A Certificateless Anonymous Authenticated Announcement Scheme in Vehicular Ad Hoc Networks*

Amizah Malip †[1], Siaw-Lynn Ng[1], and Qin Li[2]

[1]*Information Security Group, Department of Mathematics, Royal Holloway University of London, TW20 0EX United Kingdom*
[2]*Department of Computer Engineering, Nanyang Technological University, Singapore 639798*

Abstract

Vehicular ad hoc networks (VANETs) provide a safer driving environment by allowing vehicles to broadcast safety related messages and inform neighboring vehicles regarding traffic and road conditions. Safety can only be achieved if transmission of messages are reliable. However, verification of reliability may violate privacy. On the other hand, it is desirable that malicious or defective vehicles can be identified and revoked. In this paper, we propose a new protocol using certificateless signature and reputation system to achieve the sometimes contradictory requirements of a reliable, private and accountable VANET message announcement scheme.

Keywords: vehicular ad hoc networks, reliability, privacy, accountability

1 Introduction

Vehicular ad hoc networks (VANETs) provide a safer driving environment by allowing vehicles to communicate with each other (V2V) or with road side infrastructure (V2I). They permit a vehicle to broadcast *announcements* to neighbouring vehicles to improve driving safety or traffic efficiency. The drivers benefit from the system as information on traffic congestion, accidents, potholes or slippery roadways will

*This is the peer reviewed version of the following article: *A certificateless anonymous authenticated announcement scheme in vehicular ad hoc networks*. Q. Li, A. Malip and S.-L. Ng. Security and Communication Networks Vol. 7, no. 3, March 2014, pp. 588 - 601, which has been published in final form at [doi: 10.1002/sec.760]. This article may be used for non-commercial purposes in accordance With Wiley Terms and Conditions for self-archiving

†Supported by SLAB/SLAI University of Malaya and Ministry of Higher Education Malaysia.

allow receiving vehicles to respond quickly by assessing the situation and making decisions accordingly.

In order to benefit from the tools of VANETs, the system requires secure communication protocols. Safety can only be achieved if messages are *reliable*, that is the messages reflect actual situations. Receiving vehicles must have some assurance that the message is:

- sent by a legitimate source,
- unmodified,
- not a lie.

The first two goals are commonly achieved by some digital signature schemes, while the third is not as straightforward. Different techniques have been proposed to achieve the third requirement. These include *threshold method* and *trust- and reputation-based models*. In a threshold method, message truthfulness is established by knowing that messages of the same content were sent by many distinct legitimate senders. Meanwhile, in trust and reputation models, a message is considered reliable if the message generator has "good" reputation.

However, the verification of message reliability may reveal some information about the identity of the sending vehicle. It may also allow malicious entities to trace sending vehicles by comparing messages with the same signature. This may violate user privacy. On the other hand, complete anonymity may not be desirable, since malicious or defective vehicles need to be identified for maintenance and law enforcement purposes. In this paper we will consider how these often contradictory goals of reliability, privacy and accountability may be satisfied.

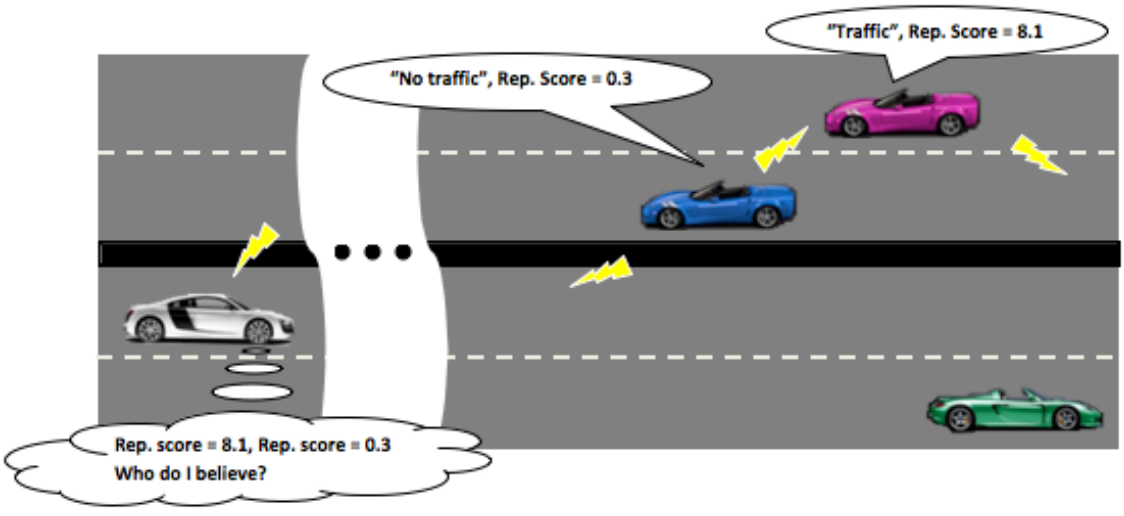

Figure 1: A scenario in the CLS announcement scheme

We first give an overview of our solution before discussing these security requirements in greater detail. In our previous work [20], we proposed a reputation

system for VANETs. The reliability of a message generated by a vehicle is reflected by its reputation score. A message is considered reliable if the message generator has sufficiently high reputation. A vehicle consistently announcing reliable messages increases its reputation score. Meanwhile, the reputation score of an unreliable message generator decreases, by means of a feedback mechanism. A feedback report consists of a numerical score, which represents a receiver's evaluation of the reliability of the relevant message. The reputation system is managed by a centralised *reputation server* (RS) whose role includes admitting and revoking vehicles from the system. It is also responsible for managing, storing and updating reputation scores based on the feedback received. This scheme does not provide much privacy since the identities of all sending and reporting vehicles are made public. The provision of privacy in a reputation system is a nontrivial matter. Here we design a novel anonymous announcement scheme for VANET using the reputation system of [20] which will satisfy all the following goals:

Reliability :

> A vehicle will only rely on an announcement if it can be certain that this announcement was broadcast by a legitimate and present vehicle (sender authenticity) without unauthorised modification (message integrity). In addition, there is a high probability that the event has happened (message truthfulness). For feedback reporting, unmodified feedbacks (report integrity) are reported by legitimate vehicles (reporter authenticity) present at the announcement. The system should also be able to tolerate a small fraction of internal adversaries (system robustness).

Privacy :

> An announcement or feedback report cannot be linked to its source by an unauthorised observer (anonymity). Different announcements or feedback reports by one vehicle cannot be linked by an unauthorised observer (unlinkability). Clearly the reputation server has to link feedback reports and scores to vehicles. However, it may not be necessary to make the *contents* of the announcements known to the server (content confidentiality).

Accountability :

> Misbehaviour can be traced to a source (traceability) and a source cannot deny having sent the message (non-repudiation). Furthermore, misbehaved vehicles should be prevented from future participation in the network or his identity reported to the authorities (revocation). This applies to both message generator and feedback reporter.

In order to simultaneously achieve these goals, we construct a scheme using a certificateless signature scheme (CLS). The adoption of CLS does not require the use of certificates, which can be unweildy in a large VANET environment, and yet

does not have the inherent key escrow problem of identity-based signature. This allows for an efficient as well as secure and anonymous announcement scheme for VANETs.

