# A Trust-based Message Evaluation and Propagation Framework in Vehicular Ad-Hoc Networks

by

Chen Chen

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathmatics
in
Computer Science

Waterloo, Ontario, Canada, 2009

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

## Abstract

In this paper, we propose a trust-based message propagation and evaluation framework to support the effective evaluation of information sent by peers and the immediate control of false information in a VANET. More specifically, our trust-based message propagation collects peers trust opinions about a message sent by a peer (message sender) during the propagation of the message. We improve on an existing cluster-based data routing mechanism by employing a secure and efficient identity-based aggregation scheme for the aggregation and propagation of the senders message and the trust opinions. These trust opinions weighted by the trustworthiness of the peers modeled using a combination of role-based and experience-based trust metrics are used by cluster leaders to compute a ma jority opinion about the senders message, in order to proactively detect false information. Malicious messages are dropped and controlled to a local minimum without further affecting other peers. Our trust-based message evaluation allows each peer to evaluate the trustworthiness of the message by also taking into account other peers trust opinions about the message and the peer-to-peer trust of these peers. The result of the evaluation derives an effective action decision for the peer.

We evaluate our framework in simulations of real life traffic scenarios by employing real maps with vehicle entities following traffic rules and road limits. Some entities involved in the simulations are possibly malicious and may send false information to mislead others or spread spam messages to jam the network. Experimental results demonstrate that our framework signicantly improves network scalability by reducing the utilization of wireless bandwidth caused by a large number of malicious messages. Our system is also demonstrated to be effective in mitigating against malicious messages and protecting peers from being affected. Thus, our framework is particularly valuable in the deployment of VANETs by achieving a high level of scalability and effectiveness.

## Acknowledgements

First and foremost, I would like to thank my supervisor, Professor Pin-Han Ho. This thesis would not have been possible without his guidance and patience over the last two years. Second, I much appreciate the comments from Professor Sagar Naik and Professor Robin Cohen, who have devoted much of their time to thesis reading.

Thanks to all the people who made this possible, especially to Jie Zhang, Issam Aib, Qi Zhang, Chenxi Zhang, Xiaodong Lin, Carol Fung, and Jiajia Han, for their insights, discussions and valuable comments.

# Dedication

This is dedicated to my grandma.

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

With the advance and wide deployment of wireless communication technologies, vehicle manufactures and research academia are heavily engaged in the blueprint of future vehicular ad-hoc networks. Tremendous efforts have been spent on their design and analysis that involve a wide range of research objectives, such as radio frequency spectrum design [1], global position system [2], group communication [3], information dissemination [4], security [5], and privacy [6]. There also emerge a rich variety of applications, including life-critical ones such as safety message sharing [7], cooperative collision avoidance [8], and secure crash reporting [9], as well as infotainment applications such as the traffic view system [10], content distribution [11], cooperative downloading [12], Wi-Fi-based vehicular Internet [13], and car-based mobile sensor computing system [14], etc.

## 1.1 Motivation

There are two types of communications in vehicular networks, i.e. vehicle-to-infrastructure (V2I) where messages are passed between vehicles and infrastructures (Road Side Units) and vehicle-to-vehicle (V2V) where messages are propagated among peers. The latter V2V communication is challenging and involves much focus on networking routing research where we realize that there are two problems within current research, as discussed follow.

First of all, although previous efforts in the field of network routings [15] have been devoted to generally ensuring a reliable message delivery, there lacks an effective message evaluation approach which is appreciated because it is useful to quantitatively present the quality of data so as to differentiate and prioritize data, and provide suggestions among a set of messages. However, data evaluation may not fully rely on the assumption of a pervasively available infrastructure such as an online central authority or road side units, due to concerns of broadness of the real vehicle environment and high deployment and communication cost.

Second, little concern so far has been focused on malicious data control in the application layer. Previous endeavors in [16, 17, 18] focus on the eviction of misbehaving peers via certification revocation through which malicious peers will be identified and restricted from further hampering the network. However, two shortcomings are evident in these approaches. On one hand, the mitigation against maliciousness is entity-oriented rather than data-oriented. In other words, they assume that the quality of data absolutely depends on the peer's honesty, which ignores the variety of data types and might be problematic under life-critical scenarios, e.g. a misbehaving peer with its revoked certificate is not capable to send S.O.S. signals upon real car accidents. On the other hand, the methodology taken towards the malicious data control is *reactive*. Specifically, it takes a considerable time for the central authority to distribute an up-to-date revocation list before malicious peers can be timely identified. A *proactive* approach is much appreciated in that upon detection of malicious data it can be immediately controlled to minimize its further negative effect on other peers. A model on detecting malicious data in vehicular networks is proposed in [19], which is the one of few research papers that we have found in our context. It features an assumed model in each vehicle that acquires the global knowledge of the network and solely evaluates the validity of data. Considering that such a model with global knowledge may not be feasible in practice, we propose that evaluation should be done in a distributed and collaborative fashion.

The question becomes that *is it possible to have a particular data dissemination scheme in vehicular networks that offers a quantitative approach to data evaluation, and at the same time detects and controls malicious data, proactively and collaboratively*? In this thesis, we attempt to answer this question using a trust-based method.

## 1.2 Contribution

Our contributions to the networking research in vehicular networks are stated as follows.

- *Improved system effectiveness.* We propose a trust-based message evaluation model that collects instant feedback during message propagation, in a distributed and collaborative manner. From a set of trust metrics such as trust opinions, experience-based and role-based trust, we further model the message quality as data trustworthiness which serves as a quantitative approach to data evaluation. Experiments demonstrate that our system is capable of proactively and effectively coping with the pervasive existence of malicious messages.

- *Improved network scalability.* A trust-based relay control model is presented in our thesis which proactively detects malicious messages during dissemination. Malicious data is therefore dropped and controlled to a local minimum without further affecting other peers. As demonstrated in our experiments, network scalability is improved in that less malicious traffic is involved.

- *Efficient and secure data aggregation.* We employ an identity-based aggregation scheme that aggregates trust opinions in an efficient and secure fashion. Trust is therefore effectively established through messaging among multiple peers in vehicular networks where the trust relationship is believed to be difficult to build due to great dynamics and ephemeral connectivity among peers.

## 1.3 Thesis Organization

The rest of the thesis is organized as follows. We demonstrate a trust-based message evaluation and propagation model in Chapter 2, and illustrate a secure, efficient and scalable trust opinion aggregation scheme in Chapter 3. Experimental simulation and analysis are conducted to evaluate our work in Chapter 4. After that, we survey some related work in Chapter 5. Finally, conclusions and future research work of the thesis are presented in Chapter 6.

# Chapter 2

# Trust-based Message Evaluation and Propagation

In this chapter, we present a trust-based message evaluation and propagation model in vehicular networks. We aim to build a data dissemination framework where data trustworthiness and trust for the vehicle entities work together to serve as a criterion of message evaluation and propagation.

This chapter is organized as follows. A quick overview of our trust-based system is first illustrated in Section 2.1, where we introduce the data design and system components. We demonstrate our trust-based message evaluation model in Section 2.2, and trust-based message propagation model in Section 2.4. The peer-to-peer trust module is presented in Section 2.3. We further discuss the handling of the presence of authority in Section 2.5, and finally summarize this chapter in Section 2.6.

## 2.1 System Overview

The basic idea of our proposed system is to evaluate and disseminate a message based on its quality. We design our system in a way that messages can be evaluated in a distributed and collaborative fashion. At the same time, the dissemination distance of a particular message is largely dependent on its quality, so that messages of good quality propagate to the furthest distance while malicious data, such as spams, are controlled to a local minimum.

We model the message quality using a trust-based approach. In other words, the quality of a message is mapped to a trustworthiness value, which can be computed from a collection of distributed feedback from other peers in the network. Specifically, during the message propagation, the peer who receives the message can instantly provide feedback, namely, a *trust opinion* generated from an equipped *analysis module*. A set of trust opinions are appended to the message during message propagation. For those who receive the message, their *action module* may decide to trust or distrust it by computing its trustworthiness from an aggregated list of trust opinions. Details are available in Section 2.2.

Apart from the trust modeling on data quality, we further model the behavior of vehicle entities using a *peer-to-peer trust* approach proposed by Minhas et al. [20] – each peer's behavior is mapped to a trust value which is dependent on the quality of messages and trust opinions from the peer. We present this peer-to-peer trust module in Section 2.3.

Figure 2.1 gives an example of message propagation under the scenario of a six-laned highway. Sender $S$ broadcasts a message to its neighborhood. Each neighbor sends back a trust opinion. These trust opinions are collected and merged into the sender's message by a relaying vehicle. Upon the reception of a relayed message, the relayer broadcasts it to its neighbors, collects their trust opinions if there is any, merges them into the message, and sends it to the next hop[1] in a recursive way. Details of message propagation are discussed in Section 2.4.

[1] Hop is a network routing term used for the next gateway (peer) to which packets should be forwarded along the path to its final destination.

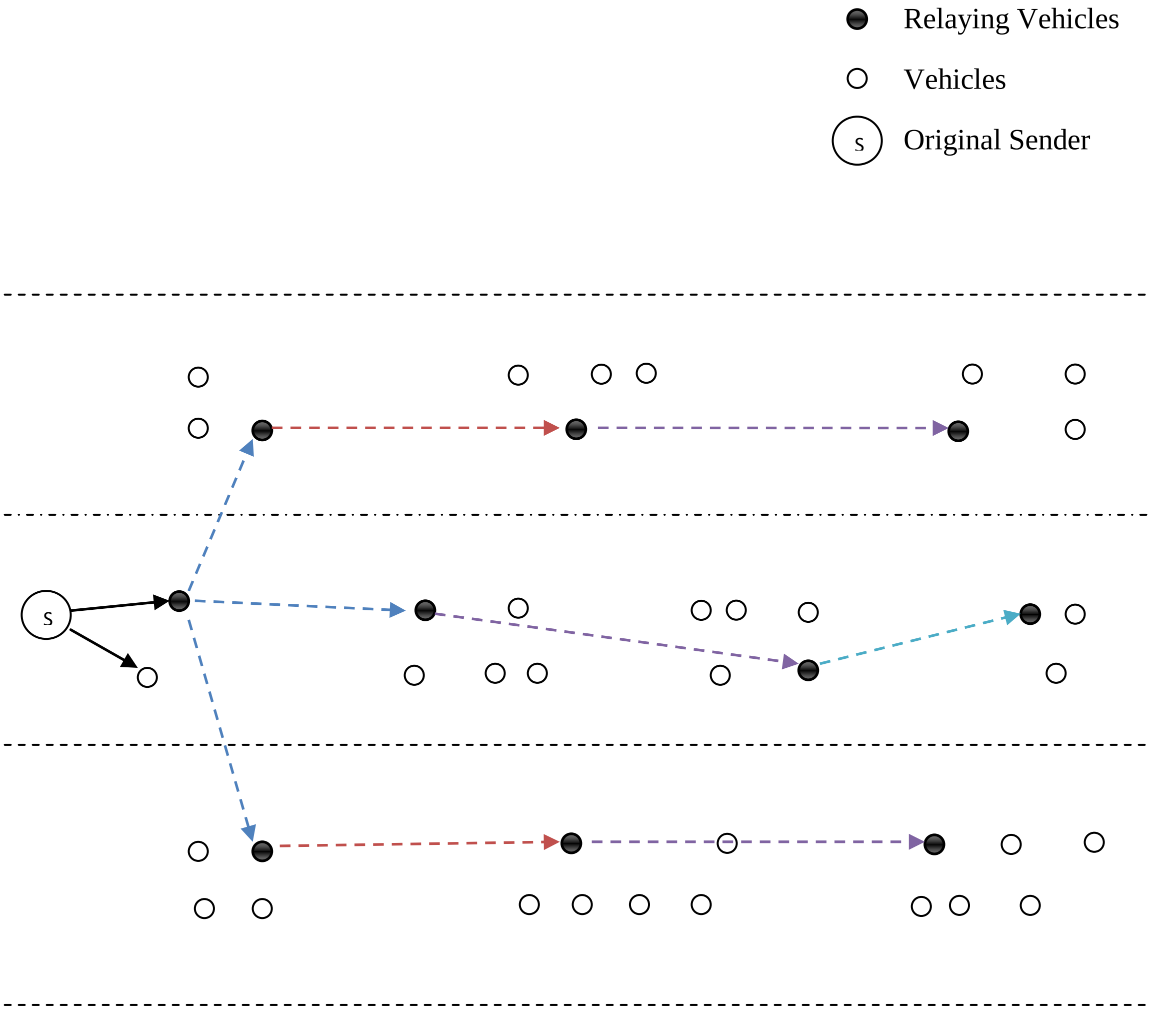

Figure 2.1: Message Propagation in Vehicular Networks

### 2.1.1 Data Design

In our system, there are three types of data: sender message, trust opinion, and aggregate message. We show the purpose and design of each in the following:

#### Sender Message

A sender $S$ prepares the message

$$M = [event, confidence, time, location] \tag{2.1}$$

where $confidence \in [0, 1]$, $time \in N$, and $location \in N \times N$. Sender confidence provides flexibility in reporting an event – higher confidence indicates the sender itself is more confident in the reported event. Time is a positive integer and location is a geographical coordinate, both being available from an equipped GPS device.

Please note that sender ID is not included in $M$; instead, it is included in the signed message $M' = [M, \mathrm{ID}, S]$, where $S$ is the signature. To ensure a secure data dissemination environment, we require sender messages be signed. The requirement of signatures applies to trust opinions as well.

#### Trust Opinion

Trust opinion is a message that serves as an evaluation of the sender message. Evaluation is conducted by comparing the reported event with the peer's current knowledge, which may come from a number of equipped car sensors, the local database, or even human interactions. We assume there is an *analysis module* that provides such an opinion. More specifically, the analysis module provides a mapping function $f : M \times K \rightarrow R \times C$ that maps the sender message $M$ and local knowledge $K$ to the product of reaction $R$ and confidence $C$, where $R = \{trust, \neg trust\}$ denotes whether the evaluator trusts the message, and $C = [0, 1]$ denotes how much confidence is placed.

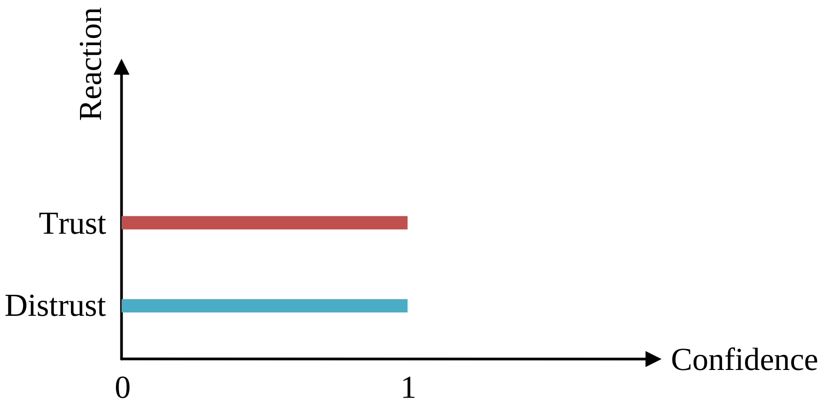

Figure 2.2: Trust Opinion

Trust opinion is a combination of reaction and confidence. Valid, reliable, correct messages should be trusted but cheating, unreliable, or jamming messages should be distrusted. The derivation of reaction is dependent on multiple factors, and each type of message may have its unique metric. For example, given that the event of sender message is observational, such as weather conditions, the neighboring vehicles are capable to give correct reactions based on their own observations.

Higher confidence values indicate that the evaluator itself is more confident in its evaluation. The generation of confidence values should be based on multiple factors as well, such as past peer interaction history, user preference, time and location difference between the sender and evaluator. For observational events, it is commonly believed that confidence decreases as time elapses and distance grows. For example, a peer witnesses a car accident and broadcasts a message. High confidence values should be placed for those nearby vehicles, but for those further peers who are not sure of the car accident, they may give a low confidence value, or not even provide a trust opinion.

In practice, upon reception of sender message $M'$, an evaluator $V_i$ verifies $M'$, extracts

$M$, analyzes $M$, and generates the trust opinion

$$O_i = [reaction, confidence] \tag{2.2}$$

. $V_i$ signs $M$ and $O_i$ into

$$O'_i = [M, O_i, \mathrm{ID}_i, S_i] \tag{2.3}$$

with its identity $\mathrm{ID}_i$ and signature $S_i$.

### Aggregate Message

An aggregate message is a combination of the sender message and trust opinions.

For example,

$$A = [M, \mathrm{ID}_0, O_1, \ldots, O_n, \mathrm{ID}_1, \ldots, \mathrm{ID}_n, AGGR(S_0, S_1, \ldots, S_n)] \tag{2.4}$$

is an aggregate from the sender message $M' = [M, \mathrm{ID}_0, S_0]$ and trust opinions from $n$ direct evaluators $V_i : O'_i = [M, O_i, \mathrm{ID}_i, S_i]$ where $i \in [1, n]$.

In our system, the size of an aggregate message increases with the number of collected feedbacks. Considering that the size of an aggregate message cannot grow infinitely large due to the limited packet payload[2], we require that trust opinions should not be appended to the aggregate message once its maximum size is reached – by doing so, we may lose subsequent trust opinions but an adequate number of previous trust opinions are preserved, whose feedbacks are usually more confident and sufficient for message evaluation.

$AGGR$ is our signature aggregation scheme. Flexible aggregation is necessary to allow that any third party can perform an aggregation. Secure aggregation is needed to ensure no data repudiation. Meanwhile, efficient aggregation is appreciated to minimize the extra information for signature verification. Such an aggregation scheme is illustrated in Chapter

[2] In computer networks, a network packet contains two parts: the header that indicates the source and destination as well as other meta information, and the payload which is the actual data being transmitted.

3, where any third party can combine multiple signatures into one aggregate signature, achieving both flexibility and efficiency without compromising security.

**An Example**

A vehicle $V_0$ discovered a car accident and broadcast a sender message $M' = [M, \mathrm{ID}_0, S_0]$, where $M$ is the message containing the event description "car accident", sender confidence, time and location where $V_0$ spotted the accident, $\mathrm{ID}_0$ is $V_0$'s digital identity, and $S_0$ is the signature on $[M, \mathrm{ID}_0]$.

Not to lose generality, we assume there are another two vehicles near $V_0$, namely $V_1$ and $V_2$. $V_1$ receives the message $M'$ and provides a trust opinion with a trust reaction and 0.8 confidence, while $V_2$ distrusts the message $M'$ and provides a distrust reaction and 0.5 confidence. The trust opinion from $V_1$ is $O'_1 = [M, O_1, \mathrm{ID}_1, S_1]$, where $O_1 = [trust, 0.8]$, $\mathrm{ID}_1$ is the identity of $V_1$, and $S_1$ is the signature on $[M, O_1, \mathrm{ID}_1]$. Similarly, the trust opinion from $V_2$ is $O'_2 = [M, O_2, \mathrm{ID}_2, S_2]$, where $O_2 = [\neg trust, 0.5]$.

Aggregation on $M'$, $O_1$ and $O_2$ can be done by any third party and these messages are aggregated into the aggregate message $A = [M, \mathrm{ID}_0, O_1, O_2, \mathrm{ID}_1, \mathrm{ID}_2, AGGR(S_0, S_1, S_2)]$. Please note that the actual proposed aggregation data format is slightly different from the example here. We leave the details of message aggregation to Chapter 3.

### 2.1.2 System Components

We design our system based on several components, as shown in Figure 2.3.

