collision detection protocol are illustrated in figures 12, 13, and 14. The “head” and “tail” of the messages, which are not shown in detail, include the type of message, the length of the message, a timestamp, the certificate used for sender verification, short-term linkability information, and a group signature.

8.3.1 Movement reporting message

Movement reporting messages, which are illustrated in figure 12, allow vehicles to self-report information about their movements to neighboring vehicles. This information includes both the relative position, $x$, to the intended first-hop receiver and the absolute position, $X$, for the reasons explained

Figure 14: The content of a relay message, where $m_i$ for integer $i$ from 0 to $n_{\text{entries}} - 1$ are each a message.
in section 8.2. To avoid the ambiguity of whether \( x \) is the position of the receiver relative to the sender or the position of the sender relative to the receiver, \( x \) will be the position of the sender relative to the receiver; for Cartesian coordinates, using north and east as the components, the perspective can be switched by negating each component. The expected precision of the absolute position is specified with radius \( r_{\text{error}} \).

The velocity, \( V \), and acceleration, \( A \), are also included to allow recipients to predict the future positions of the sending vehicle. Since vehicles are assumed to be able to determine their velocity and acceleration relative to the Earth more accurately than relative to other vehicles, \( V \) and \( A \) are both relative to the Earth.

For simplicity, \( x \), \( V \), and \( A \) are expressed in Cartesian coordinates, each with a north and an east component. To express directions to the south or the west, these values can be negated. The units for \( x \), \( r \), \( V \), and \( A \) used is chosen to be from the MKS (meter, kilogram, second) system of units to avoid the need to specify units. \( X \) is specified as coordinates in degrees to maximize compatibility with GPS receivers.

### 8.3.2 Observation messages

Observation messages, illustrated in figure 13, allow vehicles to share observations of neighboring vehicles, such as what may be gathered from the ranging sensors. Observations of multiple vehicles can be concatenated into one message to reduce the overhead from sending the “head” and “tail” portions of the message; \( n_{\text{entries}} \) specifies the number of observations in one message.

\( x \) is the position of the observed vehicle relative to the observer. \( V \) and \( A \) remain the same as in movement update messages as described in section 8.3.1 but describe the observed vehicle instead of the sender.

The relative position of the observed vehicle to the receiver can be determined by summing the vector for the position of the sender relative to the receiver and the vector for the position of the observed vehicle relative to the sender.

### 8.3.3 Relay messages

Messages can be relayed to other vehicles by enclosing the entire message to be relayed, including the “head” and “tail”, within a relay message as illustrated in figure 14. Movement reporting message, observation messages, and even other relay messages can be included in relay message. As with observations in observation messages, multiple messages can be concatenated to be sent in one relay message. \( n_{\text{entries}} \) specifies the number of other messages included in the relay message; for this count, an included relay message counts as only one message even if the included relay message contains many other messages.

33
The position of a vehicle described in a relayed message can be determined relative to the final receiver by summing the vectors of relative positions of every vehicle along the way.

The nested structure of relay messages also indicates a reverse path from the recipient to the sender of a nested message. As long as the vehicles along the way have not moved enough to break a link, this reverse path can be used to send replies in other protocols.

9 Conclusion

So far, this project has investigated the threats that face participants of a cooperative collision avoidance system and has determined requirements for such a system to defend against these threats. Along the way, this project has also investigated the other requirements for cooperative collision avoidance, including what needs to be communicated among the vehicles, what equipment each vehicle needs, and what infrastructure needs to be in place to support the system. These results outline important aspects of cooperative collision avoidance and can guide not only this project, but future research into cooperative accident avoidance.

Furthermore, many security solutions for other VANET applications were evaluated in the context of cooperative collision avoidance. Not only did this survey identify solutions that can be applied to cooperative collision avoidance, it also highlights potential pitfalls that the system can encounter. This investigation yielded various methods of verifying shared information and identified group signatures as a solution for ensuring that adversaries can be caught while protecting the privacy of well-behaving vehicles. These advances will help to deter adversaries and promote participation when the system is finally deployed.

Several new solutions for protecting or facilitating cooperative collision avoidance have also been devised. These solutions include new approaches to verify received information and a protocol to support cooperation that either incorporates or is compatible with most of the previously mentioned security solutions.

Although this project makes many advances towards ensuring the reliability of a cooperative collision avoidance system, more future work remains to be done before the system is robust enough for the life-or-death situations that it will need to handle. This future work includes formally evaluating the new solutions presented in this project; accounting for factors such as varying vehicle sizes and overpasses; developing a protocol for negotiating cooperative maneuvers; investigating methods to quickly and reliably notify others about malicious vehicles; and enabling vehicles to use these tools intelligently.

Much work also remains to expand the scope of this cooperation to improve transportation efficiency and to provide overall accident avoidance. Eventually, cooperation may lead to accident-free transportation with both the efficiency of mass transit and the flexibility of personal vehicles.
10 Acronyms and abbreviations

CA Certificate Authority
DSRC Dedicated Short-Range Communication
ECU Electronic Control Unit
GPS Global Positioning System
LIDAR Light Detection and Ranging
MAC Message Authentication Code
MLGS Message-Linkable Group Signature
RA Regional Certificate Authority
RF Radio-Frequency
RL Revocation List
RSU Roadside Unit
TACK Temporary Anonymous Certified Key
VANET Vehicular Ad-hoc Network
VLC Visible-Light Communication

11 Appendix

11.1 References for the poster

Below is a mapping between the reference numbers used in the poster (“Secure Cooperative Accident Avoidance for Vehicles” presented by Jimmy C. Chau on May 6, 2011 for the MS Project Symposium at Boston University) and the reference numbers used in this paper.
<table>
<thead>
<tr>
<th>Reference number on poster</th>
<th>See reference number</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>14</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>13</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td>11</td>
<td>9</td>
</tr>
<tr>
<td>12</td>
<td>5</td>
</tr>
</tbody>
</table>

References