The organisation of the paper is as follow. In Section 2, we discuss some published literature closely related to our work. We describe the preliminaries of the scheme in Section 3 and present our certificateless signature scheme in Section 4. Section 5 presents the operation of the scheme and the scheme is analysed in Section 6. Finally, Section 7 concludes the paper.

2 Related Work

This section will focus on some vehicle-generated announcement schemes that address the issues of reliability, privacy and accountability in VANETs and discuss arising issues.

In a large VANET environment where the absence of initial trust is assumed, a vehicle will only trust an announcement if it can be certain that the message:

- was broadcast unmodified from a legitimate vehicle,
- is not a falsehood.

The first requirement is commonly solved by means of digital signature technique [2, 7, 9–12, 14, 19, 21, 22, 25, 27]. Signing a message using valid credentials (for instance, public key certificates) from a trusted authority (TA) assures sender authenticity and message integrity. However, the overhead associated with certificate management such as storage, distribution, revocation and computational cost of certificate generation and verification, may be prohibitive [15]. In addition, if the public keys used to verify signatures are bound to identities in the credentials, it will allow tracking and profiling of signing vehicles.

We consider two aspects of privacy: *anonymity* and *unlinkability*. The problem of anonymity can be addressed by using group signature [7, 9, 11, 21], where the group is the set of all cars in the system. Group signatures allow each group member to sign on behalf of the group. A verifier knows the signer belongs to a group without being able to associate the signature with a particular signer. Another approach to privacy is by using pseudonymous public keys. The use of randomly chosen and changing identifiers, referred to as *pseudonyms*, has been proposed in [7, 14, 19, 27]. Messages can be linked if it is verified using the same pseudonym. However, linking messages will be more challenging if vehicles change and update pseudonyms regularly. Pseudonyms can be preloaded or self-generated. The former method gives rise to other problems such as certificate management and large storage space. To eliminate these problems, [7] proposed that each vehicle generates its own pseudonyms. The rate at which pseudonyms are updated depends on various factors such as the degree of privacy required by a vehicle. However, an issue associated with this technique is the problem of distinguishability of message origin, which we discuss later in the section.

In our work, we adopt a certificateless signature scheme that allows entity authentication and message integrity while eliminating the necessity of certificates. A vehicle periodically preloads a set of credentials and its reputation scores from the TA and generates its own pseudonymous key pairs. Each signing key is updated after its short lifetime expires, hence messages signed with a signing key is only linkable over its short validity period. This approach is in common with most pseudonym schemes. However, unlike the pseudonym schemes of [7, 14, 19, 27], our scheme vastly reduces the overhead associated with certificate management.

To achieve the second requirement, different techniques have been proposed. These include the *threshold method* [7, 9–11, 14, 19, 21, 25, 27] and *trust-* and *reputation-based models* [2, 12, 22].

The threshold method is used in many recent announcement protocols [7, 9–11, 14, 19, 21, 25, 27]. It is based on the assumption that an announced event is more likely to be true if many distinct vehicles are reporting the same event within a time interval. Hence if a vehicle receives a number of messages exceeding some threshold, it will assume the event is true and act upon it. The threshold may be fixed system-wide [10, 25] or flexible [9, 11, 14, 19]. It has to be chosen carefully. It should not be too high that insufficient endorsement hinders the user from acting upon the information. It should not be too low that the decision may be affected by the presence of adversaries.

In order to adopt the threshold method, message origins has to be distinguishable. When a vehicle receives a number of announcements of a certain event, it needs to be sure that each message originated from a different source. This directly contradicts the requirement for unlinkability. However, such verification is needed to avoid the Sybil attack [13] where a vehicle signs the same message multiple times using different identities. Some types of digital signature allow distinguishability of origin [9, 11], while others do not [7, 21, 27]. In our scheme, we do not rely on multiple messages to evaluate reliability. We also do not require distinguishability of message origin. We consider a message to be reliable only if the message generator has sufficiently high reputation. Not only do we require less computation, this also allows a vehicle to make decisions and act upon messages quickly.

Several trust- and reputation-based models were proposed, for instance [2, 12, 22]. Dötzer et al. [12] proposed a reputation model using an approach called *opinion piggybacking*. Upon receiving a message, each receiving vehicle appends its own opinion about the reliability of the message before it forwards the message to neighboring vehicles. The opinion may be based on the content of the message or previous aggregated opinions attached to the message. To generate an opinion, a receiving vehicle has to verify, compute and aggregate all previous opinions appended to the message. This is a significant computational burden on receiving vehicles. Our scheme is computationally efficient as evaluation of message reliability only requires a receiving vehicle to verify one signature provided that the message generator has sufficient high reputation.

Minhas et al. [22] employs three variation of trust models to evaluate message reliability: *role-based trust*, *majority-based trust* and *experience-based trust*. Role-based trust assumes vehicles with certain predefined role, such as the traffic patrol or law

enforcing authorities, have higher trust value compared to other vehicles. Each vehicle possesses a certificate from a trusted authority, for identification and authentication purposes. Meanwhile, majority-based trust is similar to the threshold method we discussed earlier. In experience-based trust, the trustworthiness of a vehicle is evaluated based on how truthful they were in their past direct interaction. Such a model requires a vehicle to establish long term relationship with other vehicles. This may not be practical in a large VANET environment. This also implies that a vehicle is required to store information of other vehicles encountered in the past, which may cause storage problem. A similar approach of experience-based trust was proposed by Patwardhan et al. in [2]. In our scheme, we propose a more practical model. A vehicle may provide feedback, which represents an evaluation of the reliability of the message received. These feedbacks accumulate to a vehicle's reputation score. Hence, short term encounters lead to a long term trust, represented by a vehicle's reputation score.

However, the issue of privacy was not addressed in these schemes [2, 12, 22]. In addition, they adopt a decentralised infrastructure where the problem of accountability arise. In our scheme, we utilise the already existing trusted infrastructure of VANETs that allow us to design a secure authenticated announcement scheme that preserve privacy while achieving the property of accountability.

3 Preliminaries

In this section, we review some fundamental background for a certificateless signature scheme.

3.1 Certificateless Cryptography

Public Key Cryptography. In public key cryptography (PKC), each user has a public and a private key. The private key is kept secret while the public key is published. A public key of a user is associated with the user by a certificate, that is, a signature of a trusted Certificate Authority (CA) on the public key. This allows the receiver to be sure that the public key that they have is the correct public key for the sender. A receiver who wants to use the public key must verify the corresponding certificate for the validity of the key. Hence, we require a public key infrastructure - a series of trusted third parties that can be relied upon to vouch for the connection between an identity and a particular public key. Inevitably this feature causes a CA to require a large amount of storage and computing time managing the certificates.

Identity-based Cryptography. To avoid the certificate management problem, Shamir [28] introduced the concept of identity based cryptography (ID-PKC). The idea was then practically deployed by Boneh and Franklin in [4]. An identity-based scheme removes the need for a public key infrastructure by setting an entity's public key to be equal to its digital identity. A key generator center (KGC) generates the entity's private key using a master secret. An inherent problem of ID-PKC is thus the "key

escrow" problem: the KGC knows the user's private keys and has to be complete trusted.