Message evaluation contains two modules: analysis module and action module. The analysis module is where trust opinions are generated. The module analyzes the message validity, correctness and accuracy based on local knowledge, and attempts to provide a trust opinion of either "trust" or "distrust". If a trust opinion can be provided, it is broadcast

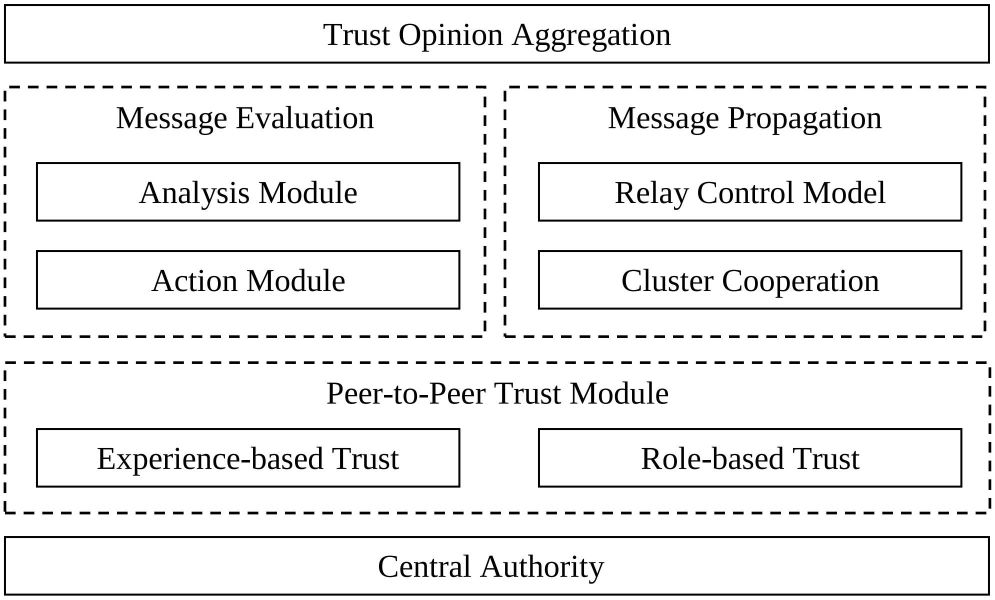

Figure 2.3: System Framework

and appended to the message. The action module is where a local decision (trust or distrust) is made. The peer derives a local action using a trust-based computation model in Section 2.2.2.

Message propagation consists of two components: cluster cooperation and the relay control model. Based on cluster-based routing mechanisms, the cluster cooperation serves as the foundation that makes the message propagation and trust aggregation possible. The relay control model works as a filter that controls the relay of messages based on a trust model in Section 2.4.2.

The trust opinion aggregation scheme ensures that message evaluation and propagation can be done with little interference on each other. It provides high flexibility that during message propagation, trust opinions can be aggregated in a secure and efficient fashion. We show the detail of trust opinion aggregation in Chapter 3.

Peer-to-peer trust module manages the trust of peers. Inspired by the work by Minhas et al. [20], we employ two types of trust: *role-based trust* and *experience-based trust*. A small minority of vehicles are assigned to a specific role and a role-based trust, such as police cars. For other vehicles, they are associated with an experience-based trust. Each peer maintains a list of experience-based trust for other peers.

The central authority is an offline entity which 1) assigns roles and updates role-based trust; 2) collects distributed experience trust from peers; 3) computes a global experience trust for all peers, and offers an up-to-date copy for download; 4) praises or punishes peers according to their behaviors in the network.

We show in the following the details of each component except for the central authority module. The work in [21] provides a sound approach to the computation of global trust from distributed local trust, which is an appendix to our work here.

## 2.2 Message Evaluation

### 2.2.1 Analysis Module

The analysis module works on the generation of trust opinion upon message reception. The purpose and data design of trust opinions is described in Section 2.1.1. One important hardware design principle is that the trust opinion should always be generated before any disclosure of the existing trust opinions in the message. In other words, the generation of the trust opinion is purely based on the peer's local knowledge, such as the direct observation. By doing so, we are capable of coping with gambling peers who give trust opinions by strategically guessing the message trustworthiness from others' trust opinions, so as to quickly and maliciously increase their trust.

For example, upon reception of an aggregate message, the gambling peer gives a "trust" reaction if there are more peers who trust the message in the list of trust opinions; otherwise, the gambling peers gives a "distrust" reaction. Because our system praises peers of honesty, if the majority of peers are honest, the gambling peer could get praised by simply following the opinion of the majority.

The design of such an analysis module would involve much concern from the perspective of hardware design, such as tamper-proof devices, car sensors and human-computer interactive interfaces. For now, we assume the existence of such an analysis module. The design of which is beyond our work scope and left for future.

### 2.2.2 Action Module

The action module works on the local actions taken towards the message. A local action can be derived from the trust opinions in the message. Specifically, an aggregate message trustworthiness is computed and mapped to a set of actions $\{trust, \neg trust\}$[3].

Let $A$ denote the aggregate message, $s$ denote the original sender, $P(trust) = \{i | \text{ ID}_i \in A \text{ and } O_i = [trust, c_i] \in A\}$ denote the peers who contribute trust opinions of "trust" reactions, and $P(\neg trust) = \{i | \text{ ID}_i \in A \text{ and } O_i = [\neg trust, c_i] \in A\}$ denote the peers with "distrust" reactions. Let $t_A$ denote the aggregate trustworthiness of aggregate message $A$. The action module of peer $p$ computes:

$$t_A = \frac{c_s + \sum\limits_{i \in P(trust)} c_i - \sum\limits_{i \in P(\neg trust)} c_i}{1 + |P(trust)| + |P(\neg trust)|} \tag{2.5}$$

where $c_s \in [0, 1]$ is the sender confidence in the sender message, and $c_i \in [0, 1]$ is the confidence in the trust opinion given by peer $i$. It is obvious to prove that $t_A \in (-1, 1]$. When the message $A$ is fully distrusted, we have $c_s = 0$, $P(trust) = \emptyset$, and $c_i = 1$ for $i \in P(\neg trust)$, and thus $t_A \to -1$. When $t_A = 1$, we have $c_s = c_i = 1$ for $i \in P(trust)$ and $P(\neg trust) = \emptyset$ which indicates that the message is fully trusted by peers.

[3] The distinction of trust and distrust has been introduced and modeled in previous research in artificial intelligence, such as the work by Tran [22].

Considering that the sender is a different role from those who provide trust opinions, we employ a sender weight factor $\gamma > 0$ that determines how much weight is placed on the sender. The computation of $t_A$ becomes:

$$t_A = \frac{\gamma c_s + \sum\limits_{i \in P(trust)} c_i - \sum\limits_{i \in P(\neg trust)} c_i}{\gamma + |P(trust)| + |P(\neg trust)|} \tag{2.6}$$

The value of $\gamma$ can be customized by each peer in the network. Setting $\gamma$ to a larger value indicates that the peer places more trust on the sender. The case $\gamma = 1$ amounts to Equation 2.5.

Next, we employ the peer-to-peer trust module here. Each peer $i$ is associated with a trust metric $T_i \in [0, 1]$, which is defined in Section 2.3. We combine the trust of each peer into the computation of aggregate trustworthiness as follows:

$$t_A = \frac{\gamma c_s T_s + \sum\limits_{i \in P(trust), T_i \geq \tau} c_i T_i - \sum\limits_{i \in P(\neg trust), T_i \geq \tau} c_i T_i}{\gamma T_s + \sum\limits_{i \in P(trust), T_i \geq \tau} T_i + \sum\limits_{i \in P(\neg trust), T_i \geq \tau} T_i} \tag{2.7}$$

where $\tau \in [0, 1]$ is the trust threshold customized by each peer $p$. The trust threshold helps filter trust opinions from those peers that are not highly trusted. $\tau$ can be set to a higher value close to 1 so that only trust opinions from highly trusted peers are trusted. In practice, the value of $\tau$ should be determined by the availability of trust opinions. For example, $\tau$ can be set higher when a larger number of trust opinions are available.

Last, the action module implements a mapping $f_{action} : T_A \rightarrow \{trust, \neg trust\}$ that maps the message trustworthiness to an action:

$$f_{action} = \begin{cases} trust & , \quad \text{if } t_A \geq \varphi \\ \neg trust & , \quad \text{if } t_A < \varphi \end{cases} \tag{2.8}$$

where $\varphi \in [-1, 1]$ is the action threshold. The value of $\varphi$ can be personalized by each peer: a higher action threshold indicates the peer is more "cautious" of trusting other peers and vice versa.

**Special case:** under the situation where the traffic is extremely sparse, both $P(trust)$ and $P(\neg trust)$ may be $\emptyset$ and the message only contains the sender's identity. If we simply compute the aggregate trustworthiness using the Equation 2.7, it becomes

$$t_A = \frac{\gamma c_s T_s}{\gamma T_s} = c_s \tag{2.9}$$

where the trust of the sender is eliminated and thus not considered. So in order for the action module to trust the message, along with the previous requirement in Equation 2.8 that $t_A \geq \varphi$, we further require that the action module can trust the message only if $T_s \geq \tau$.

As a summary, we show how our action module works in the following pseudocode.

**Algorithm 1** Action Module

```
$V_A \Leftarrow$ verify $A$.
if $V_A =$ false then
    return $\neg trust$.
else
    compute the value of $t_A$ using Equation 2.7.
    if $t_A < \phi$ then
        return $\neg trust$.
    else
        if $P(trust) = \emptyset$ and $P(\neg trust) = \emptyset$ then
            if $T_s < \gamma$ then
                return $\neg trust$.
            end if
        end if
        return $trust$.
    end if
end if
```

## 2.3 Peer-to-Peer Trust Module

In this section we present the peer-to-peer trust module proposed by Minhas et al. [20]. Each peer's trust is evaluated by a trust metric: *either* a role-based trust, *or* an experience-based trust. Let $T_i \in [0, 1]$ denote the peer-to-peer trust of peer $i$, we have

$$T_i = \begin{cases} T_{role}(i) & \text{if peer } i \text{ has a role,} \\ f(T_{exp}(i, p)) & \text{otherwise.} \end{cases} \tag{2.10}$$

where $T_{role}(i) \in [0, 1]$ is the role-based trust of peer $i$, and $T_{exp}(i, p) \in [-1, 1]$ is the experience-based trust of peer $i$ from peer $p$'s perspective. Since the values of $T_{role}$ and $T_{exp}$ have different ranges, we map the value of $T_{exp}$ to the same range of $T_{role}$ by employing the function $f(x) = (x + 1)/2$. Please note that we simply employ a mapping function here to map the values to the same range, although there could be other better mapping functions. We leave the work of developing these mapping functions to the future.

### 2.3.1 Role-based Trust

It is known that although most vehicles are for personal purposes, a small number of entities have their specific responsibilities in the traffic system, e.g, police cars. Roles are assigned to them and it is reasonable to assign multiple levels of trust to different roles. The underlying assumption is that vehicles of the same role would behave in a similar way so that any third party can estimate their trust level before any interaction happens. The roles and role-based trust values in our system are fixed by the offline central authority. To demonstrate the utilization of role-based peer trust, we define three different roles, from the highest to the lowest trust:

1. Authority, such as police cars, traffic controllers, and road-side units which serve as part of road infrastructure, and so on.

2. Public Services, which could be ambulance, fire truck, school bus, public transits, road maintenance cars, etc.

3. Professional Cars, e.g. driver training vehicles, cars whose drivers have ten years of safe driving experience, etc.

We denote the role-based trust of peer $i$ as $T_{role}(i)$, where $T_{role} : \text{ID} \rightarrow [0, 1]$; 1 means absolute trust and 0 represents absolute distrust. The vehicle identity can be mapped to its role and then the role-based trust. In practice, vehicles periodically download from central authority an up-to-date list of roles, each with a list of vehicle identities.

### 2.3.2 Experience-based Trust

For most of the vehicles who do not have a role, we use the experience-based peer trust to dynamically reflect a peer's trustworthiness in the system. The behavior of a peer is evaluated by other peers, each of whom maintains the trust for a list of peers in the system. The list of trust is preserved in peer's local repository, and loosely synchronized with the central authority who collects, analyzes these lists, and reaches a decision on praising or punishing certain peers.

The peer updates the trust for others by comparing their messages to its own experience. More specifically, upon reception of an aggregate message, the peer may not have the direct experience to judge the trustworthiness of message. However, after some time when direct experience becomes available, the peer can updates the trust for those who either reported or evaluated the event.

For instance, peer $X$ is the original sender of a fake message $M$. Peer $Y$ receives the message and immediately finds out it is fake. It appends its own trust opinion to $M$ and updates the trust of $X$. The trust of $X$ with respect to $Y$ is now reduced. And then the message $M$ arrives at $Z$, whose analysis module cannot judge the trustworthiness due to lack of direct observations. The action module of $Z$ derives a local action for $Z$ but this does not account for any experience here. And soon, $Z$ obtains its direct observation and realizes that $X$'s message is not true and $Y$'s opinion is valid. It will update both $X$ and $Y$'s trust values. The trust value of $X$ with respect to $Z$ is decreased, and the trust of $Y$ with respect to $Z$ is increased.

Motivated by the model of Minhas et al. [20], we denote the peer $p$'s trust from $k$'s perspective as $T_{exp}(p,k)$, whose value is scaled to $[-1,1]$ :$-1$ means absolute distrust and 1 means absolute trust. We simplify the notation of $T_{exp}(p,k)$ as $T$ in the following. Peer $k$ increases the trust of $p$ by

$$T \leftarrow \begin{cases} \lambda^{t}(1-c\alpha)T + c\alpha & \text{if } T \geq 0 \\ \lambda^{-t}(1+c\alpha)T + c\alpha & \text{if } T < 0 \end{cases} \tag{2.11}$$

if $p$ acts honestly, otherwise decreases $T$ by

$$T \leftarrow \begin{cases} \lambda^{t}(1+c\beta)T - c\beta & \text{if } T \geq 0 \\ \lambda^{-t}(1-c\beta)T - c\beta & \text{if } T < 0 \end{cases} \tag{2.12}$$

where $\alpha, \beta \in (0,1)$ are increment and decrement factors, $c \in [0,1]$ is the confidence value placed by $p$ in the message, $\lambda \in (0,1)$ is a forgetting factor, and $t \in [0,1]$ is the time closeness between the current interaction and the previous one.

As explained by Minhas et al. [20], the values of $\alpha$ and $\beta$ should be subjective to the road situations and message types. For example, when traffic is sparse, these values should be set larger, considering the number of messages is low. For emergency related events, the values should be larger so as to increase or decrease peer trust more rapidly. Besides, as proposed by Minhas et al. [20] and introduced by Tran [22], it is appreciated that $\beta > \alpha$ based on the common assumption that peer trust is difficult to build up but easy to tear down.

The Equation 2.11 and 2.12 are extended from the work of Tran [22]. Compare to Tran's work, we add the confidence $c$ as an factor because peers, including the sender, play different roles in the message's trustworthiness by placing different confidence values. This can be explained by the design of Equation 2.7, which computes the message's aggregate trustworthiness from a peer's trust and confidence. For example, between two peers with the same peer-to-peer trust, the one who has placed a confidence $c = 1$ is making greater impact than the other with a confidence $c = 0.1$. Consequently, those with higher confidence would increase or decrease their trust faster than those with lower confidence. In other words, if a peer provides a correct trust opinion, it should be praised by how much confidence it has placed in the message – the higher confidence value it gives, the more it should be praised. This also applies to the other direction, i.e. the punishment when a peer gives a wrong trust opinion.

Following the proposal of Minhas et al. [20], we also model the time closeness $t$ as

$$t = \begin{cases} (t_c - t_e)/t_{max} & \text{if } t_c - t_e < t_{max} \\ 1 & \text{otherwise} \end{cases} \tag{2.13}$$

where $t_c$ is the current time and $t_e$ is the event time in the message; $t_{max}$ is the maximum time for a peer to totally forget the experience that happened before time $t_c - t_{max}$. The value of $t_{max}$ is dependent on the frequency of the interactions between two peers in the network, and thus it should be set large under sparse traffic scenarios or small under dense traffic situations.

## 2.4 Message Propagation

### 2.4.1 Cluster Cooperation

In a decentralized network where peers are independent from each other, it is hard to define a strategy to effectively and efficiently collect trust opinions and propagate aggregate messages. Specifically, it is difficult to provide a strategy that defines to which (geographical) scale and in which fashion that trust opinions can be collected, packed and relayed in one aggregate message.

To achieve a scalable message and trust opinion aggregation, we rely on the cluster-based data routing model, whose example is shown in Figure 2.4. A number of cluster-based routing protocols have been proposed and studied in mobile ad-hoc networks [23, 24, 25] and vehicular networks [26, 27, 28]. One common goal of the protocols in vehicular networks is to achieve scalability for the vehicle-to-vehicle messaging. By grouping peers into multiple clusters, the system becomes scalable by having message relay done between cluster leaders instead of between two neighboring peers. The latter relay method is not scalable because the messaging between all pairs of neighboring peers will result in channel congestion and thus reversely threshold the scalability.

Our proposed cluster cooperation scheme relies on existing cluster-based routing protocols in vehicular networks [26, 27, 28], and extends them in two aspects. First, clusters are used to aggregate trust opinions. Trust opinions from peers in the cluster are aggregated and propagated along with the message itself. We will explain this with our example in the following context. Second, we employ the majority opinion computed from trust opinions as the decision of relay control model, which is introduced in Section 2.4.2.