Certificateless Cryptography. In 2003, Al-Riyami and Paterson [1] introduced the concept of certificateless public key cryptography (CL-PKC) which eliminates the use of certificates in PKC and solves the key escrow problem in ID-PKC. The basic idea of CL-PKC is that the user constructs a public/private key pair by combining a value generated by a TA using its master key with a random secret value generated by the user. We describe such a signature scheme in the next section.

3.2 A Certificateless Signature Scheme

3.2.1 Pairings and Computational Problems

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ and be an additive group and a multiplicative group, respectively, of the same prime order q. Let P denote a random generator of $\mathbb{G}_1$ and $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ denote a bilinear map which is typically constructed by Weil or Tate pairing with properties:

1. Bilinearity: $e(Q, W + Z) = e(Q, W) \cdot e(Q, Z)$ and $e(Q + W, Z) = e(Q, Z) \cdot e(W, Z)\ \forall\ Q, W, Z \in \mathbb{G}_1$. Consequently, we have $e(aP, bQ) = e(P, Q)^{ab}, \forall\ P, Q \in \mathbb{G}_1$ and $\forall a, b \in \mathbb{Z}_q^*$.

2. Non-degeneracy: $\exists\ P, Q \in \mathbb{G}_1$ such that $e(P, Q) \neq 1$.

3. Computability: there exists an efficient algorithm to compute $e(aP, bQ)\ \forall\ P, Q \in \mathbb{G}_1$.

We assume that the discrete logarithm problem (DLP) is hard in both $\mathbb{G}_1$ and $\mathbb{G}_2$. The DLP is defined as follows: Given a generator P of a cyclic additive group $\mathbb{G}$ with order q, and $Q \in \mathbb{G}^*$, find an integer $a \in \mathbb{Z}_q^*$ such that $Q = aP$. In addition, we assume the computational Diffie-Hellman problem (CDHP) in $\mathbb{G}_1$. The CDHP is defined as follows: Given a generator P of a cyclic additive group $\mathbb{G}$ with order q, and given (aP, bP) for unknown $a, b \in \mathbb{Z}_q^*$, compute abP.

3.2.2 Certificateless Signature Scheme

The certificateless signature scheme (`CLS`) from [29] initialises the system by running the set up algorithm `CLSsetup`(1^k) where 1^k is a security parameter. `CLSsetup`() chooses the groups $\langle \mathbb{G}_1, \mathbb{G}_2, e \rangle$, where $\mathbb{G}_1, \mathbb{G}_2$ are groups of prime order q and e is a bilinear pairing, $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$. It also selects 3 cryptographic hash functions $\mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3$, each of which maps from $\{0,1\}^*$ to $\mathbb{G}_1$. It chooses an integer $s \in_R \mathbb{Z}_q^*$ (where $s \in_R \mathbb{Z}_q^*$ denotes choosing an element s uniformly at random from the set $\mathbb{Z}_q^*$) as its master secret key. It sets $P_0 = s \cdot P \in \mathbb{G}_1$ as the master public key. `CLSsetup`() outputs $\langle s, \texttt{CLSparams} = \langle \mathbb{G}_1, \mathbb{G}_2, e, q, P, P_0, \mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3 \rangle \rangle$. The master secret key s is kept confidential while `CLSparams` is published as system parameters. From now

on we will assume the availability of $\texttt{CLSparams}$ in the description of the remaining protocols and algorithms.

During the enrolment of an entity V with an identifier $\texttt{ID}_V \in \{0,1\}^*$, the enrolment protocol $\texttt{CLSenrol}(\texttt{ID}_V)$ is performed by the TA and V in a secure environment. This protocol consists of two parts: $\texttt{CLSenrol}_{TA}(\texttt{ID}_V)$ and $\texttt{CLSenrol}_V(x_V)$ as below.

$\texttt{CLSenrol}(\texttt{ID}_V)$
- TA runs $\texttt{CLSenrol}_{TA}(\texttt{ID}_V)$
  - Computes $Q_V = \mathcal{H}_1(\texttt{ID}_V)$.
  - Computes $x_V = sQ_V$.
  - Outputs a partial private key x_V.
- TA sends x_V securely to V.
- V runs $\texttt{CLSenrol}_V(x_V)$
  - Selects a secret value $y_V \in_R \mathbb{Z}_q^*$.
  - Sets $\texttt{sk}_V = (x_V, y_V)$.
  - Sets $\texttt{pk}_V = y_V P$.
  - Outputs $(\texttt{pk}_V, \texttt{sk}_V)$.

To sign and verify a message M, the singing and verifying algorithms, denoted by $\texttt{CLSsign}()$and $\texttt{CLSverify}()$ respectively, are performed as follows.

$\texttt{CLSsign}(M, \texttt{sk}_V = (x_V, y_V), \texttt{ID}_V, \texttt{pk}_V)$
- Computes $U = u \cdot P$ for $u \in_R \mathbb{Z}_q^*$.
- Sets $v = x_V + u \cdot \mathcal{H}_2(M, \texttt{ID}_V, \texttt{pk}_V, U) + y_V \cdot \mathcal{H}_3(M, \texttt{ID}_V, \texttt{pk}_V)$.
- Outputs signature $\theta_V = (U, v)$.

$\texttt{CLSverify}(M, (U, v), \texttt{ID}_V, \texttt{pk}_V)$
- Computes $Q_V = \mathcal{H}_1(\texttt{ID}_V)$.
- Checks if the equality $e(v, P) = e(Q_V, P)e(\mathcal{H}_2(M, \texttt{ID}_V, \texttt{pk}_V, U), U)e(\mathcal{H}_3(M, \texttt{ID}_V, \texttt{pk}_V), \texttt{pk}_V)$ holds. If it does, outputs **valid**, otherwise outputs $\perp$.

We will use this scheme $\texttt{CLS} = (\texttt{CLSsetup}, \texttt{CLSenrol}, \texttt{CLSsign}, \texttt{CLSverify})$ in our announcement scheme in Section 5.

4 A Certificateless Anonymous Authenticated Announcement Scheme in VANETs

4.1 Entities in a VANET

A VANET is composed of vehicles equipped with onboard units (OBUs) and stationary units along the road, known as access points (APs). In addition, there is a trusted party - the reputation server ($\texttt{RS}$). We also assume there are malignant entities who aim to disrupt the system.

4.1.1 Vehicles and Onboard Units

We assume that the vehicles in our VANET system are equipped with an onboard unit (OBU) and a tamper resistant device (TRD). An OBU has a short range wireless communication device which can communicate directly with neighbouring vehicles' OBUs. The TRD (or black box) has a secure storage for private information such as secret keys and other cryptographic credentials. It also executes cryptographic operations such as generating signatures correctly and is assumed to have a secure time stamping service [27]. An adversary who is in control of a black box may generate fake messages of his own choice, but information in the secure storage cannot be retrieved, and protocols will still execute correctly.

4.1.2 Reputation Server

We rely on a fully trusted reputation server (RS). The RS is responsible for the distribution and management of identities and cryptographic credentials of the vehicles. They also maintain the reputation of vehicles. This includes collecting and aggregating feedback to produce reputation scores. The RS is also responsible for revoking vehicles. Periodic communication takes place between the RS and vehicles for reputation score retrieval and for feedback reporting. The RS does not need to be online otherwise. We also assume that the RS is equipped with a secure clock.