In our example shown in Figure 2.4, vehicles are geographically grouped into 10 clusters, from $C_1$ to $C_{10}$ [4]. For each cluster $C_i$, a vehicle is randomly chosen as cluster leader $L_i$ (the black nodes). Our scheme requires that the cooperation among neighboring cluster leaders is pre-established to help build an intra-cluster link topology (the graph with dashed arrows connecting neighboring black peers) so that message can be relayed from one cluster to another. Sender $S$ in cluster $C_1$ broadcasts a message $M$ to its members who give their feedbacks $O_i$ immediately afterwards. And then, the cluster leader $L_1$ collects $O_i$ and aggregates them into the aggregate message $A$. $L_1$ sends $A$ to the next hop clusters $C_2$, $C_3$

[4] The geography-based clustering technique is proposed in the work of Raya et al. [29]

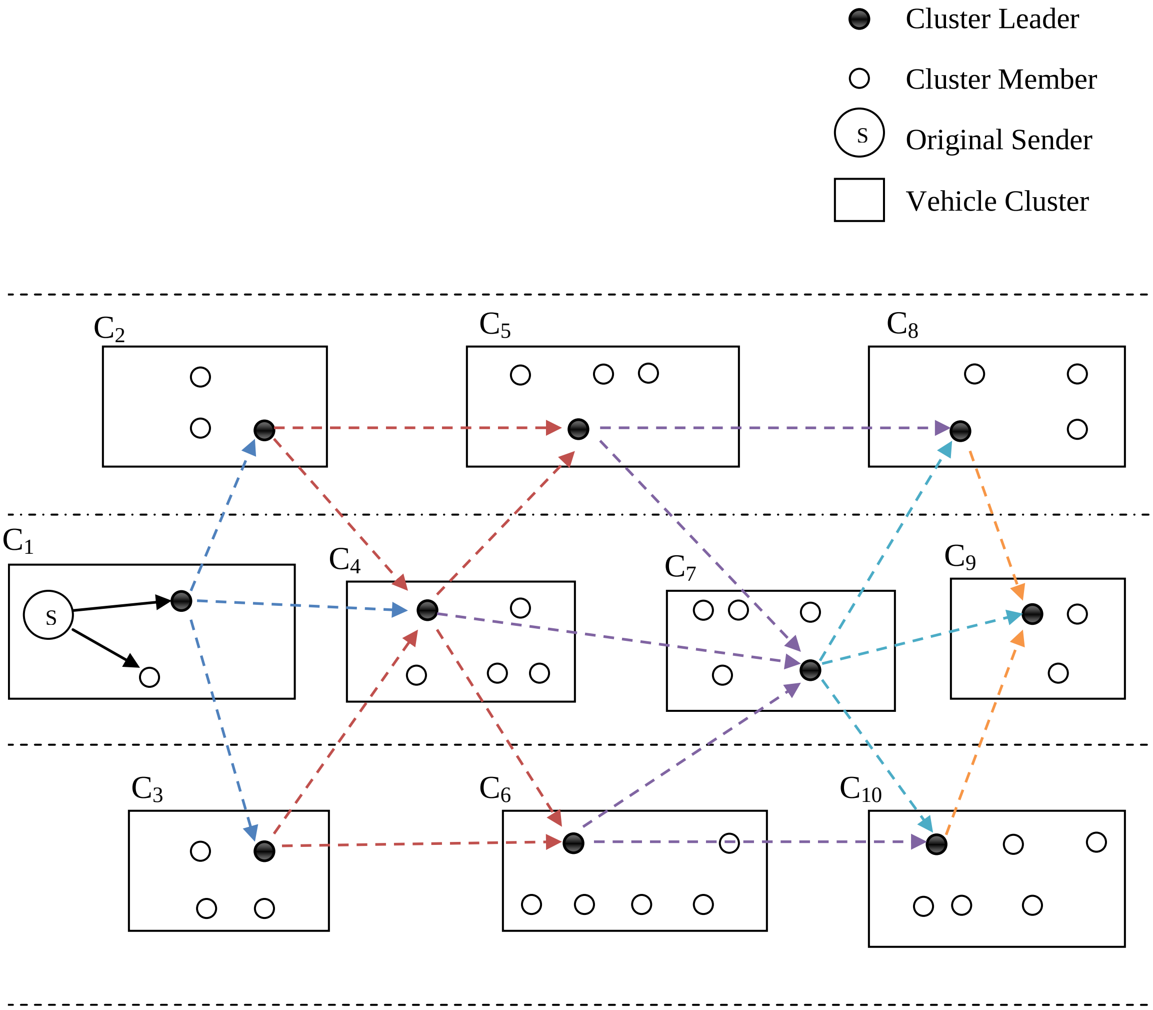

Figure 2.4: Cluster-based Message Propagation

and $C_4$. Upon reception of $A$, the cluster leader (e.g. $L_4$ here) broadcasts $A$ to its cluster members, collects their trust opinions (if any), aggregates them together with existing $A$ into the new aggregate $A'$ , computes a relaying decision, and decides whether to relay $A'$ to the next hop cluster $C_5$, $C_6$ and $C_7$.

### 2.4.2 Relay Control Model

While traditional routing algorithms [15] of vehicular networks use "time-to-live" or "hop-to-live" as a relay decision, our decision is built on the majority opinion – a message trusted by the majority should be relayed; otherwise it is to be dropped. This idea is motivated by the modeling of trust proposed by Minhas et al. [20] where majority opinion is used to judge the overall decision. Compared to their work where the time closeness and location closeness are considered as part of a peer's opinion, we use the confidence in the trust opinion and the peer-to-peer trust as the peer's opinion, as is shown in Equation 2.14 and 2.15.

Besides, considering that trustworthiness of messages ages with the time and distance – the longer time elapses, the further away the event incurs, the less accurate and reliable the data becomes – we further use a mapping function $f_{max} : \Lambda \times \Theta \to M_t \times M_d$ which maps the sender role $\Lambda$ and the event $\Theta$ to the maximum time-to-live $M_t$ and longest propagation distance $M_d$. We define such a mapping function because it is reasonable to set up different thresholds to multiple types of messages and types of senders. Take the distance $M_d$ for an example: a piece of weather information (infotainment event) can have a propagation area of 10 square miles while a life-critical message (safety-related event), e.g. "sudden brake" may only be useful within a distance of 200 meters. Similarly, the message from an authority role should propagate as far as possible. However, those not highly trusted senders should have their messages propagate less further than the messages from highly trusted peers.

In practice, when the cluster leader receives an aggregate message $A$ from previous cluster, it broadcasts $A$ to cluster members, collects trust opinions (if any) and generates new aggregate $A'$ ($A' \leftarrow A + O_1 + \ldots + O_k + \ldots$). The leader computes the majority opinion from $A'$ and then makes a relay decision.

### Computing the Majority Opinion

The majority opinion is based on the weight of majority trust and majority distrust. Let $P(trust) = \{i | \mathrm{ID}_i \in A' \text{ and } O_i = [trust, c_i] \in A'\}$, and $P(\neg trust) = \{i | \mathrm{ID}_i \in A' \text{ and } O_i = [\neg trust, c_i] \in A'\}$.

The relayer $p$ computes the weight of majority trust and distrust as

$$W_{trust} = \sum_{i \in P(trust), T_i \geq \tau} c_i T_i \tag{2.14}$$

and

$$W_{\neg trust} = \sum_{i \in P(\neg trust), T_i \geq \tau} c_i T_i \tag{2.15}$$

where $\tau$ is the trust threshold set by $p$, $c_i \in [0, 1]$ is the confidence given by peer $i$, and $T_i$ is the peer-to-peer trust. Messages can be relayed only if

$$\frac{W_{trust}}{W_{trust} + W_{\neg trust}} > 1 - \varepsilon \tag{2.16}$$

where $\varepsilon \in [0, 1]$ is a threshold set by the system to denote the maximum error rate allowed. $\varepsilon$ is embedded in the protocol and can be adaptive to the current environments, situations and data types. For example, for more critical messages such as car accidents, a lower error rate is appreciated; for weather information, a higher error rate can be allowed.

### Making a Relay Decision

Apart from the majority opinion, a relay decision is also based on the following parameters:

$m_d$ – the maximum propagation distance;

$m_t$ – the longest time to live;

$\Delta d$ – the distance between current location and event location;

$\Delta t$ – the time that has elapsed since the event;

A relayer does the following:

1. verify message $A$; in case verification fails, drop $A$;

2. compute the majority opinion, drop $A$ if Equation 2.16 does not hold;

3. compute $\Delta d, m_d, \Delta t, m_t$, if $\Delta d > m_d$ or $\Delta t > m_t$ , drop $A$;

4. generate $A'$ and relay $A'$ to the next hop clusters.

As a summary, we show how our relay control module works in the following pseudocode in Algorithm 2.

## 2.5 Presence of Authority

In the presence of authority, the system adapts itself in the following aspects:

1. Messages from authority roles are trusted and propagated to the maximum distance. All trust opinions will be ignored in message evaluation and propagation.

2. Trust opinions from authority are followed as the guidance to action module. In other words, the action module simply follows the trust opinion from authority without computing the aggregated trustworthiness of the message.

3. Authority's trust opinion serves as an guidance to the relay control model. If the authority roles provide a trust opinion for the message, the relay model simply follows the decision of authority.

**Algorithm 2** Relay Control Module

```
$V_A \Leftarrow$ verify $A$ upon reception of message $A$.
if $V_A =$ false then
    return drop.
else
    broadcast $A$ to cluster members.
    collect trust opinions $O_i$ from cluster members.
    generate $A' \Leftarrow A + O_i + \ldots$.
    compute routing decision $r$ using Equation 2.16.
    if $r =$ relay then
        if $\Delta d > m_d$ or $\Delta t > m_t$ then
            return drop.
        else
            return relay.
        end if
    else
        return drop.
    end if
end if
```

4. The role-based trust for authority is the highest and fixed as 1.

## 2.6 Summary

In this chapter, we present a trust-based message evaluation and propagation model for vehicular networks. Message evaluation is conducted during its propagation, and reversely affects message propagation through the relay control model, whose relay decisions are determined by majority opinions, using a trust-based computational approach. The peer-to-peer trust is modeled to reflect the honesty of entities in the network. Based on that, our evaluation metric computes the message trustworthiness from a set of trust opinions. In the next chapter, we are going to present an aggregation scheme, which resolves the issues on how trust opinions are generated and aggregated in a secure, efficient, and scalable fashion.

# Chapter 3

# Secure and Efficient Trust Opinion Aggregation

In this chapter, we introduce a secure and efficient trust opinion aggregation scheme in mobile ad-hoc networks, typically in vehicular networks here. The outline of this chapter is organized as follows. Key issues and challenges in trust opinion aggregation are identified in Section 3.1. After that, we introduce our extended identity-based aggregation scheme in Section 3.2, and summarize our scheme in Section 3.3.

## 3.1 Key Issues in Trust Opinion Aggregation

Figure 3.1 illustrates an example of trust opinion aggregation. The topology of aggregation is based on cluster-based message routing mechanism. Trust opinions are aggregated when message $M$ from peer $S$ propagates from cluster $C_1$ to $C_7$. Let $S_i$ denote the set of opinions from peers in cluster $C_i$, i.e. $S_i = \{O_k |$ peer $p_k$ generates trust opinion $O_k$, and $p_k \in C_i\}$.

The cluster leader of $C_1$ receives the message $M$ and a set of trust opinions $S_1$, combines them into an aggregate message $A = [M, S_1]$, and then sends $A$ to the next hop clusters. For the cluster leader of $C_i$, $i \in [2, 7]$, it receives several aggregate messages from previous hops, combines them into a new aggregate, broadcasts it to cluster members, collects trust opinions if there is any, combines these trust opinions into a newer aggregate, and relays it to next hop clusters.

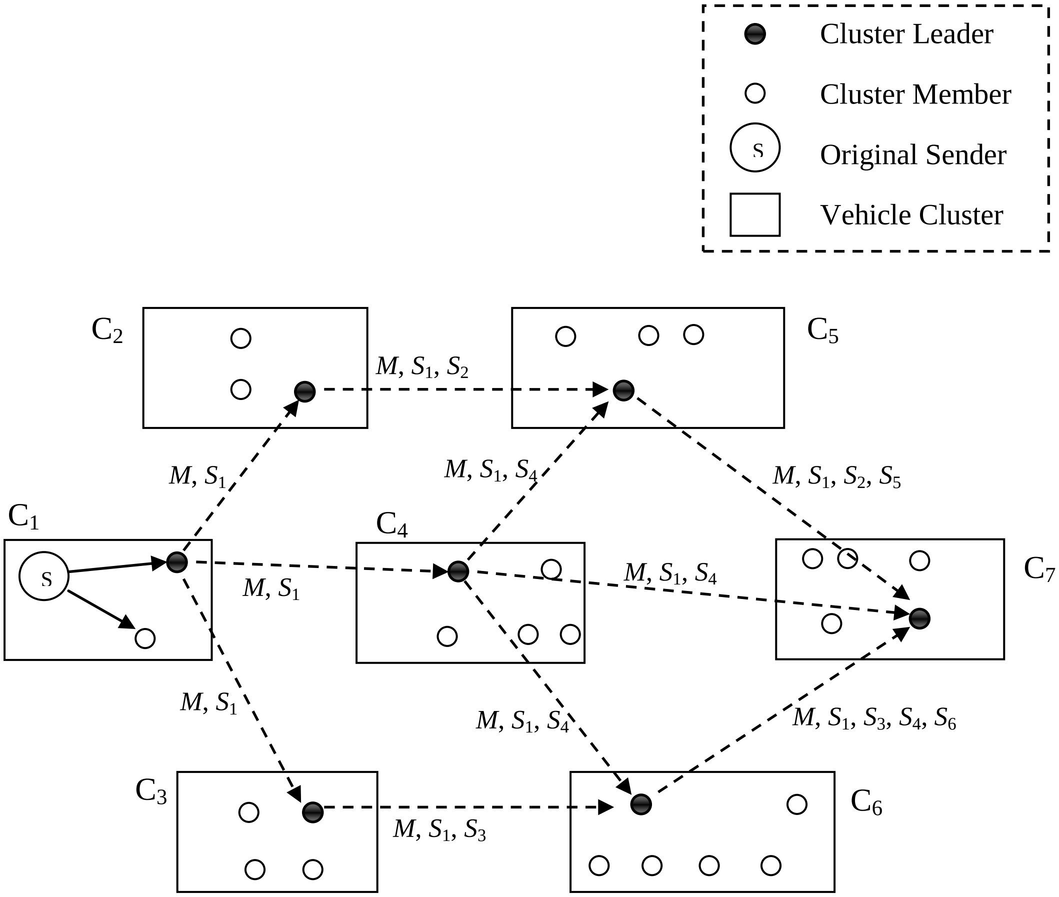

Figure 3.1: Trust Opinion Aggregation

We realize that the system must be able to resolve several important issues in security and efficiency before a workable trust opinion aggregation is available, detailed as below.

### Secure Aggregation

Security is an important factor in the design of some network systems. All data in the vehicular network need to be protected from attackers, because attacks are possible during the aggregation of trust opinions, such as data repudiation where trust opinions are modified by a malicious relayer, and sybil attack where messages are sent under arbitrarily forged identities or under the identity of another innocent peer [30]. Due to the above concerns, it is required that all data are signed before sent so that 1) data repudiation can be easily detected by verifying the data against with sender's signature; 2) message sender, or trust opinion provider cannot deny its message because of the existence of its signature; 3) identities cannot be forged or abused because each message is mapped to a valid and unique identity – the only peer who is able to sign the message.

### Efficient Aggregation

Given the fact that the necessity of signature schemes in vehicular networks is commonly agreed on by both academia and industry, we realize that security and efficiency are two contradictory aspects in vehicular networks, as it is difficult to achieve one aspect without degrading the other. In our system, it is important for us to preserve the necessary security, but at the same time, to maximize efficiency to render the aggregation scalable.

Here, the efficiency in trust opinion aggregation consists of two aspects: 1) time efficiency, the time needed to perform an aggregation; 2) space efficiency, the size needed to aggregate all trust opinions and signatures. Our concern is that previous work on data aggregation may become inefficient in either space or time cost when it comes to our trust opinion aggregation scenario. As a result, we appreciate an efficient aggregation scheme that achieves both time efficiency and space efficiency. We will present the related work in data aggregation in Chapter 5.2 (typically, the secure data aggregation methods in Chapter 5.2.1), and compare secure data aggregation methods to our proposed aggregation scheme in Chapter 4.4.

### Identity-based Aggregate Signature Algorithm

In this chapter, we propose an aggregation scheme that is based on the existing identity-based aggregate signature algorithm [31]. Specifically, the identity-based aggregate signature algorithm works as follows. Given a message $M$ and $n$ peers, each peer $i$ signs the message $M$ into $M_i = [M, \mathrm{ID}_i, G_i]$ for $i \in [1, n]$, where $\mathrm{ID}_i$ is the identity of peer $i$ and $G_i$ is the signature by peer $i$. An aggregator computes $G' = \sum_{i=1}^{n} G_i$ and generates the aggregated message $A = [M, \mathrm{ID}_1, \ldots, \mathrm{ID}_n, G']$. The computation and summation of $G_i$ is implemented over bilinear groups constructed from the modified Weil pairing over elliptic curves [32].

Our aggregation scheme featuring the employment of the identity-based aggregate signature algorithm improves both space efficiency and time efficiency. On one hand, space efficiency is achieved because all signatures are aggregated into one signature that remains in constant size. On the other hand, the inherent nature of identity-based signature does not rely on an aggregation chain as previous approaches do. Messages can be efficiently aggregated in the fixed time and thus time efficiency is achieved.

However, the original identity-based aggregate signature algorithm may not be directly ported to here because of the specific aggregation issues in vehicular networks. One issue with the aggregate signature algorithm is that it can only aggregate signatures $G_i$ on one single message $M$ as shown by the above example. Specifically, each peer $i$ signs the same message $M$ into its signature $G_i$, and the aggregate signature algorithms works correctly in computing the aggregate signature $G' = \sum G_i$. However, the algorithm does not provide a method on how to aggregate signatures when it comes to our aggregation scenario where the signatures are signed by peers on their own messages $M_i$, which are different because they are either the sender message $M$ or trust opinions $O_i$. Consequently, we need an aggregate signature algorithm that is capable of correctly combining the signatures for multiple messages into one aggregate signature.

Another issue that impedes the original identity-based aggregate signature algorithm from working correctly is the signature redundancy. We explain the concept of signature redundancy using the example in Figure 3.1 where the relayer of $C_6$ receives two aggregate messages from $C_3$ and $C_4$. The two aggregates share a duplicate set of trust opinions $S_1$ and each signature on the trust opinion in $S_1$ has a redundant copy. Elimination of duplicate trust opinions is necessary to improve space efficiency, but the original aggregate signature algorithm fails to verify the aggregate signature because it requires that each signer is a unique identity. In other words, if we have a pair of duplicate signatures, say $G_i$ and $G_j$ by $\mathrm{ID}_i$, the aggregate signature $G'$ is not verifiable. We need an aggregate signature algorithm that keeps $G'$ verifiable under the case where $\mathrm{ID}_i = \mathrm{ID}_j$ for any $i, j \in [1, n]$ and $i \neq j$.

In short, our scheme combines multiple messages and their signatures securely and improves the aggregation efficiency. Besides, redundant trust opinions can be eliminated with redundant signatures appropriately merged into an existing signature, which remains valid and verifiable.

## 3.2 Our Extended Identity-based Aggregation Scheme

In this section, we illustrate an identity-based aggregation scheme that extends the identity-based aggregate signature algorithm [31]. To start with, we explain the basic concept of identity-based signature and aggregate signature, and introduce the "Bilinear Maps" which serves as the mathematical foundation of the aggregate signature scheme. After that, we show our trust opinion aggregation scheme in details: system setup, message and trust opinion signing, trust opinion aggregation, signature verification, and proof of correctness.

### 3.2.1 Preliminaries

An identity-based signature is a signature scheme where the user's identity is used to generate its public key. The identity is a short binary string constructed from what uniquely identifies a user – could be the email address, social insurance number, etc. In a vehicular network setting, we can assume each car has a unique ID issued by the central authority. Given an ID of a vehicle, the central authority computes the unique private key and securely distributes it to the corresponding vehicle.

An aggregate signature is a single short binary string which convinces the verifier that for $1 \leq i \leq n$, signer $S_i$ signed the message $M_i$ where the $n$ signers and $n$ messages can be distinct and independent from each other. Instead of keeping $n$ distinct signatures as traditional signature schemes do, the aggregate signature is one single signature that compacts $n$ signatures.

#### Bilinear Maps

Let $G$ be a cyclic additive group generated by $P$, and $G_T$ be a cyclic multiplicative group. $G$ and $G_T$ have the same prime order $q$, i.e., $|G| = |G_T| = q$. Let $\hat{e} : G \times G \rightarrow G_T$ be a bilinear map, which satisfies the following properties:

- Bilinear: for all $P, Q, R \in G$, and $a, b \in Z$, $\hat{e}(Q, P + R) = \hat{e}(P + R, Q) = \hat{e}(P, Q) \cdot \hat{e}(R, Q)$. In particular, $\hat{e}(aP, bP) = \hat{e}(P, bP)^a = \hat{e}(aP, P)^b = \hat{e}(P, P)^{ab}$.
- Non-degenerate: there exists $P, Q \in G$ such that $\hat{e}(P, Q) \neq 1_{G_T}$.
- Computable: there is an efficient algorithm to compute $\hat{e}(P, Q)$ for any $P, Q \in G$.