We note the assumption of such a fully trusted central authority is not unrealistic. Most countries have a central authority (such as the Driver and Vehicle Licensing Agency in the United Kingdom) responsible for the regulation of vehicles.

4.1.3 Access Points

An access point (AP) is a physical device located at fixed locations. Such locations include along the highways, roads, intersections, roundabouts or traffic lights. An AP is equipped with at least a network device for short-range wireless communication. Access points are connected with the reputation server, acting as a communication interface between vehicles and the RS. The purpose of access points is to allow vehicles to communicate with the RS in a convenient and frequent manner.

4.1.4 Adversary

One of the common assumptions in VANETs is the presence of a small fraction of adversaries [9, 14, 23] in the network. This includes external and internal adversaries, and rational and irrational adversaries. More details on adversaries can be found in [24, 27].

An external attacker is an entity who is without possession of any cryptographic credentials or direct physical access to the system. On the other hand, an internal attacker is a legitimate user of the VANET who is in possession of the credentials and black box. A dishonest user may cause more damage as he can control the black box to generate messages of his choice.

We assume adversaries are motivated by selfish (rational) or malicious (irrational) intent. A rational attacker has a plan for an attack to achieve his personal benefit where the benefits outweigh the cost. Meanwhile, an irrational attacker may attempt to degrade the reputation of others or impact the availability of the network without a personal gain. For example in [27], a terrorist may intentionally cause traffic accidents and delays to create chaos without considering the consequences. In our scheme, we consider rational adversaries.

There are two types of attacks we shall consider in our scheme: *reputation manipulation* and *message fraud attack*. Reputation manipulation attack means an adversary inflates or deflates the reputation score of a target vehicle, where this target vehicle may be the adversary himself. An adversary may also deceive receiving vehicle into accepting a fake message as valid. We call this a message fraud attack.

Communication channel. In our scheme, we assume that wireless channel between the reputation server RS, sending vehicles V_s and receiving vehicles V_r is public; that is, an entity can eavesdrop on the communication passing through the channel. The admission and initialisation of vehicles into the system is assumed to be conducted in a secure environment.

4.2 Scheme Overview

The reputation server RS initialises the system by installing a secure certificateless signature scheme CLS adopted from [29], along with a secure signature scheme SS, a secret key encryption scheme SKE and a public key encryption scheme PKE. The RS generates and publishes system parameters to vehicles in the network. The installation of the various schemes and associated keys onto vehicles is conducted during the admission of vehicles into the network. Operation of the scheme consists of *setup*, *reputation score retrieval*, *message broadcast*, *message verification*, *feedback reporting*, *reputation update* and *revocation*.

To communicate in the network, a vehicle V periodically retrieves a set of credentials from the RS via its nearest access point. To announce a safety related message, V anonymously sign the message with its reputation score attached and broadcast to neighboring vehicles. Each signing key key is valid over a short period. A receiving vehicle V' verifies the message based on the validity of the signature and the reputation of V. It may or may not provide a feedback about V. If it chooses to, V' provides a feedback score to rate its experience with V and signs a feedback to report to the RS. Upon receipt of the feedback, the RS validates the report based on V''s signature and timing of the feedback reported. It then computes the latest reputation of V and updates its database.

We note again that the reputation system we use here is essentially that of [20]. The novelty of our scheme lies in the use of privacy and authentication techniques based on certificateless signature. We will include all the description of the reputation system for the sake of completion.

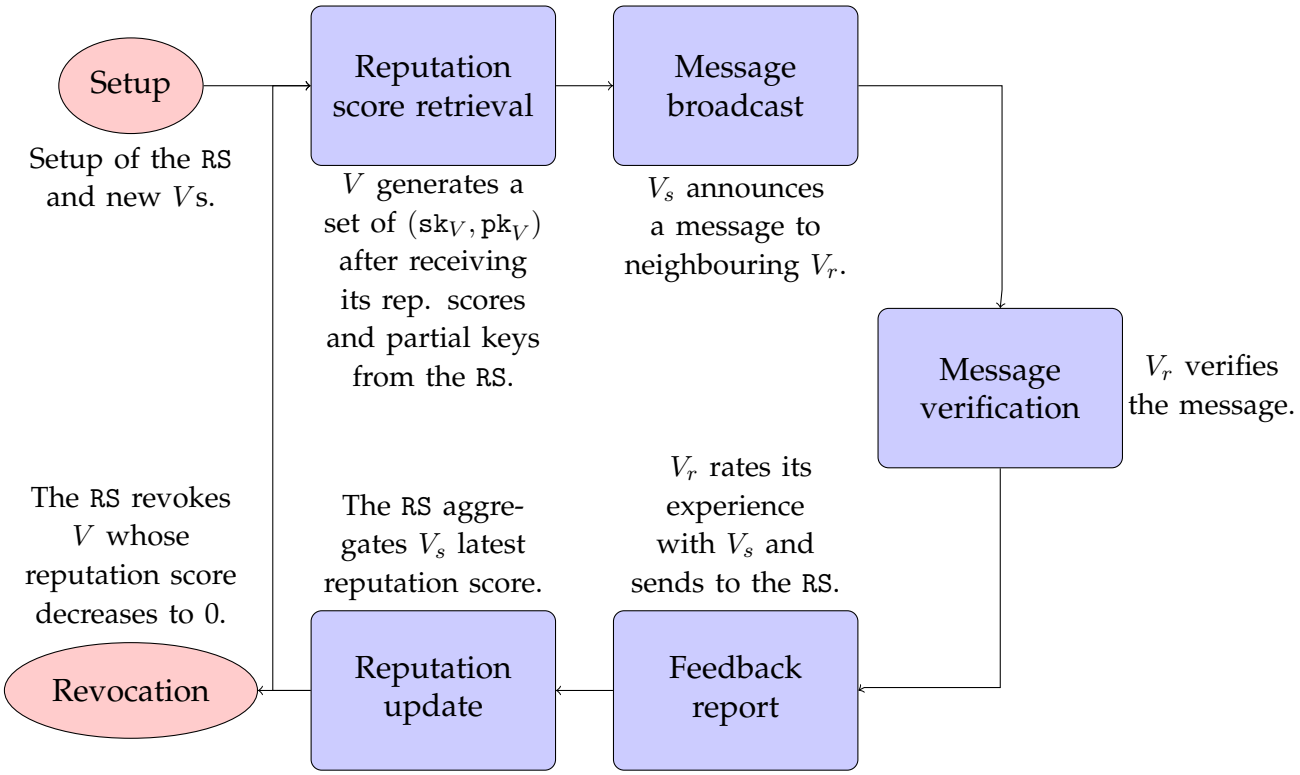

Figure 2: Flowchart of the scheme operation.

5 Scheme Operation

We describe our scheme by showing how reputation of a vehicle is formed, propagated, updated and utilised to determine the trustworthiness of vehicles. We note that the $\mathtt{CLSsetup}()$, $\mathtt{CLSenrol}()$, $\mathtt{CLSsign}()$ and $\mathtt{CLSverify}()$ used in this section has been presented in Section 3.2.2.