Such a bilinear map $\hat{e}$, which is often called an *admissible pairing*, can be constructed by the modified Weil pairings on elliptic curves [32]. The groups ($G$ and $G_T$) on such a map are called bilinear groups, where Decisional Diffie-Hellman problem is easy to resolve but Computational Diffie-Hellman problem is believed to be hard [32]. E.g., given any $a, b, c \in Z_q$ and $P, aP, bP, cP \in G$, there exists an efficient algorithm to determine whether $ab = c$ by checking if $\hat{e}(aP, bP) = \hat{e}(P, cP)$ holds. However, it is theoretically hard to compute $abP \in G$.

### 3.2.2 System Setup

The central authority (CA) generates the system parameters as shown below:

1. generates groups $G$ and $G_T$ of order $q$, an admissible pairing $\hat{e} : G \times G \to G_T$;
2. chooses an arbitrary generator $P \in G$;
3. chooses a random $s \in Z/qZ$, and computes $Q = sP$;
4. chooses three hash functions $H_1, H_2 : \{0,1\}^* \to G$ and $H_3 : \{0,1\}^* \to Z/qZ$.

The system parameters $\{G, G_T, \hat{e}, P, Q, H_1, H_2, H_3\}$ are made public. The secret kept by CA is $s \in Z/qZ$.

Next, CA generates the public key $K_i$ and private key $k_i$ for each vehicle $V_i$. Given a vehicle $V_i$ with $\mathrm{ID}_i$, the CA generates two pairs of keys $(K_{i,0}, k_{i,0})$ and $(K_{i,1}, k_{i,1})$, where $K_{i,j} = H_1(\mathrm{ID}_i, j), k_{i,j} = sK_{i,j}, j \in \{0, 1\}$.

Let $K_i = K_{i,0}||K_{i,1}$ and $k_i = k_{i,0}||k_{i,1}$, where $||$ denotes concatenation. Please note that both $K_i$ and $k_i$ are unique for each vehicle $v_i$. Given an $\mathrm{ID}_i$ any third party can compute $K_i$ but $k_i$ can only be generated by the CA.

### 3.2.3 Message and Trust Opinion Signing

There are two types of identities who sign messages: 1) the sender, who is the message originator; and 2) the evaluator, who gives a trust opinion. We explain how they sign their messages in the following two cases.

**Case 1**: a vehicle $V_0$ is the message originator, who signs its message $M$ as follow:

1. computes $P_M = H_2(M) \in G$;
2. computes $h_0 = H_3(M, \mathrm{ID}_0) \in Z/qZ$;
3. generates a random $r_0 \in Z/qZ$ and initiates $c_0 = 1 \in Z$;
4. computes $S_0 = r_0 P_M + c_0 k_{0,0} + c_0 h_0 k_{0,1}$, $T_0 = r_0 P$;
5. generates the signed message $M_0 = [M, c_0, \mathrm{ID}_0, S_0, T_0]$.

**Case 2**: a vehicle $V_i$ evaluates the message $M$, generates its trust opinion $O_i$, and signs $M$ and $O_i$ as follow:

1. computes $P_M = H_2(M) \in G$;
2. computes $h_i = H_3(M, O_i, \mathrm{ID}_i) \in Z/qZ$;
3. generates a random $r_i \in Z/qZ$ and initiates $c_i = 1 \in Z$;
4. computes $S_i = r_i P_M + c_i k_{i,0} + c_i h_i k_{i,1}$, $T_i = r_i P$;
5. generates the signed message: $M_i = [M, O_i, c_i, \mathrm{ID}_i, S_i, T_i]$.

The computation of $S_i$ and $T_i$ is implemented over the group $G$. The combination of both $S_i$ and $T_i$ serves as the signature for the message $M$ (case 1) or $M$ and $O_i$ (case 2).

### 3.2.4 Trust Opinion Aggregation

We explain how our scheme aggregates trust opinions under two cases. Case 1 applies to the situation when there are no duplicate trust opinions or signatures while case 2 handles the signature redundancy.

**Case 1**: a third party $V'$ combines the original signed message with $n$ signed trust opinions from $n$ distinct nodes. Specifically, we have:

$V_0 : M_0 = [M, c_0, \mathrm{ID}_0, S_0, T_0]$ and $V_i : M_i = [M, O_i, c_i, \mathrm{ID}_i, S_i, T_i]$ for $i \in [1, n]$.

$V'$ computes

$$S' = \sum_{i=0}^{n} S_i \text{ and } T' = \sum_{i=0}^{n} T_i \tag{3.1}$$

and generates the aggregate

$$A' = [M, O_1, \ldots, O_n, c_0, c_1, \ldots, c_n, \mathrm{ID}_0, \mathrm{ID}_1, \ldots, \mathrm{ID}_n, S', T'] \tag{3.2}$$

The summation of $S_i$ and $T_i$ is implemented over the group $G$.

**Case 2:** a third party $V'$ wants to combine two aggregates on the same message $M$ into a larger aggregate, and it is possible that there exist duplicate trust opinions. Please note that instead of merging aggregates on different messages, our system only combines multiple aggregates for the same message $M$. Not to lose generality, we assume that $V'$ receives the following aggregates:

$$A_1 = \begin{bmatrix} M, O_1, \ldots, O_p, O_{p+1}, \ldots, O_{p+k}, \\ c_{0,1}, c_{1,1}, \ldots, c_{p,1}, c_{p+1,1}, \ldots, c_{p+k,1}, \\ \mathrm{ID}_0, \mathrm{ID}_1, \ldots, \mathrm{ID}_p, \mathrm{ID}_{p+1}, \ldots, \mathrm{ID}_{p+k}, \\ S_1, T_1 \end{bmatrix} \tag{3.3}$$

and

$$A_2 = \begin{bmatrix} M, O_{p+1}, \ldots, O_{p+k}, O_{p+k+1}, \ldots, O_{p+k+q}, \\ c_{0,2}, c_{p+1,2}, \ldots, c_{p+k,2}, c_{p+k+1,2}, \ldots, c_{p+k+q,2}, \\ \mathrm{ID}_0, \mathrm{ID}_{p+1}, \ldots, \mathrm{ID}_{p+k}, \mathrm{ID}_{p+k+1}, \ldots, \mathrm{ID}_{p+k+q}, \\ S_2, T_2 \end{bmatrix} \tag{3.4}$$

where $A_1$ and $A_2$ share $k$ duplicate trust opinions, i.e. $O_{p+i}$ for $i \in [1, k]$. Please note that for $i \in \{0\} \cup [p+1, p+k]$, $\mathrm{ID}_i$ may have different $c_i$ values in $A_1$ and $A_2$ due to various paths of aggregation, so we denote them as $c_{i,1}$ and $c_{i,2}$.

$V'$ computes $S' = S_1 + S_2$, $T' = T_1 + T_2$,

$$c'_i = \begin{cases} c_{i,1} & \text{, for } i \in [1, p] \\ c_{i,1} + c_{i,2} & \text{, for } i \in \{0\} \cup [p+1, p+k] \\ c_{i,2} & \text{, for } i \in [p+k+1, p+k+q] \end{cases} \tag{3.5}$$

and generates the new aggregate

$$A' = \begin{bmatrix} M, O_1, \ldots, O_p, O_{p+1}, \ldots, O_{p+k}, O_{p+k+1}, \ldots, O_{p+k+q}, \\ c'_0, c'_1, \ldots, c'_p, c'_{p+1}, \ldots, c'_{p+k}, c'_{p+k+1}, \ldots, c'_{p+k+q}, \\ ID_0, \mathrm{ID}_1, \ldots, \mathrm{ID}_p, \mathrm{ID}_{p+1}, \ldots, \mathrm{ID}_{p+k}, \mathrm{ID}_{p+k+1}, \ldots, \mathrm{ID}_{p+k+q}, \\ S', T' \end{bmatrix} \tag{3.6}$$

### An Example

We give an example showing how aggregation on two aggregates works. Suppose we have four vehicles $V_0, V_1, V_2, V_3$ where $V_0$ is the original message sender, and two existing aggregates $A_1 = [M, O_1, O_2, \mathrm{ID}_0, \mathrm{ID}_1, \mathrm{ID}_2, c_{0,1}, c_{1,1}, c_{2,1}, S_1, T_1]$ that combines messages from $V_0, V_1, V_2$, and $A_2 = [M, O_2, O_3, \mathrm{ID}_0, \mathrm{ID}_2, \mathrm{ID}_3, c_{0,2}, c_{2,2}, c_{3,2}, S_2, T_2]$ that combines messages from $V_0, V_2, V_3$. Both aggregates share a common trust opinion $O_2$ that is duplicate. An aggregator generates the new aggregates $A'$ as

$$A' = [M, O_1, O_2, O_3, \mathrm{ID}_0, \mathrm{ID}_1, \mathrm{ID}_2, \mathrm{ID}_3, c'_0, c'_1, c'_2, c'_3, S', T']$$

where $c'_0 = c_{0,1} + c_{0,2}$, $c'_1 = c_{1,1}$, $c'_2 = c_{2,1} + c_{2,2}$, $c'_3 = c_{3,2}$ and $S' = S_1 + S_2$, $T' = T_1 + T_2$.

### 3.2.5 Signature Verification

**Case 1**: verify a signed message with no trust opinions yet. Given a signed message $M_0 = [M, c_0, \mathrm{ID}_0, S_0, T_0]$, the verifier checks if

$$\hat{e}(S_0, P) = \hat{e}(T_0, P_M)\hat{e}(Q, c_0K_{0,0} + c_0h_0K_{0,1}) \tag{3.7}$$

holds, where $P, Q$ are public system parameters, $P_M = H_2(M)$, $K_{0,j} = H_1(\mathrm{ID}_0, j)$ for $j \in \{0, 1\}$ and $h_0 = H_3(M, \mathrm{ID}_0)$.

**Case 2**: verify an aggregate which contains the message $M$ and trust opinions from $n$ distinct nodes. Given an aggregate

$$A = [M, O_1, \ldots, O_n, c_0, c_1, \ldots, c_n, \mathrm{ID}_0, \mathrm{ID}_1, \ldots, \mathrm{ID}_n, S_{n+1}, T_{n+1}] \tag{3.8}$$

where $M$ is signed by vehicle $\mathrm{ID}_0$ and $(M, O_i)$ is signed by vehicle $\mathrm{ID}_i$, for $1 \leq i \leq n$, the verifier checks if the following equation holds:

$$\hat{e}(S_{n+1}, P) = \hat{e}(T_{n+1}, P_M)\hat{e}(Q, \sum_{i=0}^{n} c_iK_{i,0} + \sum_{i=0}^{n} c_ih_iK_{i,1}) \tag{3.9}$$

where $P_M = H_2(M)$, $K_{i,j} = H_1(\mathrm{ID}_i, j)$, for $i \in [0, n], j \in \{0, 1\}$, and $h_0 = H_3(M, \mathrm{ID}_0), h_i = H_3(M, O_i, \mathrm{ID}_i)$, for $i \in [1, n]$.

### 3.2.6 Correctness

**Case 1**: the verification of a signature on a pure message without trust opinions.

Verify $M_0 = [M, c_0, \mathrm{ID}_0, S_0, T_0]$.

$$\begin{aligned}
& \hat{e}(S_0, P) \\
= \; & \hat{e}(r_0 P_M + c_0 k_{0,0} + c_0 h_0 k_{0,1}, P) \\
= \; & \hat{e}(r_0 P_M, P)\hat{e}(c_0 k_{0,0} + c_0 h_0 k_{0,1}, P) \\
= \; & \hat{e}(P_M, r_0 P)\hat{e}(c_0 s K_{0,0} + c_0 h_0 s K_{0,1}, P) \\
= \; & \hat{e}(P_M, r_0 P)\hat{e}(c_0 K_{0,0} + c_0 h_0 K_{0,1}, sP) \\
= \; & \hat{e}(P_M, T_0)\hat{e}(c_0 K_{0,0} + c_0 h_0 K_{0,1}, Q) \\
= \; & \hat{e}(T_0, P_M)\hat{e}(Q, c_0 K_{0,0} + c_0 h_0 K_{0,1})
\end{aligned}$$

**Case 2**: the verification of an aggregate of trust opinions.

$$A = [M, O_1, \ldots, O_n, c_0, c_1, \ldots, c_n, \mathrm{ID}_0, \mathrm{ID}_1, \ldots, \mathrm{ID}_n, S_{n+1}, T_{n+1}].$$

$$
\begin{aligned}
& \hat{e}(S_{n+1}, P) \\
= \ & \hat{e}(\sum_{i=0}^{n} S_i, P) \\
= \ & \hat{e}(\sum_{i=0}^{n} r_i P_M + \sum_{i=0}^{n} c_i k_{i,0} + \sum_{i=0}^{n} c_i h_i k_{i,1}, P) \\
= \ & \hat{e}(\sum_{i=0}^{n} r_i P_M, P)\hat{e}(\sum_{i=0}^{n} c_i k_{i,0} + \sum_{i=0}^{n} c_i h_i k_{i,1}, P) \\
= \ & \hat{e}(P_M, \sum_{i=0}^{n} r_i P)\hat{e}(\sum_{i=0}^{n} c_i s K_{i,0} + \sum_{i=0}^{n} c_i h_i s K_{i,1}, P) \\
= \ & \hat{e}(P_M, \sum_{i=0}^{n} r_i P)\hat{e}(\sum_{i=0}^{n} c_i K_{i,0} + \sum_{i=0}^{n} c_i h_i K_{i,1}, sP) \\
= \ & \hat{e}(P_M, T_{n+1})\hat{e}(\sum_{i=0}^{n} c_i K_{i,0} + \sum_{i=0}^{n} c_i h_i K_{i,1}, Q) \\
= \ & \hat{e}(T_{n+1}, P_M)\hat{e}(Q, \sum_{i=0}^{n} c_i K_{i,0} + \sum_{i=0}^{n} c_i h_i K_{i,1})
\end{aligned}
$$

**Case 3**: the verification of a merged aggregate $A'$ from aggregate $A_1$ and $A_2$, as shown in Section 3.2.4.

$$
\begin{aligned}
& \hat{e}(S', P) \\
= \; & \hat{e}(S_1 + S_2, P) \\
= \; & \hat{e}(S_1, P)\hat{e}(S_2, P) \\
= \; & \hat{e}(T_1, P_M) \\
& \hat{e}(Q, (c_{0,1}K_{0,0} + c_{0,1}h_0K_{0,1}) + \sum_{i=1}^{p+k}(c_{i,1}K_{i,0} + c_{i,1}h_iK_{i,1})) \\
& \hat{e}(T_2, P_M) \\
& \hat{e}(Q, (c_{0,2}K_{0,0} + c_{0,2}h_0K_{0,1}) + \sum_{i=p+1}^{p+k+q}(c_{i,2}K_{i,0} + c_{i,2}h_iK_{i,1})) \\
= \; & \hat{e}(T_1 + T_2, P_M)\hat{e}(Q, \sum_{i=0}^{p+k+q}(c_i'K_{i,0} + c_i'h_iK_{i,1})) \\
= \; & \hat{e}(T', P_M)\hat{e}(Q, \sum_{i=0}^{p+k+q}(c_i'K_{i,0} + c_i'h_iK_{i,1}))
\end{aligned}
$$

## 3.3 Summary

In this chapter, we present a secure, efficient and scalable trust opinion aggregation scheme for our trust-based message evaluation and propagation model. Our aggregation scheme extends the identity-based aggregate signature algorithm, and thus achieves high efficiency and scalability without compromising security.

First and foremost of all, high security is reinforced throughout the aggregation process, either it being prior-aggregation or posterior-aggregation. Second, our aggregation method achieves high time efficiency since multiple signatures can be aggregated in one pass by summing them up mathematically. High space efficiency is made possible in that signatures are compacted into one aggregate signature, with additional information for verification minimized and redundancy eliminated. Third, our aggregation is flexible because there is no negative effect and little difference on whomever and whenever to perform the aggregation.

Further analysis of our aggregation scheme is illustrated in Chapter 4.4.

# Chapter 4

# Evaluation

In this chapter, we present an evaluation of our trust-based message evaluation and propagation model. System effectiveness and scalability are evaluated through simulation experiments on vehicular network environments. After that, the scalability and efficiency of our secure aggregation scheme are illustrated by comparing it to existing aggregation methods.

## 4.1 Experiment Setup

Implemented in C++, our simulation tool allows us to simulate real life traffic scenarios by employing real maps with vehicle entities following traffic rules, road limits, and a full list of customizable parameters defined in our trust model. Compared to other existing vehicular network simulation tools [33, 34, 35, 36], our tool is specially designed for trust modeling and cluster-based messaging among potentially thousands of nodes, and thus achieves more flexibility and consumes an incredibly low amount of computational resources.

We use a map of the East York area of Toronto, as shown in Figure 4.1. Roads are partitioned into multiple road segments, and vehicles are clustered geographically by road segments. We set the length of road segment to 0.5 kilometers, because peers within such

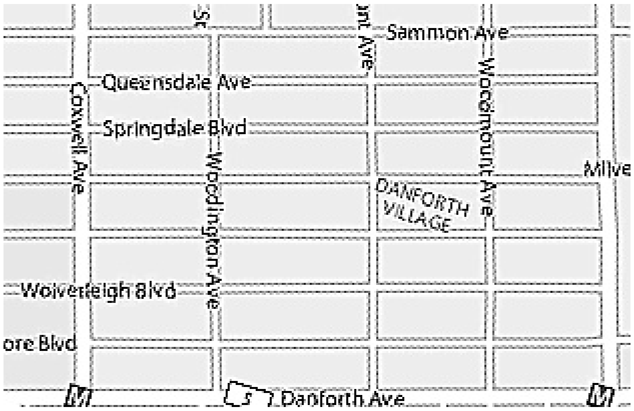

Figure 4.1: Map for Simulating VANET

a distance can reliably communicate with each other, according to [1]. Vehicles are moving in the map in any possible directions and in different speeds. Entering a new road segment indicates that the peer is switching from one cluster to another.

We list parameters for our trust modeling in Table 4.1. The purposes and details of these parameters have been introduced in Chapter 2. In our experiment, the sender weight factor $\gamma$ is set to 2 to double the weight of the sender in the message evaluation. Assuming that peer dishonesty is well tolerated by the system, we set the peer's trust threshold $\tau$ to 0.1, and the maximum error rate $\varepsilon$ in relay control model to 0.8. We also set $\beta/\alpha = 10$ based on the principle that trust is hard to build up but easy to tear down. Please note that all these values may not be optimal. In our evaluation below, our focus is to demonstrate the effectiveness of our model.

Additional parameters for simulating the vehicular network are listed in Table 4.2. Our experiment simulates a total number of 1125 vehicle entities. We set 2% of them as authority roles, such as police cars, road side units, and traffic controllers. The authority entities are fully reliable and trustworthy, and alway capable to provide other peers with valid observations and trust opinions.

Table 4.1: Parameters for Trust Modeling

| Parameter | Description | Value |
|---|---|---|
| $\gamma$ | sender weight factor | 2 |
| $\tau$ | trust threshold | 0.1 |
| $\varphi$ | action threshold | 0.2 |
| $\alpha$ | experience-trust increment factor | 0.01 |
| $\beta$ | experience-trust decrement factor | 0.1 |
| $\lambda$ | experience forgetting factor | 0.95 |
| $t_{max}$ | maximum time for experience (second) | 100 |
| $\varepsilon$ | error rate allowed for message relay | 0.8 |
| $m_d$ | maximum message propagation distance (km) | 5.5 |
| $m_t$ | message's longest time to live (second) | 150 |

Table 4.2: Parameters for Vehicular Network Simulation

| Parameter Description | Value |
|---|---|
| percentage of authority roles | 2% |
| average number of vehicles per cluster | 5 |
| probability of turning left/right at the cross | 0.2 |
| road segment length for one cluster | 0.5 km |
| maximum distance for trust opinion | 1 km |
| vehicle speed | [15, 30] m/s, road dependent |

Average number of vehicles per cluster is set to 5 to reflect the road situation in regular hours. The evaluation of effect of traffic density is left for future work. Vehicle speed is dependent on weather condition, traffic density, and speed limit of the road. To simplify our experiment, we assign a unique average speed to each road, where the vehicle's speed randomly varies ±10% from the average speed.