The operation of the scheme consists of the following phases: *setup*, *reputation score retrieval*, *message broadcast*, *message verification*, *feedback reporting*, *reputation update* and *revocation*.

5.1 The Setup

The setup creates system parameters and long term keys of the entities in the system. We describe the setup of the reputation server and vehicles as follow.

5.1.1 Setup of the Reputation Server

The RS installs:

- A secure certficateless signature scheme $\mathtt{CLS} = (\mathtt{CLSsetup}, \mathtt{CLSenrol}, \mathtt{CLSsign}, \mathtt{CLSverify})$ as described in Section 3.2.2.

- A secure signature scheme, defined by $\mathtt{SS} = (\mathtt{KGen}_{\mathtt{SS}}, \mathtt{SSsign}, \mathtt{SSverify})$ where $\mathtt{KGen}_{\mathtt{SS}}, \mathtt{SSsign}$ and $\mathtt{SSverify}$ denotes key generation, signing and verifying operation for a signature scheme respectively.
- A secure symmetric key encryption scheme, defined by $\mathtt{SKE} = (\mathtt{KGen}_{\mathtt{SKE}}, \mathtt{SKEnc}, \mathtt{SKDec})$ where $\mathtt{KGen}_{\mathtt{SKE}}, \mathtt{SKEnc}$ and $\mathtt{SKDec}$ denotes symmetric key generation, encryption and decryption respectively.
- A secure public key encryption scheme, defined by $\mathtt{PKE} = (\mathtt{KGen}_{\mathtt{PKE}}, \mathtt{PKEnc}, \mathtt{PKDec})$ where $\mathtt{KGen}_{\mathtt{PKE}}, \mathtt{PKEnc}$ and $\mathtt{PKDec}$ denotes public key generation, encryption and decryption respectively.

The $\mathtt{RS}$ then runs:

- The algorithm $\mathtt{CLSsetup}(1^k)$ as in section 3.2.2 to get $\langle s, \mathtt{CLSparams} = \langle \mathbb{G}_1, \mathbb{G}_2, e, q, P, P_0, \mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3 \rangle \rangle$.
- The $\mathtt{KGen}_{\mathtt{PKE}}$ to generate a key pair $(\mathtt{PK}_{\mathtt{RS}}, \mathtt{SK}_{\mathtt{RS}})$ used to encrypt and decrypt session keys (please refer to Section 5.2).
- Selects another secure hash function $\mathcal{H}_4 : \{0,1\}^n \rightarrow \mathbb{Z}_q^*$.
- Publishes $\mathtt{params} = \langle \mathtt{CLSparams}, \mathcal{H}_4, \mathtt{PK}_{\mathtt{RS}} \rangle$.
- Installs a time-discount function, denoted by TimeDiscount. This is a non-increasing function whose range is $[0,1]$. It takes input a non-negative value representing a time difference, and outputs a number between 0 and 1. For example, it can be defined as:

$$\mathsf{TimeDiscount}(t) = \begin{cases} 1 - t/\Psi_{TD} & \text{if } t < \Psi_{TD}; \\ 0 & \text{if } t \geq \Psi_{TD}, \end{cases}$$

  where Ψ_{TD} is a positive constant.
- Installs the reputation aggregation algorithm Aggr as described in section 5.6.1.

5.1.2 Admission of New Vehicles

- Each vehicle is assign with a unique identity $\mathtt{ID}_V \in \{0,1\}^*$.
- The $\mathtt{RS}$ installs $\mathtt{CLS}, \mathtt{SS}, \mathtt{SKE}$ and $\mathtt{PKE}$ schemes onto each vehicle V where V generates a pair of unique long term key pair:
  1. The key pair (pk_V, sk_V) generated using $\mathtt{KGen}_{\mathtt{SS}}$.
  2. The key pair $(\mathtt{PK}_V, \mathtt{SK}_V)$ generated using $\mathtt{KGen}_{\mathtt{PKE}}$.

The authentication process to validate pk_V and $\mathtt{PK}_V$ to the RS takes place during the V's registration before it is admitted into the system. The secret keys are stored within a V's black box while the public keys are kept within its onboard unit.

- The RS creates a database that will store the following data for every vehicle in the system: a vehicle's identity $\mathtt{ID}_V$, a unique long term public key pk_V and $\mathtt{PK}_V$, a set of pseudonyms $\mathtt{pseu}_V^i$, reputation scores $\mathtt{repscore}_V^i$, timestamps t_V^i on $(\mathtt{pseu}_V^i, \mathtt{repscore}_V^i)$, and all feedback reported for the vehicle (see Section 5.2 and 5.5).

- We require three configurable public parameters $\Psi_{\mathtt{RS}}$, Ψ_t and $\mathbb{T}$. The parameter $\Psi_{\mathtt{RS}}$ acts as a threshold and is used by a vehicle to determine whether or not another vehicle is reputable. It is a constant between 0 and 1. The parameter Ψ_t also acts as a threshold and used to determine whether or not a message tuple is sufficiently fresh for feedback reporting. The parameter $\mathbb{T}$ is a large time interval, over which a *sufficiently large* number of vehicles report feedback relating to a vehicle.

5.2 Reputation Score Retrieval

In this phase, a vehicle V retrieves from the RS a set of its credentials $\mathtt{Cre}_V = \{(\mathtt{pseu}_V^i, \mathtt{repscore}_V^i, t_V^i, x_V^i) : i = 1, \cdots, n\}$, where $\mathtt{pseu}_V^i$ is a random string, $\mathtt{repscore}_V^i$ is a reputation score, t_V^i is a timestamp for the validity period of $(\mathtt{pseu}_V^i, \mathtt{repscore}_V^i)$, and x_V^i is a partial secret key.

When it drives into the wireless communication range of an access point, the communication takes place as follow.

- V identifies itself to the RS and sets up a session key with RS.

  1. V signs a request message $reqkey$ together with its public key pk_V:

  $$\sigma \leftarrow \mathtt{SSsign}_{sk_V}(pk_V, reqkey).$$

  2. V generates a random session key $\mathtt{skey}_V$ using $\mathtt{KGen}_{\mathtt{SKE}}$ to encrypt the request and signature:

  $$\mathtt{req}_V \leftarrow \mathtt{SKEnc}_{\mathtt{skey}_V}(pk_V, reqkey, \sigma).$$

  3. V encrypts the session key using RS's public key $\mathtt{PK}_{\mathtt{RS}}$:

  $$\mathtt{key}_V \leftarrow \mathtt{PKEnc}_{\mathtt{PK}_{\mathtt{RS}}}(\mathtt{skey}_V).$$

  4. V sends $\{\mathtt{req}_V, \mathtt{key}_V\}$ to RS via the wireless channel.

- RS sends V its reputation scores and partial keys for the CLS.