As mentioned in Chapter 2.2.1, the trust opinion is purely based on peer's local knowledge. In our experiment, we assume that all messages are observational. From this assumption, we further assume that the analysis module can provide trust opinions only when $\Delta d$, the geographical distance between the event and the peer, is smaller than $d_{max}$, the maximum distance for trust opinions. As a result, we can have the confidence value in the trust opinion determined by the geographical closeness: the closer the event is, the higher confidence value should be provided. In our experiment, confidence $c$ is calculated as

$$c = \begin{cases} (d_{max} - \Delta d)/d_{max} & \text{if } \Delta d < d_{max} \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

## 4.2 System Effectiveness

In this section, we evaluate the system effectiveness in terms of the capability for the system to mitigate against malicious messages and protect peers from being affected. We define the attack model as follow: attackers jeopardize the network by broadcasting misleading messages on fake events, such as "traffic congestion here", so as to cheat peers and maximize their own interest.

One of our evaluation metrics is the "average number of wrong actions per peer". An instance of "wrong action" indicates that one malicious message is trusted by a certain peer whose action module computes an answer of "trust" instead of "distrust" for the misleading event.

Table 4.3: Extra Parameters for Evaluation of System Effectiveness

| Parameter Description | Value |
|---|---|
| percentage of malicious peers | 10% |
| frequency of malicious messages | 30 seconds / message |
| analysis module's detection rate | uniform distribution, [0.05, 0.95] |

Extra parameters for evaluating system effectiveness are listed in Table 4.3. 10% of peers in the system are attackers, each of whom sends a malicious message after every 30 seconds, which is approximately the time of driving from one cluster to another. Considering that the analysis module generates trust opinions, we define the detection rate $d_{rate}$ as follow:

$$d_{rate} = Pr\{D|M\},\ D \text{ is a successful detection given a malicious message } M. \tag{4.2}$$

The analysis module generates a trust opinion of "distrust" upon a successful detection, otherwise "trust", maliciousness undetected. To better reflect the real situation, we assume that the capability to detect malicious messages varies among peers. In our experiment, the peer's detection rate follows the uniform distribution in $[0.05, 0.95]$, except for those authority roles, whose detection rate is highest and fixed to 1.

### 4.2.1 Effect of Trust Opinions

In this section, we begin with the effect of trust opinions. We demonstrate how much the system effectiveness is improved by comparing the average number of wrong actions under three trust opinion modes, as follow:

1. No trust opinions. The action module ignores all trust opinions and makes a local action by itself: when the peer is within the maximum distance where a trust opinion

is available, the action module follows the reaction of analysis module – if the analysis module trusts the message, the action model follows it, and vice versa. Otherwise, the peer simply trusts the message.

2. Trust opinions plus majority-based voting. The action module computes a local action from majority consensus, which can be reached by voting among all trust opinions without considering the trustworthiness of peers included in the trust opinions. The computation of the majority consensus follows Equation 2.6.

3. Trust opinions plus experience-based trust, which is what we have introduced in our model. A local action is computed from trust opinions, by not only considering the majority consensus, but also applying each peer's trustworthiness as the weight of its vote. This is what we have proposed in Equation 2.7.

The effect of three trust opinion modes is illustrated in Figure 4.2. We run the simulation for a duration of 60 minutes and sample the data after every 5 minutes. Each peer makes an average number of approximately 46 wrong actions if trust opinions are excluded. However, this number drastically drops to 19 (i.e. by 65%) if trust opinions are considered.

Compared to majority-based voting on trust opinions, the employment of experience-based trust further decreases the number of wrong actions globally as the system evolves. This is because once a peer obtains its own experience after being cheated by a malicious message, it will update the experience-based trust for those who have provided trust opinions for that message. The malicious sender's trust is shortly decreased by continuously sending malicious messages. At the same time, those weak detectors, who have a low detection rate and occasionally contribute wrong trust opinions, are becoming less trustworthy. As a result, the action module improves its accuracy and correct decision rate by mitigating against malicious senders and relatively promoting the weight of strong detectors.

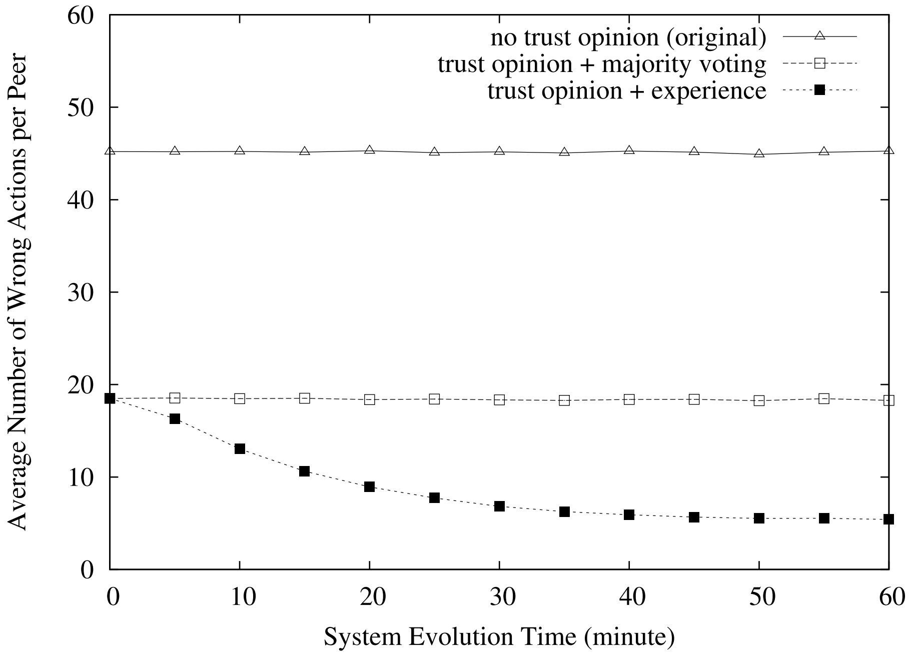

Figure 4.2: The Effect of Trust Opinions

### 4.2.2 Effect of Peer-to-Peer Trust

Next, we evaluate the effect of peer-to-peer trust model. Since peer-to-peer trust consists of two components: role-based trust and experience-based trust, we compare the system effectiveness under four cases: 1) Original situation. Actions are taken after conducting a majority-based voting on trust opinions, peer-to-peer trust ignored. 2) Role-based trust. The system will include authority roles, whose trust opinions are always followed by peers. 3) Experience-based trust. Peers are updating the trust for others as soon as personal experience becomes available. 4) Role-based trust plus experience-based trust. This is the combination of case 2 and 3.

In our trust model, the peer-to-peer trust is used in both the action module and the relay control model. In order to demonstrate the effect of peer-to-peer trust on action module, we evaluate the system effectiveness under two scenarios, namely without and with the relay control model, as shown in Figure 4.3 and Figure 4.4. In the absence of the relay control model, both good and bad messages are relayed to the furthest distance without being dropped.

Several conclusions can be drawn from the two figures.

1. Role-based trust improves the system effectiveness in both scenarios, due to the fact that authority roles are helpful in two ways. First, the trust opinion from authority is always followed by the action module of peers. Since authority is always trustworthy and of strong detection capability, the number of wrong actions is decreased. Second, the trust opinion from authority determines whether a message is to be relayed or dropped. When the relay control model is turned on, the propagation of malicious messages is limited and thus the negative effect is restricted. This explains why role-based trust decreases the number of wrong actions more in the scenario with relay control than the one without relay control.

2. Experience-based trust improves the system effectiveness as well. As explained earlier, peers accumulate experience and lower the experience-based trust for malicious peers and weak detectors. As a result, the average number of wrong actions is gradually decreased as system evolves. The performance of the both curves (exp and role+exp) is about the same after 60 minutes, which indicates that the experience-based trust plays a greater part in lowering the wrong decision rate than role-based trust, as system evolves for a longer time.

3. When there is no experience-based trust or authority role, the effect of relay control may be neglected, since there is little difference between the highest lines in both figures. The explanation lies in the inherent nature of trust opinions. In most cases, trust opinions aid peers to detect a malicious message, and thus further peers will not be affected even if the message is relayed. In case that trust opinions cannot help

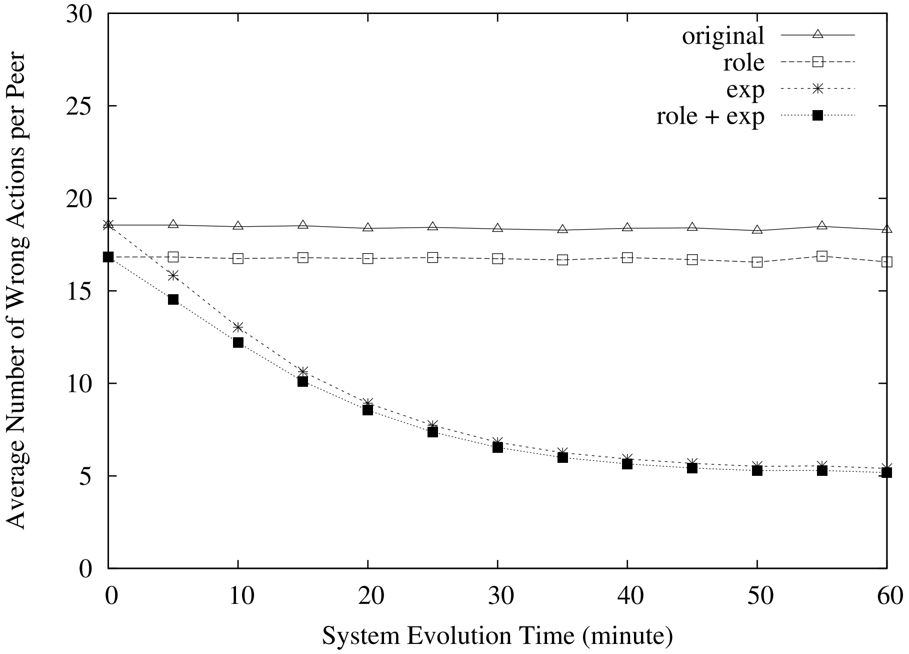

Figure 4.3: The Effect of Peer-to-Peer Trust (without Relay Control)

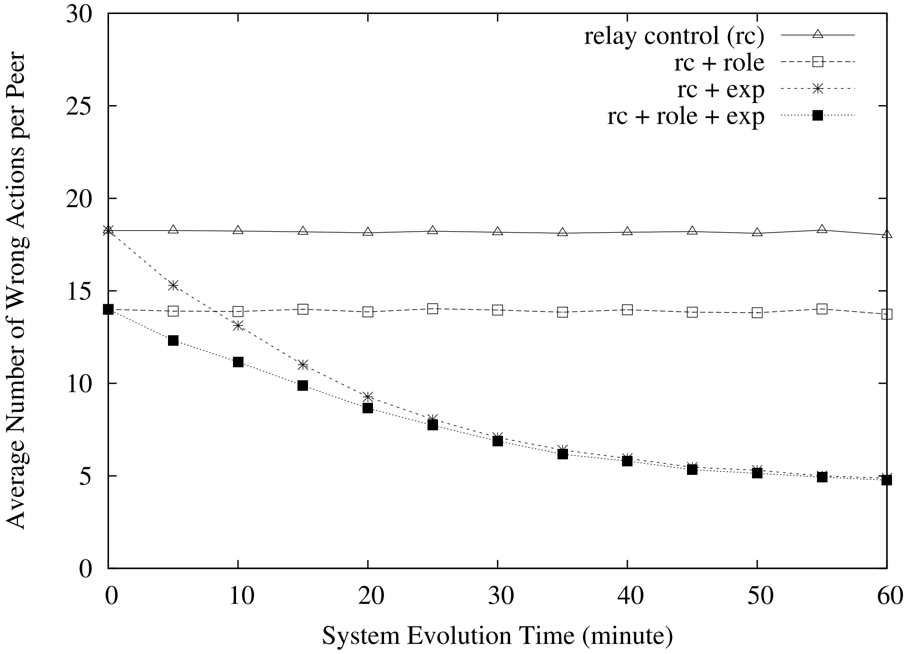

Figure 4.4: The Effect of Peer-to-Peer Trust (with Relay Control)

detect the maliciousness, most peers will fall victim due to trust opinions, and it is highly unlikely that message will be dropped in the middle because the relayer itself is highly likely to become a victim. At the same time, the experiment demonstrate one important observation that, even if every relayer is an attacker that forwards the malicious data, which amounts to no relay control, the system effectiveness is still preserved with little affection by malicious relaying behaviors.

### 4.2.3 Social Impact v.s. Honesty

Instead of using the average number of wrong actions per peer, we use another evaluation metric "number of deliveries" to demonstrate the system effectiveness from the perspective of social impact. One delivery of the sender is defined as one message reception by some receiver. We study the social impact of peers with different honesty levels. The honesty $h$ of a peer can be defined in possibly many ways, such as

$$h = (1 - \frac{\text{number of malicious messages sent}}{\text{number of messages sent}}) \times 100\% \tag{4.3}$$

We set three honesty levels in our experiment, namely 100%, 50%, and 0% honesty. Figure 4.5 shows the accumulative number of deliveries as system evolves. Three peers are randomly chosen from the system, each assigned to a different honesty level. After a simulation for 20 hours, it becomes obvious that the peer of 100% honesty has the largest number of deliveries, since its messages are trusted and relayed to the longest distance. The accumulative curve for 0% honesty ranks the lowest because most messages from fully dishonest peers are restricted from propagation. It grows even slower as system evolves, because peers become more experienced so that relay control model becomes more accurate in filtering malicious messages.

Figure 4.6 is an alternative graph showing the social impact versus peer honesty. We sample the number of deliveries for each hour and show the trend of each curve as system

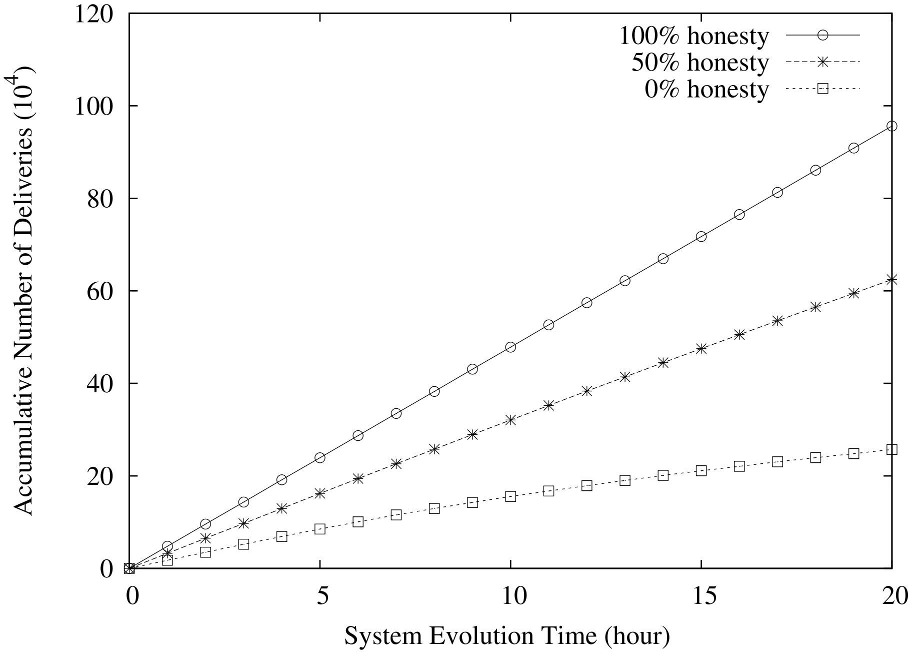

Figure 4.5: Accumulative Number of Deliveries with Evolution Time

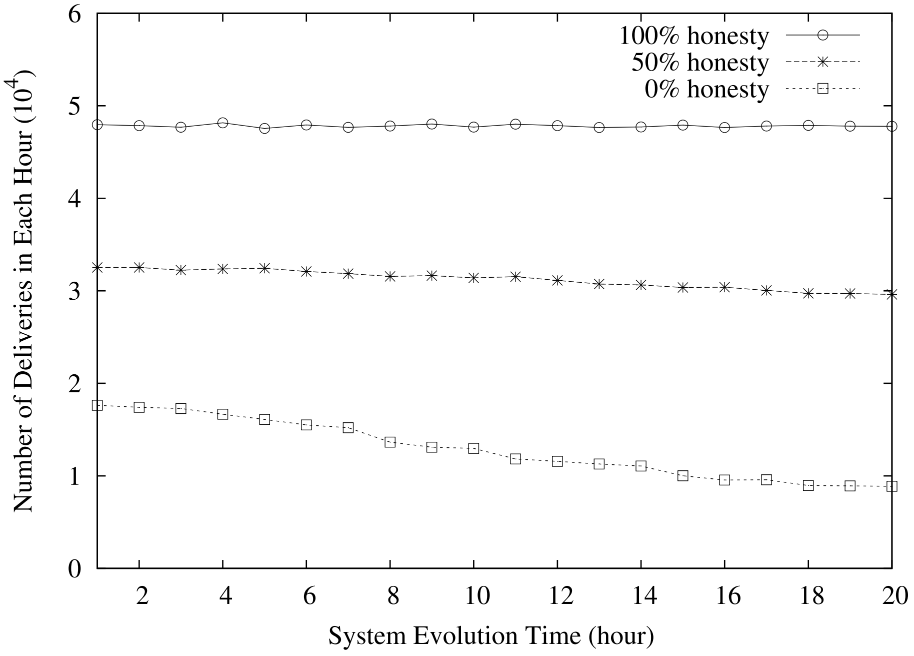

Figure 4.6: Number of Deliveries in Each Hour

evolves for 20 hours. Similar to the observations in Figure 4.5, dishonest peers would have less social impact than honest peers.

## 4.3 Scalability

In this section, we evaluate the system scalability. Our trust model can improve scalability by relay control model, which detects and filters malicious messages during propagation.

First of all, we define the attack model as follow: different from those defined in Section 4.2, the attackers here abuse their local vehicular network by sending spams frequently, which could be out-of-date information or repeated messages. Spams might not be misleading as that in previous attacker model, but they take up a certain portion of wireless resources and lower the utilization rate of available bandwidth, and thus should be treated as malicious.

Extra parameters for the evaluation of scalability are listed in Table 4.4. Assuming that spam is easier to detect than misleading messages as the pattern of spams has less variety, we increase the detection rate of analysis module globally by setting it to the uniform distribution from 0.4 to 1.0. We also include fewer attackers by setting the percentage of spammers to 1%, each of whom sends one spam every 5 seconds, which is much more frequent than misleading messages.

Table 4.4: Extra Parameters for Evaluation of Scalability

| Parameter Description | Value |
|---|---:|
| percentage of spammers | 1% |
| spam sending frequency | 5 seconds / message |
| detection rate of analysis module | uniform distribution, [0.4, 1.0] |

Our evaluation of scalability features three metrics: average propagation distance of spam, average number of received messages per peer, and global relay effectiveness. Each evaluation metric compares the performance among a subset of six predefined scenarios, shown as follows:

1. Original. Without regard to the trustworthiness of messages, they are simply relayed to the next hop, until the furthest allowed distance is reached.

2. Relay Control (RC). Whether a message is relayed or not is based on majority opinion, which does not take the role-based and experience-based trust into consideration. In otherwords, the weight of majority trust and distrust is computed as

   $$W_{trust} = \sum_{i \in P(trust)} c_i \tag{4.4}$$

   and

   $$W_{\neg trust} = \sum_{i \in P(\neg trust)} c_i \tag{4.5}$$

   compared to the Equation 2.14 and 2.15 in Chapter 2.

3. RC + Role. Role-based trust is taken into concern when relay control model computes the majority opinion. Trust opinions from authority roles are strictly taken as the relay decision.

4. RC + Exp. Experience-based trust is used as the weight of each peer in the computation of majority opinion.

5. RC + Role + Exp. A combination of scenario 3 and 4, which is our trust-based relay control model.

6. 100% Detection, which is the ideal case that each peer is capable to thoroughly detect any maliciousness, i.e. with detection rate $d_{rate} = 1.0$.

### 4.3.1 Average Propagation Distance of Spam

One of our evaluation metrics for scalability is the average propagation distance of spam. Assuming that the number of messages that can be relayed in a fixed period of time has an upper bound due to limited wireless channel resources, our system becomes more scalable as more normal messages can be relayed, which is achieved by detecting and controlling spam within a shorter distance.

The maximum propagation distance without relay control is 5.5 km as defined in our experiment. The relay control reduces the distance of spam by nearly half, as can be clearly observed from Figure 4.7. Authority roles further restrict the spam within approximately 2 kilometers away from origin, due to the fact that authority roles have assisted its cluster relayer to drop the spam at an earlier phase of propagation. From the curves of rc+exp and rc+role+exp, we can conclude that the experience-based trust plays a greater part in

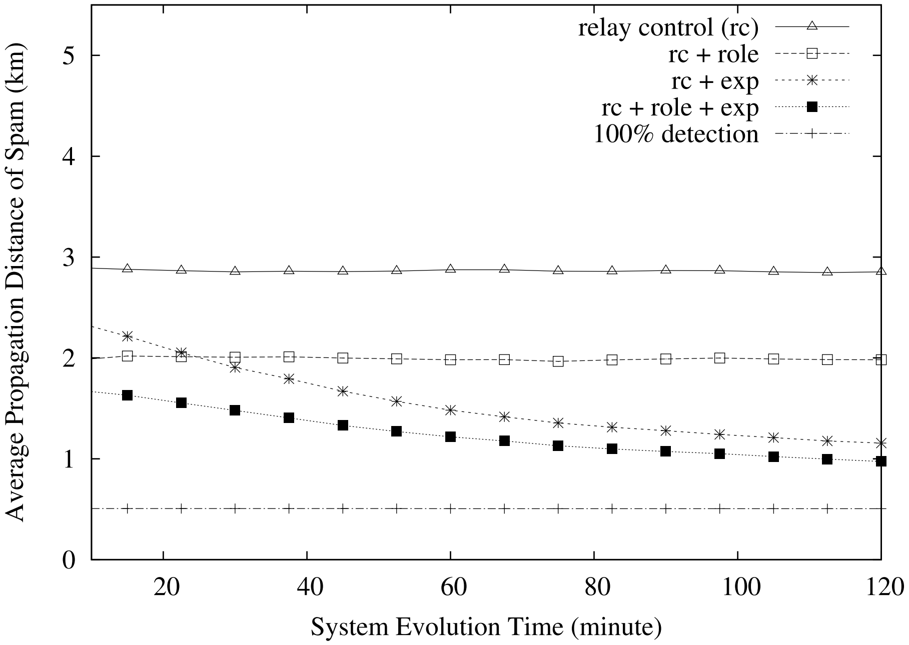

Figure 4.7: Average Propagation Distance of Spam

spam control as our experiment simulates for a longer time. This also explains why rc+role achieves better performance at the beginning but is sooner overwhelmed by rc+exp after 30 minutes. Besides, both curves demonstrate the trend of converging to the curve of 100% detection, under which scenario the spam is always dropped and never relayed to neighbor clusters, in other words, restricted within 0.5 kilometers – the length of cluster defined in our experiment.

The explanation to the trend of convergence lies in two aspects. First, peers acquire a better understanding of spammers, whose experience-based trust is gradually decreased to the extent that their messages are not trusted and not relayed. Second, the relay control model works more effectively in filtering spam as the weight of strong detectors' trust opinion is relatively promoted, as peers with weak detection power are falling less trustworthy as system evolves.

### 4.3.2 Average Number of Received Messages per Peer

Another evaluation metric is the average number of messages received for each peer. Two sets of experiments are conducted, as shown in the following.

In the first experiment, scalability is studied among six scenarios as the percentage of spam is adjusted from 0% to 100%. We track a total number of 14400 messages during a simulation for 2 hours. Experiment results are displayed in Figure 4.8.

Similar to the explanation to the evaluation of the average propagation distance of spam, the average number of received messages decreases as the percentage of spam increases, due to the relay control model. We notice that the rc+exp curve outperforms the rc+role curve when the percentage of spam is greater than 23%. This is because peers learn better about spammers during a fixed period of time, as more spams are available when the spam ratio is raised.

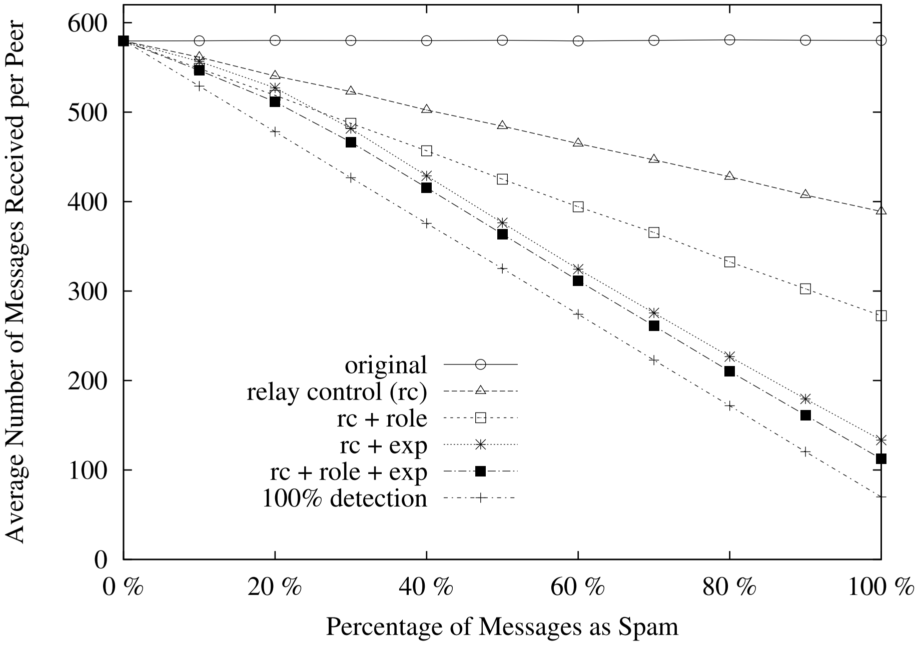

Figure 4.8: Average Number of Received Messages per Peer

In the second experiment, we evaluate the accumulative number of received spams per peer as system evolves. Simulation is conducted for a short duration of 50 minutes, as well as for a long duration of 230 minutes, as shown in Figure 4.9 and Figure 4.10.

From the simulation of a short time, we can see that the rc+exp curve is higher than the rc+role curve until approximately 33 minutes later. The explanation to this is that the experience-based trust plays a greater part than role-based trust when enough experience is obtained. After simulating for a longer time, the rc+exp curve and rc+role+exp curve grow almost as slowly as the 100% detection curve, which indicates that attackers are well identified with their spam detected and controlled.

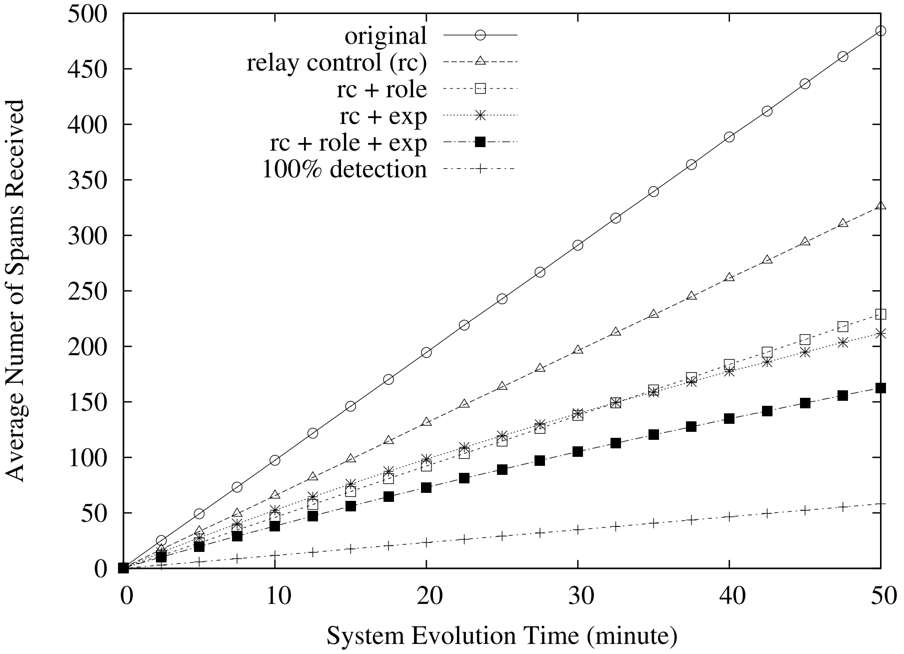

Figure 4.9: Average Number of Spams Received per Peer (Short Time)

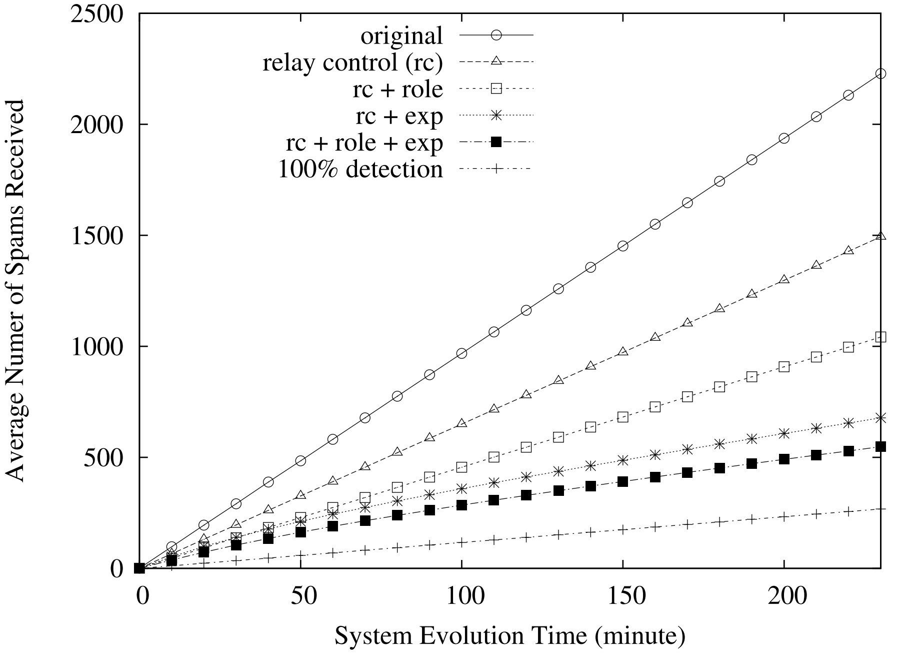

Figure 4.10: Average Number of Spams Received per Peer (Long Time)

### 4.3.3 Global Relay Effectiveness

We further evaluate system scalability using the third metric "global relay effectiveness", which measures how effectively that normal messages are relayed in the global presence of a considerable amount of spams. Specifically, we define the global relay effectiveness $R$ as

$$R = \frac{1}{N} \sum_{i=1}^{N} R_i \tag{4.6}$$

where $N$ is the total number of clusters, and $R_i$ is the relay effectiveness for a single cluster $C_i$, which is computed as

$$R_i = (1 - \frac{\text{number of relayed spams by } C_i}{\text{number of relayed messages by } C_i}) \times 100\% \tag{4.7}$$

The attackers here take a different strategy. Instead of sending spam every 5 seconds, the spammers jam the network in a random frequency but achieve the ultimate goal that the overall number of sent spams is approximately twice as that of messages by other peers.

We illustrate the global relay effectiveness in Figure 4.11. Attack is suspended until 5 minutes later. Since then, as shown in the original case, the effectiveness drops to around 42% after 120 minutes, and would finally converge to the expected value 33.3% as system evolves for an infinitely long time. Spams are restricted from dissemination after we apply the relay control model. Role-based trust always improves the effectiveness in that spams are further restricted. The global relay effectiveness stops ceasing and begins to recover after 35 minutes if the experience-based trust is applied, as can be observed from curve rc+role+exp and rc+exp. As peers become more experienced, the capability of the system to cope with spammers is strengthened.

## 4.4 Analysis of Trust Opinion Aggregation

We compare the security level, space efficiency, and time efficiency, among a multiple of existing secure aggregation methods: concatenate-signature-based (Conc.), onion-signature-

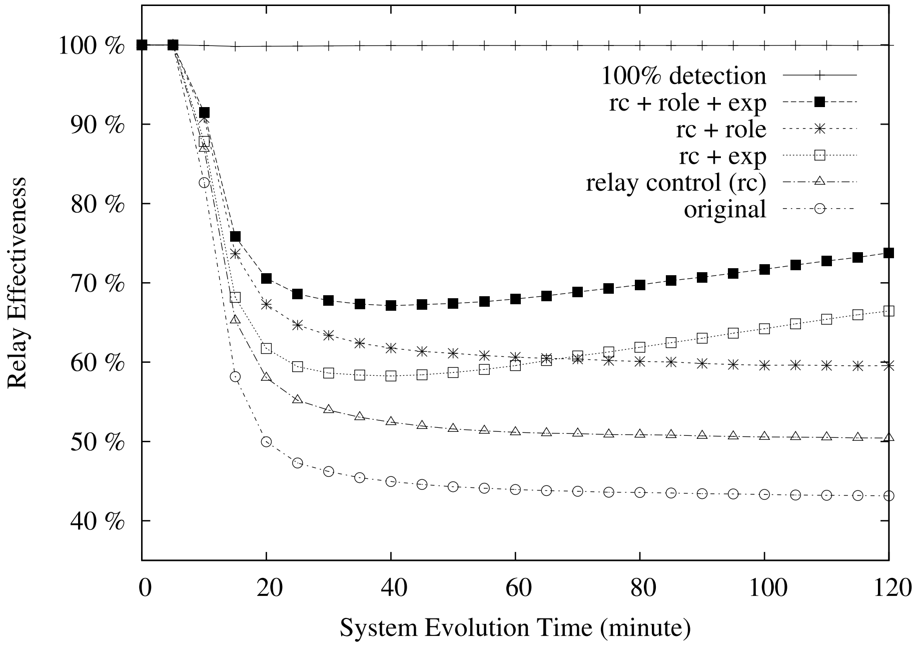

Figure 4.11: Global Relay Effectiveness

based (Onion), hybrid-signature-based (Hybrid), and our identity-based aggregation (ID-based) in the following. The details of the first three methods are illustrated in Chapter 5.2.1.

### 4.4.1 Security Level

The security level of concatenate signature and identity-based signature is high because neither signatures can be forged nor trust opinions can be repudiated without being detected. Onion signature has a low level of security because only two signatures are kept for verification – in case of two colluding peers, the data can be arbitrary generated or modified without being detected. The hybrid signature strikes a balance between the concatenation signature and onion signature by placing multiple pairs of signatures in the message. So we assign a medium security level to hybrid signatures.

### 4.4.2 Space Efficiency

The sender message "event, confidence, location, time" cost 21 bytes, if we assign 8 bytes to "event", 1 byte to "confidence", 8 bytes to "location", and 4 bytes to "time". Each trust opinion is assigned one byte (1 bit for reaction and 7 bit for confidence), and each vehicle identity in our region uses 3 bytes ($2^{24} \approx 1.6$ million vehicles). Given a message with $n-1$ trust opinions, the space cost is $21 + (n-1) + 3n = 4n + 20$ bytes, plus the cost of signatures, as shown below.

For non identity-based aggregation methods, if we apply the ECDSA signature scheme [32], which is adopted by IEEE 1609.2 standard [37], the signature size is 42 bytes. At the same time, one certificate must be transmitted along with one signature. If we use the certificate presented in IEEE 1609.2 standard [37], the certificate costs 125 bytes. For concatenate-signature-based aggregation, each vehicle identity would cost $42 + 125 = 167$ bytes for its signature plus certificate, so the size of aggregate message will be $4n + 20 + 167n = 171n+20$ bytes, since each identity occupies 167 bytes. Since onion-signature-based aggregation only keeps two signatures, the size of aggregate message is $4n + 20 + 167 \times 2 = 4n+354$ bytes. As for hybrid-signature-based aggregation, we define a degree $k \in [1, \lfloor n/2 \rfloor]$ to denote how many pairs of signatures are kept in the message. The aggregate size falls into $4n + 20 + 167 \times 2k = 4n + 334k + 20$ bytes.

For identity-based aggregation, if we want to achieve the same security level of ECDSA, the signature will be 42 bytes (21 bytes for $S$ and another 21 bytes for $T$). In order to maintain the list of $c_i$, $i \in [1, n]$ for $n$ identities, $2n$ bytes are needed. However, since the value of $c_i$ is usually small, less than $2n$ bytes might be consumed in practice if a compaction algorithm such as variable-length encoding is deployed. The ultimate size of aggregate message falls in an interval between $4n + 62$ and $6n + 62$ bytes.

We compare the aggregate size among these aggregation methods in Figure 4.12. The degree $k$ for hybrid signature is set to 4. Our identity-based aggregation method achieves

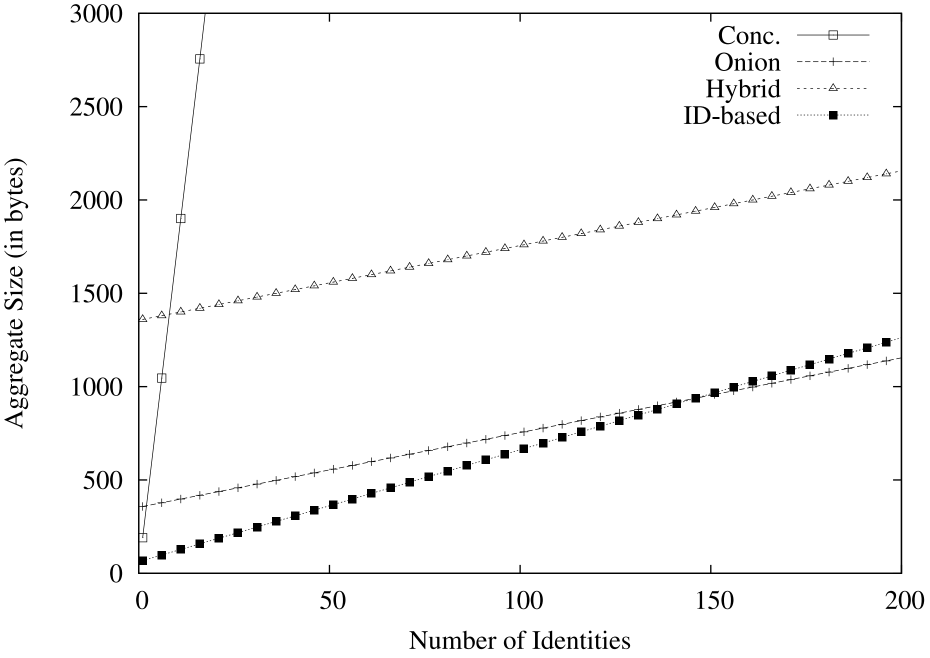

Figure 4.12: Space Efficiency

the best space efficiency when the number of identities is smaller than 146, although it is later surpassed by the onion-signature-based aggregation method, which, in practice, should not be employed due to its inherent disadvantage of low security and high aggregation delay.