1. RS decrypts $\texttt{key}_V$ using $\texttt{PKDec}_{\texttt{SK}_{\texttt{RS}}}$ to obtain $\texttt{skey}_V$. It then uses $\texttt{SKDec}_{\texttt{skey}_V}$ to decrypt $\texttt{req}_V$. Now RS has pk_V, $reqkey$, and the signature σ.
2. Upon verification of σ using SSverify, RS gets V_s's current reputation score repscore from the database and computes $\texttt{repscore}_V^i$ to be used at time beginning t_V^i by calculating $\texttt{repscore}_V^i = \texttt{repscore} \cdot \mathsf{TimeDiscount}(t_c - t_V^i)$, where t_c denotes the current time, until $\texttt{repscore}_V^i$ goes below the reputation threshold $\Psi_{\texttt{RS}}$. Suppose that $i = 1, \ldots, n$.
3. For $i = 1, \ldots n$, RS runs $\texttt{CLSenrol}_{\texttt{RS}}(\{\texttt{pseu}_V^i, \texttt{repscore}_V^i, t_V^i\})$ to obtain x_V^i:

$$x_V^i \leftarrow \texttt{CLSenrol}_{\texttt{RS}}(\{\texttt{pseu}_V^i, \texttt{repscore}_V^i, t_V^i\}).$$

4. RS forms $\texttt{Cre}_V = \{(\texttt{pseu}_V^i, \texttt{repscore}_V^i, t_V^i, x_V^i) : i = 1, \cdots, n\}$.
5. RS encrypts $\texttt{Cre}_V$ with the session key $\texttt{skey}_V$ using SKEnc:

$$\texttt{EncCre}_V \leftarrow \texttt{SKEnc}_{\texttt{skey}_V}(\texttt{Cre}_V).$$

6. RS sends $\texttt{EncCre}_V$ to V via the wireless channel.

- V generates public/private keys for the CLS.

1. V decrypts $\texttt{EncCre}_V$ using SKDec with the session key $\texttt{skey}_V$.
2. For $i = 1, \ldots n$, V runs $\texttt{CLSenrol}_V(x_V^i)$ to obtain a set of key pairs $(\texttt{pk}_V^i, \texttt{sk}_V^i)$:

$$(\texttt{pk}_V^i, \texttt{sk}_V^i) = (y_V^i P, (x_V^i, y_V^i)) \leftarrow \texttt{CLSenrol}_V(x_V^i).$$

3. The set of secret keys $\texttt{sk}_V^i$ is stored in the black box while the rest of the other parameters and public keys are stored in the vehicle's OBU.

The retrieval period varies, depending on how often a vehicle would like to obtain its latest reputation score or before it runs out of keys. A vehicle is likely to retrieve its credentials when its time-discounted reputation value $\texttt{repscore}_V^i \cdot \mathsf{TimeDiscount}(t_c - t_V^i)$, where t_c denotes the current time, is approaching or below the reputation threshold $\Psi_{\texttt{RS}}$. There is a tradeoff between the frequency a vehicle retrieves its keys from the RS and the efficiency of the scheme. Long intervals between retrieval period may be desirable as it may ease the management load of the RS who does not need to compute the keys for a vehicle frequently. However, it will lead to storage problems if a vehicle needs to preload a lot of keys for use over a long period of time.

Meanwhile, a shorter interval between retrieval periods solves the storage problem as a vehicle only needs to store fewer keys for a shorter time duration. It also provides a simpler means of revocation. Once it runs out of keys, a misbehaved vehicle would not be able to obtain the next set of keys from the RS. However, it implies frequent interactions between the RS and a vehicle.

5.3 Message Broadcast

An announcing vehicle V_s generates a road-related message $\texttt{msg}$ and broadcasts it to its neighbouring vehicles. This is described as follows.

1. V_s forms a $\texttt{MSG} = (h = \mathcal{H}_4(\texttt{msg}), t_b)$ where $h = \mathcal{H}_4(\texttt{msg})$ is a hash of the message and t_b is the time when the message was announced.

2. The $\texttt{CLSsign}()$ takes as input $\texttt{MSG}$, V_s's signing key $\texttt{sk}^i_{V_s}$, $\{\texttt{pseu}^i_{V_s}, \texttt{repscore}^i_{V_s}, t^i_{V_s}\}$ and V_s's public key $\texttt{pk}^i_{V_s}$. It returns a signature $\theta_{V_s} = (U, v)$.

$$\theta_{V_s} \leftarrow \texttt{CLSsign}(\texttt{MSG}, \texttt{sk}^i_{V_s}, \{\texttt{pseu}^i_{V_s}, \texttt{repscore}^i_{V_s}, t^i_{V_s}\}, \texttt{pk}^i_{V_s})$$

3. V_s forms a *message tuple* $M = (\texttt{msg}, t_b, \theta_{V_s}, \texttt{pk}^i_{V_s}, \texttt{pseu}^i_{V_s}, \texttt{repscore}^i_{V_s}, t^i_{V_s})$ and broadcasts M to its neighbouring vehicles.

5.4 Message Verification

Upon receiving the message tuple M, a receiving vehicle, say V_r, performs the following procedure:

1. It determines whether it is interested in the message $\texttt{msg}$. If it is, it computes $h = \mathcal{H}_4(\texttt{msg})$.

2. V_r inputs θ_{V_s} into its trusted hardware. The trusted hardware retrieves the current time t_r from its embedded clock, and then stores the tuple (θ_{V_s}, t_r) within the trusted hardware. The trusted hardware outputs t_r to V_r.

3. V_r determines whether the broadcasting vehicle is reputable, that is, $\texttt{repscore}^i_{V_s} \cdot \mathsf{TimeDiscount}(t_r - t^i_{V_s}) \geq \Psi_{\texttt{RS}}$.

4. V_r determines message freshness. A message is considered to be fresh if $t_r - t_b \leq \Psi_t$ where Ψ_t is very short time period after a sending vehicle announced a message.

5. V_r runs $\texttt{CLSverify}(\texttt{MSG}, \theta_{V_s}, \{\texttt{pseu}^i_{V_s}, \texttt{repscore}^i_{V_s}, t^i_{V_s}\}, \texttt{pk}^i_{V_s})$. If it returns **accept**, then the signature θ_{V_s} is considered valid. Otherwise V_r **rejects** the message.

The message $\texttt{msg}$ is considered reliable if all the above requirements are satisfied. The message tuple M is kept for future feedback reporting. If it does not fulfill the requirements, V_s is not considered as trustworthy and $\texttt{msg}$ is not considered as reliable and will not be taken into consideration. In the latter case, if Steps 4 and 5 are positive, then the message tuple M is still stored for future feedback reporting. Otherwise it is discarded.

5.5 Feedback Reporting

In this phase, when vehicle V_r has its own experience about the event that the message msg describes, it is able to judge the trustworthiness of the message. Then if V_r wants to report feedback to the reputation server, it performs the following procedure.

1. V_r generates a feedback rating $\texttt{feedrate} \in \{0, 1\}$ where $\texttt{feedrate} = 1$ if msg is reliable and $\texttt{feedrate} = 0$ if msg is not reliable.
2. V_r forms a $\texttt{feedback} = (\texttt{feedrate}, h, t_r, t_b, \theta_{V_s}, \texttt{pk}^i_{V_s}, \texttt{pseu}^i_{V_s})$.
3. V_r runs the $\texttt{CLSsign}()$ that takes as input a feedback, a feedback reporter's signing key $\texttt{sk}^j_{V_r}$, $\{\texttt{pseu}^j_{V_r}, \texttt{repscore}^j_{V_r}, t^j_{V_r}\}$ and V_r's public key $\texttt{pk}^j_{V_r}$. It returns a signature $\theta_{V_r} = (U', v')$.

$$\theta_{V_r} \leftarrow \texttt{CLSsign}(\texttt{feedback}, \texttt{sk}^j_{V_r}, \{\texttt{pseu}^j_{V_r}, \texttt{repscore}^j_{V_r}, t^j_{V_r}\}, \texttt{pk}^j_{V_r})$$

4. V_r casts a feedback report = $(\texttt{feedback}, \theta_{V_r}, \texttt{pk}^j_{V_r})$.

When V_r drives into the wireless communication range of a AP, it sends the feedback report to the RS via the AP.

5.6 Reputation Update

In this phase, the reputation server updates the reputation score $\texttt{repscore}_{V_s}$ of vehicle V_s. Upon receipt of a feedback report, the RS retrieves $(\texttt{repscore}^i_{V_s}, t^i_{V_s}, \texttt{repscore}^j_{V_r}, t^j_{V_r})$ from its database based on $(\texttt{pseu}^i_{V_s}, \texttt{pseu}^j_{V_r})$ for V_s and V_r respectively before performs the algorithm below.

1. It determines whether $t_r - t_b \leq \Psi_t$ where Ψ_t is small. This is performed to ensure that a receiving vehicle cannot forward this message to other colluding vehicles and together launch an attack to manipulate the reputation of the broadcasting vehicle.
2. It runs $\texttt{CLSverify}(\texttt{feedback}, \theta_{V_r}, \{\texttt{pseu}^j_{V_r}, \texttt{repscore}^j_{V_r}, t^j_{V_r}\}, \texttt{pk}^j_{V_r})$. If it returns **accept**, then the signature θ_{V_r} is considered valid. Otherwise RS **rejects** the feedback.
3. It runs $\texttt{CLSverify}(\texttt{MSG}, \theta_{V_s}, \{\texttt{pseu}^i_{V_s}, \texttt{repscore}^i_{V_s}, t^i_{V_s}\}, \texttt{pk}^i_{V_s})$. If it returns **accept**, then the signature θ_{V_s} is considered valid. Otherwise RS **rejects** the feedback.
4. If the checks pass then the reputation server considers the feedback report as valid and stores it in the database.

The RS applies the reputation aggregation algorithm Aggr (Section 5.6.1) on all stored feedback relating to V_s in order to compute the latest reputation score for vehicle V_s. It then replaces the previous reputation score in the database with its latest reputation score.

5.6.1 The Reputation Aggregation Algorithm

In this section, we describe the reputation aggregation algorithm Aggr from [20]. The Aggr computes the latest reputation score for a vehicle V based on all stored feedback, as follows:

1. The Aggr selects all feedback reported for V whose corresponding message tuple was broadcast from $\mathbb{T}$ time ago up to now. Any feedback whose corresponding message was broadcast earlier than $\mathbb{T}$ time ago is ignored, and deleted if necessary for data storage efficiency. We denote t_a as the time when this aggregation is running.

2. Multiple feedback reported by a vehicle V_z for V is aggregated into one intermediate value $\hat{r}_{V_z}$. Let $\mathcal{F}_{V_z}$ denote the set of feedback reported by V_z for V and whose corresponding message was broadcast from $\mathbb{T}$ time ago up to now. Each entry in $\mathcal{F}_{V_z}$ has feedback rating $\texttt{feedrate}_b$ corresponds to the message broadcasted at time t_b. The value $\hat{r}_{V_z}$ is aggregated using weighted average as follows:

$$\hat{r}_{V_z} = \frac{\sum\limits_{\texttt{feedback} \in \mathcal{F}_{V_z}} \texttt{feedrate}_b \cdot \big(\mathbb{T} - (t_a - t_b)\big)}{\sum\limits_{\texttt{feedback} \in \mathcal{F}_{V_z}} \big(\mathbb{T} - (t_a - t_b)\big)}. \tag{1}$$

   This gives more recent feedback a greater weight than less recent feedback. Let $\mathcal{V}$ denote the set of vehicles that each has reported at least one feedback for V in the past $\mathbb{T}$ time. The value $\hat{r}_{V_z}$ is computed for each vehicle $V_z \in \mathcal{V}$.

3. Let $\mathcal{V}^-$ denote the set of vehicles reporting at least one negative feedback for V in the past $\mathbb{T}$ time. The latest reputation score $\texttt{repscore}_V$ is computed as follows:

$$\texttt{repscore}_V = \begin{cases} \frac{\sum\limits_{V_z \in \mathcal{V}} \hat{r}_{V_z}}{|\mathcal{V}|} & \text{if } |\mathcal{V}^-| < \Psi_{nf}; \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

   where Ψ_{nf} is a configurable public parameter. The intuition of this equation is that $\texttt{repscore}_V$ is computed as the average of $\hat{r}_{V_z}$ if not too many vehicles reported negative feedback for V in the past $\mathbb{T}$ time; otherise $\texttt{repscore}_V$ decreases to 0, indicating that V has conducted message fraud attack.

5.7 Revocation

In our paper, a vehicle retrieves a set of credentials $\texttt{Cre}_V = (\texttt{pseu}_V^i, \texttt{repscore}_V^i, t_V^i, x_V^i)$ from the RS. The design of our scheme allows a shorter interval of credentials retrieval. Frequent credentials retrieval allows a vehicle to obtain its latest reputation score as the reputation score of a vehicle evolves, based on the reliability of messages that the vehicle announces. A vehicle whose reputation score decreases to 0 will be revoked from the system. The RS will stop issuing pseudonyms, reputation scores, timestamps and the partial keys. A misbehaved vehicle would then not be able to

compute its secret and its public key. Therefore, it would not be able to participate in future communication in the network.

6 Analysis

In this section, we analyse the security of our scheme, and evaluate its performance. We compare our scheme with schemes that adopt a pseudonyms method [7, 27], which is of the most interest to us in this work.

6.1 Security Analysis

We compare our scheme with Hybrid [7] and pseudonymous public key (PPK) [27] based on three main security requirements of reliability, privacy and accountability. We consider the eight security requirements as discussed in Section 1 and summarised our finding in Table 1 below.

6.1.1 Reliability

The requirement of sender authenticity and message integrity are satisfied in all three schemes, as long as the digital signature techniques used are secure. Similarly, in feedback reporting, reporter authenticity and report integrity are achieved if the digital signature schemes are secure. We also require the signature, public and symmetric key encryption schemes adopted during reputation score retrieval phase in section 5.2 to be secure. This is to ensure that the RS provides the correct credentials after verifying the legitimacy of a requesting vehicle V.

However, the property of message truthfulness is not provided in Hybrid as distinguishability of message origin is difficult to achieve without an online trusted authority. Hence, the threshold technique cannot be adopted. It is also not satisfied in the scheme proposed in PPK as, for the same message, a signing vehicle can disguise as multiple vehicles. In our scheme a message is regarded as truthful if the message originator has a "good" reputation. Hence in order to lie successfully, an adversary could do one of two things: it can manipulate the reputation score of the sending vehicle, or it can manipulate the message content of an announcement.