### 4.4.3 Time Efficiency

Let $T$ [1] denote the average time cost for a peer to verify one single signature and generate a trust opinion. The concatenate-signature-based and identity-based aggregation takes $2T$ to generate an aggregate message with $n$ trust opinions from $n$ peers – for the first $T$ the message is broadcast to each peer who verifies the message, signs and attaches its

[1] In practice, the verification time $T_v < 500$ ms with 200 identities, according to Zhang et. al [38]. The generation time of trust opinions is depending on the assumed analysis module.

own opinion; for the second $T$ these opinions are sent back to the aggregator who verifies and combines these opinions with its own trust opinion into the new aggregate message. For the onion-signature and hybrid-signature-based aggregation, it requires an aggregation sequence of $n$ peers where messages are verified, attached an opinion and signed from the head peer to the tail peer, and thus the total cost is $nT$.

As a summary of this section, the comparison of secure aggregation methods is shown in Table 4.5.

Table 4.5: Comparison of Secure Aggregation Methods

| | Security Level | Aggregate Size (in bytes) | Aggregation Delay |
|---:|---:|---:|---:|
| Conc. | high | $171n + 20$ | $2T$ |
| Onion | low | $4n + 354$ | $nT$ |
| Hybrid | medium | $4n + 334k + 20$ | $nT$ |
| ID-based | high | $4n + 62 \sim 6n + 62$ | $2T$ |

## 4.5 Summary

In this chapter, we demonstrate the system effectiveness and scalability through an experimental simulation. We also illustrate the advantage of our identity-based aggregation method by comparing its space and time efficiency to other existing methods.

Throughout the experiment, we realize that the system effectiveness and scalability are dependent on two important factors: a) peer's local knowledge, and b) control of malicious messages. With a better understanding and stronger control of malicious messages, system effectiveness is improved as less peers are effected. At the same time, the system becomes more scalable as malicious messages are more likely to be detected and dropped.

As the peer accumulates more experience and derives local decisions from a more reliable set of trust opinions, a better local knowledge is acquired. Although the accumulation of experience can be time-consuming, the experience-based trust demonstrate a strong effect on system effectiveness and scalability. The existence of authority and role-based trust improves the quality of trust opinions from which less wrong actions are derived.

The control of malicious messages is implemented by our relay control model. Although it only improves system effectiveness slightly due to the dominating effect of trust opinions, the relay control model greatly improves system scalability as it detects and filters malicious messages.

As for the aggregation efficiency, our identity-based aggregation method achieves high space efficiency and time efficiency without degrading the level of security. Compared to other existing aggregation methods, it is the only applicable method for our trust-based message evaluation and propagation model.

# Chapter 5

# Related Work

In this chapter, we survey the work related to our trust-based message evaluation and propagation framework. We mainly focus on two important perspectives: trust modeling and data aggregation in mobile/vehicular ad-hoc networks.

## 5.1 Trust Modeling

The work on trust modeling in mobile/vehicular ad-hoc networks can be categorized as two subsets: 1) trust establishment, which addresses the issues in how to define and establish the trust between two mobile entities; 2) trust utilization, which deals with the problems in how to apply the currently available set of trust to achieve the specific goal, under the proposed application scenario.

### 5.1.1 Trust Establishment

The paper [39] is one of the earliest work on trust modeling in mobile ad-hoc networks. It identifies several important properties of trust establishment, such as the specification of admissible types of evidence, the generation, distribution and evaluation of trust evidences in mobile ad-hoc networks, which are different from those of trust establishments in Internet.

A trust establishment scheme called Hermes is introduced in [40] with the objective of reliable delivery and routing in mobile ad-hoc networks. The trust between two neighboring peers is modeled as trustworthiness, which takes confidence into concern and can be computed using a Bayesian approach based on an empirical set of first-hand observations of packet forwarding behavior of neighboring peers. Choosing the best route between the source and destination amounts to determining the shortest path, where the weigh of the path is computed from a set of peer-to-peer trust between the peers within the path. The work in [41] extends [40] in that recommendation trust is introduced to model the trust between two non-neighboring peers. The trust to a remote peer is established by collecting recommendations from a set of nodes.

Similar to [40, 41], the work in [42] models the trust evaluation as a path optimization problem on a direct graph where each peer is a vertex and trust between two neighboring peer is an edge. The author introduces the semiring-based evaluation metric that features in two binary operators, $+$ and $*$. The former operator is used for trust computation over a path of peers while the latter one is to compute the optimal aggregated trust among a set of available paths. Two operators can be reloaded via different semiring algorithms so as to adapt to various conditions.

Sun et al. [43, 44] present an information theoretic framework to quantitively measure and model the trust in ad-hoc networks. It first defines three trust axioms: a) concatenation propagation of trust does not increase trust; b) multi-path propagation of trust does not reduce trust; c) trust based on multiple observations from a single source should not be higher than the multiple observations from multiple independent sources. An entropy-based trust model and a probability-based model are introduced in which the author shows how to compute trust along a path as well as the overall trust among a set of paths. The trust value between two neighboring nodes are based on observations. Third, the paper discusses how to obtain, evaluate, and update trust when it comes to ad hoc routing. Briefly speaking, each node maintains its trust record about other nodes. The source node finds multiple routes to the destination node when the source node wants to establish a route to the destination node. The source node evaluates the packet-forwarding trustworthiness of each node on a route, either by its own trust record or by requesting recommendations from other nodes. After the best trustworthy route is chosen, data is transmitted. After the transmission, the source node updates the trust records based on its observation of route quality. Compared to their work, our system model requires that each peer maintains a list of other peers and derives their trust from messaging and posterior-experience.

The proposed methodologies in [40, 41, 42, 43, 44] may not work effectively in vehicular networks because in practice the trust cannot be established, maintained or retrieved unless a reliable route is available, which is hard to establish in a highly dynamic environment such as vehicular networks. Previous trust modeling endeavors in ad-hoc networks, such as improving routing quality, and deriving reliability between arbitrary peers, may become effortless when it comes to vehicular networks, due to its two basic inherent properties. First, peer connection is ephemeral as vehicle entities are moving fast with little time for interaction. Second, interaction between two entities is highly infrequent due to peer's mobile nature and broad real world environment. As a result, trust establishment is difficult and even if trust can be established between two vehicle entities, it may be out-of-date and uncertain.

Considering the uncertainty property of trust establishment in mobile ad-hoc networks, Balakrishnan et al. [45] expresses the notion of ignorance during the establishment of trust relationships between mobile nodes. Subjective logic based model is employed to denote the trust as a three dimensional metric: belief, disbelief, and uncertainty. The uncertainty represents the ignorance between two nodes. Such representation is useful since an existing peer may not have a record of past evidence towards a newcomer/stranger peer, in which case assigning an arbitrary trust value could bring about problems. Compared to their work, our trust model proposes a different methodology which takes two factors into deployment, namely, a set of fixed roles and the aging factor in experience-based trust. Roles decrease the uncertainty in that their trust are fixed. Aging factor ages the trust between two entities until new interactions are available.

### 5.1.2 Trust Utilization

The work in [40, 41, 42, 43, 44] targets the goal of reliable packet delivery from the perspective of source routing in mobile ad-hoc networks. In vehicular networks, similar attempts to apply trust to routing may face challenges in that most of proposed source routing algorithms in mobile ad-hoc networks may not work well in vehicular networks, plus the fact that trust establishment is more challenging in the vehicular environment.

The work in [46] employs trust into the data evaluation in vehicular networks. In contrast to traditional views of entity-level trust, it proposes data-centric trust establishment that deals with the evaluation of trustworthiness of messages from other peers instead of vehicle entities themselves. A set of trust metrics are defined to represent the data trust from multiple dimensions, such as vehicle's security status, peer type and event type. Based on Bayesian interference and Dempster-Shafer Theory, they evaluate the decision logic which outputs the trust values of various data regarding a particular event. Although it shares some commonalities with ours, such as the employment of data trust over peer trust, their work has two shortcomings that fundamentally impedes their model from being applied in reality. First, trust relationship in entities can never be reliably established and only ephemeral trust in data is built, and thus the data-centric trust has to be established again and again for each event, which may not be applicable to situations under the sparse environment where few interactions/data are available. Second, there lacks a bridge that effectively connects the data trust and peer trust together, in that the peer trust is assumed to be always available in their decision logic and peers are not responsible for their misbehavior, neglecting the dynamics such as dishonest and malicious attackers.

Golle et al. [19] propose an approach to detect and correct malicious data in vehicular networks. They assume that each vehicular peer is maintaining a model which consists of all the knowledge that the peer has about the network. Data is trusted if it agrees with the model with a high probability. Otherwise, a heuristic is invoked to restore data consistency by finding the simplest explanation possible. Multiple explanations are ranked and the peers accept the data if it is consistent with the most highly ranked one(s). However, they assume that each vehicle has the global knowledge of the network and solely evaluates the validity of data, which may not be feasible in practice. Our work also provides high resistance and security against malicious entities using a fundamentally different way of message evaluation. Instead of relying on an assumed model and seeking explanations, messages in our model are evaluated in a distributed and collaborative fashion by collecting multiple opinions during their propagation.

Minhas et al. [20] demonstrate an expanded trust management model for agents in vehicular networks, which features in the role-based trust and experience-based trust as the evaluation metric of vehicular entities, as well as the majority consensus model, which is computed from peer trust and used as an evaluation metric for data trustworthiness. Based on the work of Minhas et al., we extend their trust management model in three aspects. First, we have designed a trust propagation scheme that effectively establishes the trust relationship between vehicular agents through message dissemination and trust opinion aggregation so that trust can be more widely established among more agents. Second, we work on the details of the aggregation algorithm that ensures a secure and efficient trust opinion aggregation. Third, we have proposed an action model that aids vehicle agents to evaluate the trustworthiness of each message instead of relying on the majority consensus model, which is adapted and used in our relay control model.

## 5.2 Data Aggregation

Some recent work on comfort or safety applications in vehicular networks is largely dependent on data dissemination and aggregation. While discussing them all in detail would exceed the scope of our work, we first focus on several major applications that use aggregation techniques to achieve a specific goal. After that, we point out the stringent security necessity and list several recent solutions on secure data aggregation in vehicular networks, each with pros and cons.

Self-Organizing Traffic-Information System (SOTIS) [47, 48] assumes each vehicle is equipped with a map based on which the road is partitioned into multiple segments. Each vehicle broadcasts data about the traffic information of the road segments within its transmission range. The broadcast data are collected and aggregated into one average value, together with a time stamp indicating the freshness. The system believes an aggregate is better if it has a newer time stamp, but the time stamp is generated upon aggregation instead of observation of an event. Due to the way the aggregated value is computed, the system can only handle numerical values so that not all types of data can be aggregated. And also, aggregate may not best reflect the road situation for that all messages are equally treated.

Similar to [47, 48], a TrafficView system is modeled in [10] where the traffic view is recovered by vehicles distributing their speed and position information. Aggregation is improved in three aspects: 1) observations are weighted by the distance between the aggregator and the sender and the oldest time stamp among all observations is chosen as the time stamp for the aggregate; 2) a cost-based or ratio-based algorithm is applied to selectly aggregate a subset of messages and discard the rest, given a limited size of the final aggregate; 3) information aging is taken into consideration while updating the record. However, such aggregation cannot apply to complex city road scenarios because the distance is a one dimensional metric and thus cannot recover the whole planar traffic view.

A hierarchical aggregation scheme of traffic information is introduced in [49]. Instead of carrying specific values, the aggregate features in the Flajolet-Martin sketch used as a probabilistic approximation. The employment of Flajolet-Martin sketches brings in the advantage that aggregates are duplicate insensitive, allowing multiple aggregates with their values combined and merged into one aggregate without increasing the aggregate size. At the same time, such probabilistic aggregation has two drawbacks. First, aggregate cannot accurately present a value, especially when it is small. Second, when multiple observations are compacted and merged into one value, the quantitative information such as the number of observations is discarded, ignoring the fact that an aggregate of more observations is usually more reliable than an aggregate of less ones.

Lochert et al. [50] proposes a similar aggregation scheme where the travel time for a road segment is hierarchically aggregated to estimate the overall time between two landmarks. Their design relies on an infrastructure and thus suffers from its availability and scalability and cannot be used here.

### 5.2.1 Secure Aggregation

The above mentioned aggregation methodologies mainly target at basic requirements in a particular application scenario, assuming that the network is under a reliable, fully-trusted, and peer-friendly environment. Security related issues were not taken into consideration, which renders the network vulnerable when confronted by a malicious party. A secure aggregation mechanism requires that data are aggregated without being modified. Any maliciousness can be detected and prevented from hampering the system at an early stage.

Before secure aggregation was first studied for vehicular networks, several work attempted to satisfy the security demands of aggregation in sensor networks. The work [51] assumes a static binary-tree alike network topology for data aggregation. Data authentication is based on a pre-shared temporary key which may be compromised and delayed in its design. Przydatek et al. [52] provides an *aggregate-commit-prove* approach where the aggregator computes the collected data and then commits the aggregation results to the home server. The server later verifies the results by performing an interactive proof with the aggregator. Yang et al. [53] presents a hop-by-hop aggregation method based on the principles of *divide-and-conquer*, which partitions the whole network into subgroups in a recursive and hierarchical fashion, and *commit-and-test*, which recursively aggregates data from subgroups and identifies suspicious groups based on multiple group aggregates.

We realize that these aggregation protocols for sensor networks may not be suitable for vehicular networks because those protocols may only work under the assumption of a static network topology and reliable multiple peer-to-peer interactions, both of which are not available in vehicular networks due to great peer dynamics and lack of a prevalent infrastructure.

Instead, current approaches in vehicular networks usually require a digital signature on each message being aggregated so that neither a message can be maliciously repudiated nor could an attacker disseminate bogus information without being traced.

Raya et al. [29] illustrate a secure aggregation scheme which aggregates messages and signatures from different parties signing the message. The model features in three types of aggregate signature schemes, as shown in Figure 5.1.

1. *Concatenate Signature*: the vehicle signs a message $m$ and rebroadcasts the new message to the aggregator, who collects all signed messages from all signers. Signatures $S_i(m)$ are concatenated and appended to $m$ so that all signatures are independent from each other. However, the verification of an aggregate takes up $n$ times of signa-

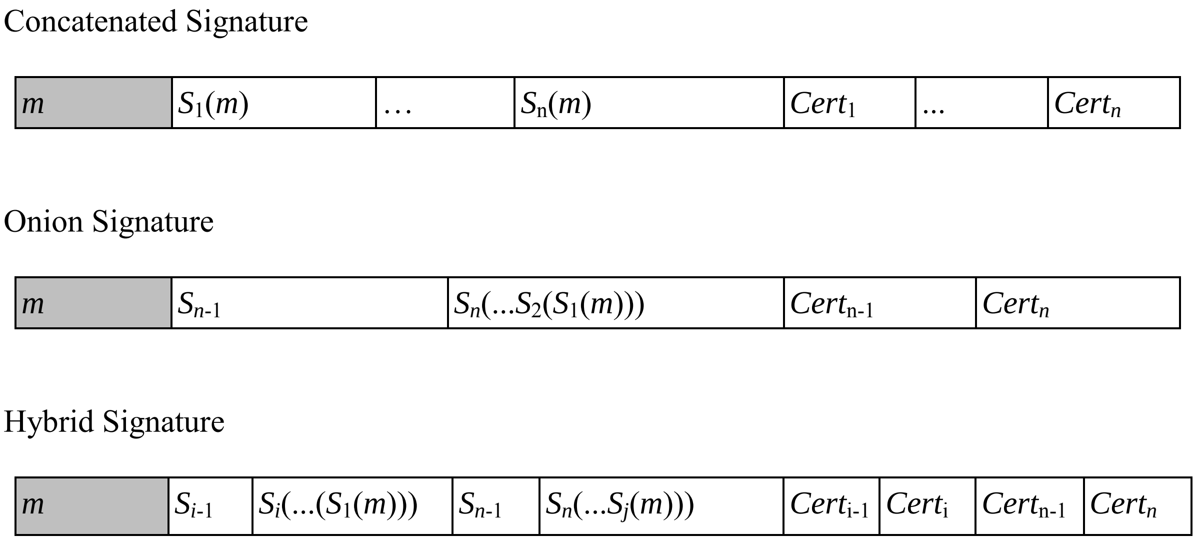

Figure 5.1: Concatenated, Onion, and Hybrid Signature

ture verifications and also, the size of an aggregate grows linearly with the number of signatures and certificates, which are space costly.

2. *Onion Signature*: an onion signature borrows the idea of onion routing. A message is over-signed during its propagation, i.e, a signer signs the signed message from its predecessor and forwards the new message to its successor. The $n$-th node verifies the signature $S_{n-1}$ before over-signing $S_{n-1}$. Such protects data validity and requires at most two signatures, reduces the communication overhead by concatenate signatures, and yet increases the computation overhead because of the inherent property of onion routing. At the same time, it opens the vulnerability window by allowing two colluding attackers to modify messages without being detected.

3. *Hybrid Signature*: although onion signature reduces the communication cost of concatenate signature, it goes insecure when nodes collude. A hybrid signature strikes a balance between the two by concatenating several onion signatures, each at a given

depth, which strengthens the security of signatures by hardening the collusion. However, there still exists a possibility that signatures can be compromised.

And also, onion signature and hybrid signature are based on a strong assumption that signature aggregation is done in a sequential order, i.e., the $n$-th signer must aggregate its own signature into the aggregate signature formed by its previous $n-1$ signers. An ordered signing sequence would reflect a chained trust relationship among peers, which may not be available or appreciated in vehicular networks. In practice, a chained aggregation may be not acceptable due to its availability and aggregation delay; we would prefer a non-chained aggregation strategy, as shown in Figure 5.2. We proposed an aggregation scheme that is non-chained and extends current identity-based aggregate signature algorithm [31], as is discussed in the following context.

Moreover, all these aggregate signatures make use of asymmetric cryptography and rely on public key infrastructure (PKI). In other words, signature verification requires additional information – the signer's public keys and certificates (e.g. $Cert_i$ in Figure 5.1). Such information, that is usually in big size compared to the message itself, needs to be carried along with the signature because it is not preferable to assume that any verifier would have already kept all public keys and certificates in its local repository. As a result, aggregation achieves high security with a trade-off of great space overhead, rending it inefficient and reversely degrading system performance.

An identity-based aggregate signature scheme [31] comes into existence in recent years and overwhelms the above signature schemes in three aspects. First, multiple signatures are compacted into one single signature while the message can be still verified. Second, instead of storing/carring public keys and certificates, the verifier only needs to obtain a list of identities of signers, which are much smaller than public keys and certificates. Third, no matter how messages are aggregated, the scheme does not imply a trust chain among peers and thus the final aggregate signature remains unique. With overall information

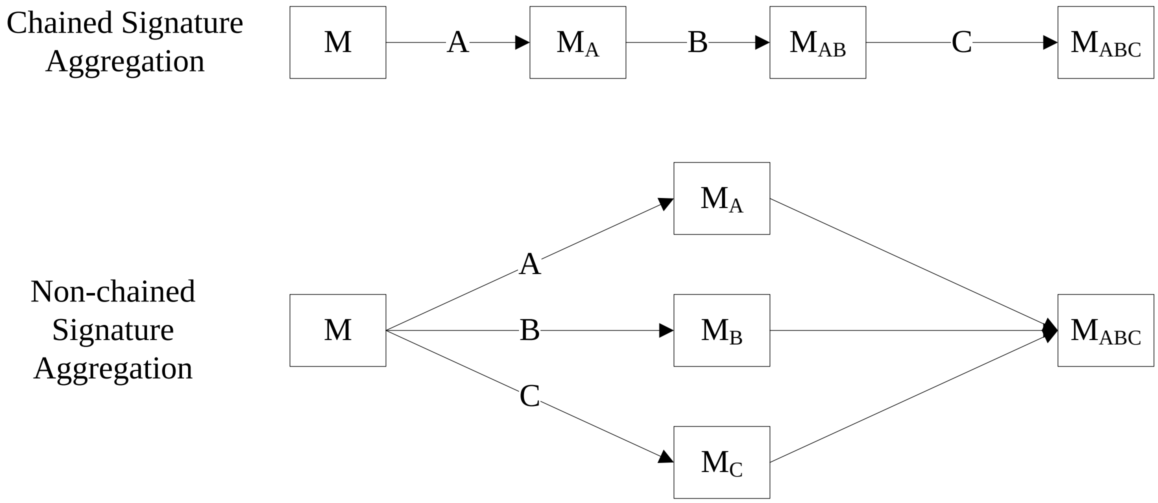
Chained Signature Aggregation
M
A
$M_A$
B
$M_{AB}$
C
$M_{ABC}$
Non-chained Signature Aggregation
M
A
B
C
$M_A$
$M_B$
$M_C$
$M_{ABC}$

Figure 5.2: Chained Aggregation and Non-chained Aggregation. $M$ is a message, $A, B, C$ are peers who signs $M$. In a chained aggregation scheme, $M$ is signed sequentially by $A$, $B$ and $C$ into $M_{ABC}$; in a non-chained aggregation scheme, $M$ is signed by each peer $i$ into $M_i$, and then all $M_i$ are aggregated by any third party into $M_{ABC}$.

required for verification minimized, it achieves high efficiency without compromising the security constraint. However, such a scheme requires $n$ different signatures from $n$ distinct signers. In other words, when it comes to merging two duplicate signed messages from the same signer, which is often true under a certain aggregation strategy, aggregation would fail because the aggregated signature cannot pass verification.

Based on [31], Zhu [54] introduces an aggregated emergency message authentication scheme for vehicular networks, where a vehicle randomly generates a pseudonym as its identity each time a new emergency message is signed. Signed messages can be aggregated and verified even if there are duplicate ones. However, their work cannot be applied into our system because there is no way to trace the signer given the signed message. This leaves ample space for an attacker who signs arbitrary fake messages under different pseudo-identities so as to forge the fact that multiple peers have participated in signing a message.

A batch message verification method [38] is proposed, where multiple signatures on different messages are compacted into one signature, so that the road side unit (RSU) can verify a bunch of messages in one pass. The identity of the signer can be traced by the central authority (CA) but the system relies on the temper-proof device which might be either hacked or working improperly.

In summary, we need an aggregation scheme that achieves efficiency without compromising security. An extended identity-based aggregation scheme is presented in our work, which resolves the stringent needs for security, efficiency, scalability and flexibility.

# Chapter 6

# Conclusion and Future Work

## 6.1 Conclusion

We present in this thesis a novel message evaluation and propagation framework in vehicular networks, where a set of trust metrics, such as trust opinion, experience-based trust and role-based trust, are used to model the data quality and the relationship between peers. Our proposed message evaluation approach is conducted in a distributed, collaborative fashion during the message propagation, and effectively increases the overall data reliability and system effectiveness by proactively detecting malicious data. Our message propagation method features in a trust-based relay control model that filters malicious data and promotes network scalability. Moreover, we illustrate an identity-based data aggregation scheme that collects and aggregates trust opinions in a secure and efficient way. Experimental and analytical results demonstrate that our approach works effectively and efficiently for the domain of vehicular networks.

## 6.2 Future Work

As shown in Chapter 2, our framework depends on the existence of trust opinions, which are generated by the assumed analysis module. The design of such a module would involve much consideration from the perspective of hardware design, such as the design of temper-proof devices, car sensors and human-computer interactive interfaces.

Our trust aggregation and message propagation model is based on the cluster-based routing scheme, which has not resolved all possible issues yet. Considering that cluster-based routing has addressed little concern to security, we might further consider the presence of malicious relayers who intentionally drop messages, and design a set of detection and revocation mechanisms, to see how our system copes with possible attacks.

Due to the practical concerns of trust opinion aggregation, e.g. it is very likely that only a subset of trust opinions are available for aggregation due to complex road settings, we may evaluate the system effectiveness when partial trust opinions are available. Moreover, since we mentioned in Chapter 4.4.2 that a variable-length encoding algorithm could further compact the size of an aggregated message, we will develop such an algorithm and evaluate the improvement in space efficiency.

As for the experimental evaluation, more complex scenarios may be employed in our simulation. For example, considering that our current metric for evaluating system effectiveness is the average number of wrong actions by each peer, we may examine the global awareness and resistance to a pre-defined set of fake events/data. The vehicle density is an important aspect whose effect on our trust model may be studied as well. Besides, more sophisticated attack models, such as peer collusion, will be evaluated in the future.

# References

[1] Jing Zhu and S. Roy. MAC for dedicated short range communications in intelligent transport system. *Communications Magazine, IEEE*, 41(12):60–67, Dec. 2003. 1, 44

[2] B. Hofmann-Wellenhof, H. Lichtenegger, and J Collins. *Global Positioning System: Theory and practice*. Springer, 1993. 1

[3] P. Mohapatra, Chao Gui, and Jian Li. Group communications in mobile ad hoc networks. *Computer*, 37(2):52–59, Feb 2004. 1

[4] L. Wischhof, A. Ebner, and H. Rohling. Information dissemination in self-organizing intervehicle networks. *Intelligent Transportation Systems, IEEE Transactions on*, 6(1):90–101, March 2005. 1

[5] Maxim Raya and Jean-Pierre Hubaux. The security of vehicular ad hoc networks. In *SASN '05: Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks*, pages 11–21, New York, NY, USA, 2005. ACM. 1

[6] Rongxing Lu, Xiaodong Lin, Haojin Zhu, Pin-Han Ho, and Xuemin Shen. ECPP: Efficient conditional privacy preservation protocol for secure vehicular communications. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pages 1229–1237, April 2008. 1

[7] Qing Xu, Tony Mak, Jeff Ko, and Raja Sengupta. Vehicle-to-vehicle safety messaging in DSRC. In *VANET '04: Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, pages 19–28, New York, NY, USA, 2004. ACM. 1

[8] Tamer ElBatt, Siddhartha K. Goel, Gavin Holland, Hariharan Krishnan, and Jayendra Parikh. Cooperative collision warning using dedicated short range wireless communications. In *VANET '06: Proceedings of the 3rd international workshop on Vehicular ad hoc networks*, pages 1–9, New York, NY, USA, 2006. ACM. 1

[9] Sumair Ur Rahman and Urs Hengartner. Secure crash reporting in vehicular ad hoc networks. In *Security and Privacy in Communications Networks and the Workshops, 2007. SecureComm 2007. Third International Conference on*, pages 443–452, Sept. 2007. 1

[10] Tamer Nadeem, Sasan Dashtinezhad, Chunyuan Liao, and Liviu Iftode. Trafficview: Traffic data dissemination using car-to-car communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 8:2004, 2004. 1, 71

[11] Seung-Hoon Lee, Uichin Lee, Kang-Won Lee, and M. Gerla. Content distribution in vanets using network coding: The effect of disk i/o and processing o/h. In *Sensor, Mesh and Ad Hoc Communications and Networks, 2008. SECON '08. 5th Annual IEEE Communications Society Conference on*, pages 117–125, June 2008. 1

[12] A. Nandan, S. Das, G. Pau, M. Gerla, and M.Y. Sanadidi. Co-operative downloading in vehicular ad-hoc wireless networks. In *Wireless On-demand Network Systems and Services, 2005. WONS 2005. Second Annual Conference on*, pages 32–41, Jan. 2005. 1

[13] Anand Prabhu Subramanian, Vishnu Navda, Pralhad Deshpande, and Samir R. Das. A measurement study of inter-vehicular communication using steerable beam direc-

tional antenna. In *VANET '08: Proceedings of the fifth ACM international workshop on VehiculAr Inter-NETworking*, pages 7–16, New York, NY, USA, 2008. ACM. 1

[14] Bret Hull, Vladimir Bychkovsky, Yang Zhang, Kevin Chen, Michel Goraczko, Allen Miu, Eugene Shih, Hari Balakrishnan, and Samuel Madden. CarTel: a distributed mobile sensor computing system. In *SenSys '06: Proceedings of the 4th international conference on Embedded networked sensor systems*, pages 125–138, New York, NY, USA, 2006. ACM. 1

[15] Fan Li and Yu Wang. Routing in vehicular ad hoc networks: A survey. *Vehicular Technology Magazine, IEEE*, 2(2):12–22, June 2007. 2, 23

[16] Maxim Raya, Daniel Jungels, Panos Papadimitratos, Imad Aad, and Jean pierre Hubaux. Certificate revocation in vehicular networks. Technical report, 2006. 2

[17] Jason J. Haas, Yih-Chun Hu, and Kenneth P. Laberteaux. Design and analysis of a lightweight certificate revocation mechanism for VANET. In *VANET '09: Proceedings of the sixth ACM international workshop on VehiculAr InterNETworking*, pages 89–98, New York, NY, USA, 2009. ACM. 2

[18] M. Raya, P. Papadimitratos, I. Aad, D. Jungels, and J.-P. Hubaux. Eviction of misbehaving and faulty nodes in vehicular networks. *Selected Areas in Communications, IEEE Journal on*, 25(8):1557–1568, Oct. 2007. 2

[19] Philippe Golle, Dan Greene, and Jessica Staddon. Detecting and correcting malicious data in VANETs. In *VANET '04: Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, pages 29–37, New York, NY, USA, 2004. ACM. 2, 70

[20] Umar Farooq Minhas, Jie Zhang, Thomas Tran, and Robin Cohen. Towards expanded trust management for agents in vehicular ad-hoc networks. In *Proceedings of the In-*

*ternational Conference on Knowledge Intensive Multi-Agent Systems (KIMAS)*, 2009. 6, 12, 17, 19, 20, 23, 70

[21] Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-Molina. The Eigentrust algorithm for reputation management in p2p networks. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 640–651, New York, NY, USA, 2003. ACM. 13

[22] Thomas Thanh Tran. *Reputation-oriented reinforcement learning strategies for economically-motivated agents in electronic market environments*. PhD thesis, Waterloo, Ont., Canada, Canada, 2004. Adviser-Cohen, Robin. 14, 19

[23] R. A. Santos, A. Edwards, R. M. Edwards, and N. L. Seed. Performance evaluation of routing protocols in vehicular ad-hoc networks. *Int. J. Ad Hoc Ubiquitous Comput.*, 1(1/2):80–91, 2005. 21

[24] C.R. Lin and M. Gerla. Adaptive clustering for mobile wireless networks. *Selected Areas in Communications, IEEE Journal on*, 15(7):1265–1275, Sep 1997. 21

[25] B. Das and V. Bharghavan. Routing in ad-hoc networks using minimum connected dominating sets. In *Communications, 1997. ICC 97 Montreal, 'Towards the Knowledge Millennium'. 1997 IEEE International Conference on*, volume 1, pages 376–380 vol.1, Jun 1997. 21

[26] Jie Wu. Dominating-set-based routing in ad hoc wireless networks. pages 425–450, 2002. 21

[27] J. Blum, A. Eskandarian, and L. Hoffman. Mobility management in ivc networks. In *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*, pages 150–155, June 2003. 21

[28] T.D.C. Little and A. Agarwal. An information propagation scheme for VANETs. In *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE*, pages 155–160, Sept. 2005. 21

[29] Maxim Raya, Adel Aziz, and Jean-Pierre Hubaux. Efficient secure aggregation in VANETs. In *VANET '06: Proceedings of the 3rd international workshop on Vehicular ad hoc networks*, pages 67–75, New York, NY, USA, 2006. ACM. 21, 73

[30] John R. Douceur. The sybil attack. In *IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems*, pages 251–260, London, UK, 2002. Springer-Verlag. 30

[31] Craig Gentry and Zulfikar Ramzan. Identity-based aggregate signatures. In *Public Key Cryptography*, pages 257 – 273, 2006. 31, 32, 75, 76

[32] Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the weil pairing. In *ASIACRYPT '01: Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security*, pages 514–532, London, UK, 2001. Springer-Verlag. 31, 33, 34, 62

[33] Christian Lochert, Andreas Barthels, Alfonso Cervantes, Martin Mauve, and Murat Caliskan. Multiple simulator interlinking environment for IVC. In *VANET '05: Proceedings of the 2nd ACM international workshop on Vehicular ad hoc networks*, pages 87–88, New York, NY, USA, 2005. ACM. 43

[34] David R. Choffnes and Fabián E. Bustamante. An integrated mobility and traffic model for vehicular wireless networks. In *VANET '05: Proceedings of the 2nd ACM international workshop on Vehicular ad hoc networks*, pages 69–78, New York, NY, USA, 2005. ACM. 43

[35] Rahul Mangharam, Daniel S. Weller, Daniel D. Stancil, Ragunathan Rajkumar, and Jayendra S. Parikh. GrooveSim: a topography-accurate simulator for geographic routing in vehicular networks. In *VANET '05: Proceedings of the 2nd ACM international workshop on Vehicular ad hoc networks*, pages 59–68, New York, NY, USA, 2005. ACM. 43

[36] Amit Kumar Saha and David B. Johnson. Modeling mobility for vehicular ad-hoc networks. In *VANET '04: Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, pages 91–92, New York, NY, USA, 2004. ACM. 43

[37] IEEE Standard 1609.2. IEEE trial-use standard for wireless access in vehicular environments - security services for applications and management messages. *IEEE Std 1609.2*, pages 0–1–105, 2006. 62

[38] Chenxi Zhang, Rongxing Lu, Xiaodong Lin, Pin-Han Ho, and Xuemin Shen. An efficient identity-based batch verification scheme for vehicular sensor networks. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pages 246–250, April 2008. 63, 77

[39] Laurent Eschenauer, Virgil D. Gligor, and John Baras. On trust establishment in mobile ad-hoc networks. In *In Proceedings of the Security Protocols Workshop*, pages 47–66. Springer-Verlag, 2002. 66

[40] C. Zouridaki, B. L. Mark, M. Hejmo, and R. K. Thomas. A quantitative trust establishment framework for reliable data packet delivery in MANETs. In *SASN '05: Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks*, pages 1–10, New York, NY, USA, 2005. ACM. 67, 68, 69

[41] Charikleia Zouridaki, Brian L. Mark, Marek Hejmo, and Roshan K. Thomas. Robust cooperative trust establishment for MANETs. In *SASN '06: Proceedings of the fourth*

*ACM workshop on Security of ad hoc and sensor networks*, pages 23–34, New York, NY, USA, 2006. ACM. 67, 68, 69

[42] G. Theodorakopoulos and J.S. Baras. On trust models and trust evaluation metrics for ad hoc networks. *Selected Areas in Communications, IEEE Journal on*, 24(2):318–328, Feb. 2006. 67, 68, 69

[43] Yan Sun, Wei Yu, Zhu Han, and K.J.R. Liu. Trust modeling and evaluation in ad hoc networks. In *Global Telecommunications Conference, 2005. GLOBECOM '05. IEEE*, volume 3, pages 6 pp.–, Nov.-2 Dec. 2005. 67, 68, 69

[44] Yan Lindsay Sun, Wei Yu, Zhu Han, and K.J.R. Liu. Information theoretic framework of trust modeling and evaluation for ad hoc networks. *Selected Areas in Communications, IEEE Journal on*, 24(2):305–317, Feb. 2006. 67, 68, 69

[45] Venkat Balakrishnan, Vijay Varadharajan, and Uday Tupakula. Subjective logic based trust model for mobile ad hoc networks. In *SecureComm '08: Proceedings of the 4th international conference on Security and privacy in communication netowrks*, pages 1–11, New York, NY, USA, 2008. ACM. 68

[46] M. Raya, P. Papadimitratos, V.D. Gligor, and J.-P. Hubaux. On data-centric trust establishment in ephemeral ad hoc networks. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pages 1238–1246, April 2008. 69

[47] L. Wischhof, A. Ebner, and H. Rohling. Information dissemination in self-organizing intervehicle networks. *Intelligent Transportation Systems, IEEE Transactions on*, 6(1):90–101, March 2005. 71

[48] L. Wischoff, A. Ebner, H. Rohling, M. Lott, and R. Halfmann. SOTIS: a self-organizing traffic information system. In *Vehicular Technology Conference, 2003. VTC 2003-*

*Spring. The 57th IEEE Semiannual*, volume 4, pages 2442–2446 vol.4, April 2003. 71

[49] Christian Lochert, Björn Scheuermann, and Martin Mauve. Probabilistic aggregation for data dissemination in VANETs. In *VANET '07: Proceedings of the fourth ACM international workshop on Vehicular ad hoc networks*, pages 1–8, New York, NY, USA, 2007. ACM. 72

[50] Christian Lochert, Björn Scheuermann, Christian Wewetzer, Andreas Luebke, and Martin Mauve. Data aggregation and roadside unit placement for a VANET traffic information system. In *VANET '08: Proceedings of the fifth ACM international workshop on VehiculAr Inter-NETworking*, pages 58–65, New York, NY, USA, 2008. ACM. 72

[51] Lingxuan Hu and D. Evans. Secure aggregation for wireless networks. In *Applications and the Internet Workshops, 2003. Proceedings. 2003 Symposium on*, pages 384–391, Jan. 2003. 72

[52] Bartosz Przydatek, Dawn Song, and Adrian Perrig. SIA: secure information aggregation in sensor networks. In *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 255–265, New York, NY, USA, 2003. ACM. 73

[53] Yi Yang, Xinran Wang, Sencun Zhu, and Guohong Cao. Sdap: A secure hop-by-hop data aggregation protocol for sensor networks. *ACM Trans. Inf. Syst. Secur.*, 11(4):1–43, 2008. 73

[54] Haojin Zhu, Xiaodong Lin, Rongxing Lu, Pin-Han Ho, and Xuemin Shen. AEMA: An aggregated emergency message authentication scheme for enhancing the security of vehicular ad hoc networks. In *Communications, 2008. ICC '08. IEEE International Conference on*, pages 1436–1440, May 2008. 76

[55] Sergio Marti, T. J. Giuli, Kevin Lai, and Mary Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 255–265, New York, NY, USA, 2000. ACM.

[56] Zhigang Wang, Lichuan Liu, MengChu Zhou, and N. Ansari. A position-based clustering technique for ad hoc intervehicle communication. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 38(2):201–208, March 2008.