In the latter case, neither an external nor an internal adversary will be able to convince receiving vehicles that a modified message is valid if the certificateless signature scheme is secure. On the other hand, an internal adversary with a high reputation score can deceive a receiving vehicle into accepting a false message easily: it simply broadcasts the false message. However, if it does this persistently over a long period, then the negative feedback will result in a decrease in its reputation score. Eventually its reputation score will decrease to 0 and it will be revoked from the system.

To manipulate the reputation score of a target vehicle V, firstly an adversary could impersonate V and broadcast false messages in order for V to receive negative feedback and thereby decrease its reputation score. This cannot be done if

the certificateless signature scheme is secure. Secondly an adversary could instead replace V's reputation score with a lower one in a broadcast message. Again this could not be done if the CLS is secure. Lastly an adversary could provide negative feedback for announcements made by V. Clearly an external adversary cannot perform this attack given that the CLS is secure. An internal adversary acting on its own can only report a false feedback per announcement, and this will have only a small impact on V's reputation score. Even if the internal adversary colludes with a group of other internal adversaries, the effect will remain small if the proportion of dishonest vehicles is small, as is the assumption. In addition, the provision of timestamps limits the vehicles who can provide feedback to those in proximity when the message is announced.

Hence we see that our scheme provides reliability and also provides system robustness in the presence of a small fraction of adversaries.

6.1.2 Privacy

In PPK and our scheme, messages are linkable only over the short validity period of a pseudonym. In Hybrid, a vehicle uses its group signing key to certify a self-generated pseudonym. The rate at which pseudonyms are updated depends on the various factors. Hence, similar to our scheme and PPK, messages signed using the same pseudonym are linkable over its short lifetime (marked $\checkmark^*$ in Table 1). This is a slight compromise of privacy in favour of reducing storage and communication costs. The length of the validity period can be adjusted according to the level of privacy required. In our scheme, this applies to both announcements and feedback reporting. The request activity for credentials made by V is also unlinkable using a secure symmetric key encryption scheme where a random session key is generated to encrypt each request. The session key is then encrypted using a secure public key encryption scheme.

Anonymity of broadcast messages is achieved by both Hybrid and PPK. In our scheme, communications for the retrieval of reputation scores and pseudonyms are protected by signatures and encryptions. As long as these schemes are secure, a vehicle and its credentials will be anonymous. Hence our scheme preserves anonymity in reputation score retrieval, message announcements and feedback reporting.

Note that the above refers to privacy against any eavesdropper apart from the trusted authority. None of the schemes provide any privacy against the TA (RS in our case) in the sense that if a set of broadcast messages were presented to the TA, it would be able to link the messages to the senders. There is also no privacy against the RS in feedback reporting. Since the RS is trusted to correctly manage the reputation system, this is not a great compromise. Note though that the RS does not know the activities of the vehicles since a feedback report only contains a hash of the message content.

6.1.3 Accountability

The property of traceability is satisfied in all schemes. The group signature in Hybrid allows a TA to open signature of malicious vehicles, where the identity of misbehaved vehicles is revealed by law enforcement authorities for liability purposes. In PPK as well as our scheme, the TA is able to search in its database and trace the identity of misbehaved vehicles.

Neither Hybrid nor PPK provides non-repudiation. The group signature technique used in Hybrid permit the issuer to create the private keys of group members. In the scheme in PPK, the TA generates the secret key for all vehicles. Therefore, these schemes does not achieve non-repudiation as the signer is not the sole holder of the signing key. In our scheme, the RS does not have access to entities' private key as it only generates an entity with a partial private key. This satisfies the requirement of non-repudiation.

Revocation in Hybrid is achieved by having a vehicle's revocation token added into the revocation list. Upon verifying a message, signature generated from a revoked vehicle will not be accepted. In PPK, the TA exhaustively search in its huge database where it stores all the anonymous certificates issued to vehicles to find the real identity of a misbehaved vehicle. In our scheme, the RS maintains a map from a vehicle's long term identity to its set of pseudonyms, where the RS can perform an inverse mapping and identify the vehicle whose reputation score decreases to 0. The authority will then stop issuing pseudonyms, reputation scores and partial keys to misbehaved vehicles and hence, these vehicles would not be able to generate its secret value, its public key and compute a full secret key to announce a message.

Security Analysis				
Security goals	Security components	Hybrid [7]	PPK [27]	Our scheme
Reliability	Sender's Authenticity	✓	✓	✓
	Message Integrity	✓	✓	✓
	Message truthfulness	×	×	✓
Privacy	Anonymity	✓	✓	✓
	Unlinkability	✓*	✓*	✓*
Accountability	Non-repudiation	×	×	✓
	Revocation	✓	✓	✓
	Traceability	✓	✓	✓

Table 1: Comparison of security analysis

6.2 Performance Analysis

We compare the performance of our scheme with Hybrid [7] and PPK [27]. The group signature (GS) in Hybrid used to certify self-generated pseudonym is adopted from [5] and we choose to employ elliptic curve cryptosystem, such as ECDSA scheme [6, 16, 17] as the basic signature algorithm to sign messages, which will also

be used in PPK. We set security level $l = 80$ bits for message signatures and $l = 128$ bits for certificates in Hybrid and PPK. This is a similar adoption of values and signature algorithm as in Hybrid, for the purpose of comparison. We summarise our findings in Tables 2 and 3. The question of how usable the system is measured in terms of message drop rate and the effect of the availability of the RS and AP is demonstrated in simulation results in [20]. We will summarise the findings here (Section 6.3)and refer the reader to [20] for details.

Computational cost. We evaluate the computational cost of signature generation and verification in the broadcast of messages. As observed in [8, 9], the two most expensive operations are multiplications in $\mathbb{G}_1$ and pairing evaluation, which we shall consider here. We compare the cost between our scheme with Hybrid [7] and PPK [27] for $t = 1$, as our scheme requires only one message provided that the message generator has sufficiently high reputation.

The signing operation in PPK requires 2 scalar multiplications and the verification requires 4 scalar multiplications. Meanwhile, the Hybrid scheme requires a vehicle to generate 2 signatures; a group signature adopted from [5] to certify a self-generated pseudonym pk and a signature similar to PPK on the announced message. The group signature requires 8 scalar multiplications and 1 pairing operation for the signing phase, while the verification phase requires 5 scalar multiplications and 3 pairing operations. The signature generation and verification on a message is then similar to PPK described earlier.

The signing procedure of our scheme requires 3 scalar multiplications and the verification requires 4 pairing operations. These findings are summarised in Table 2. We see that the computational cost for our scheme is comparable to PPK and more efficient compared to Hybrid. In addition, as noted before, in our scheme, a receiving vehicle may make a decision on whether to rely on a broadcast message immediately, while in PPK and Hybrid, a receiving vehicle typically requires a few messages before reliability can be confirmed.

Our scheme has additional operations where V_r may choose to provide a feedback to rate its experience with the message generator. In this case, the computational cost is that of 1 signature. The verification of feedback requires 2 signatures verifications. This is performed by the RS and can be done offline.

Signature length. The signature in PPK generated using elliptic curve digital signature algorithm (ECDSA)[6, 16, 17] comprises of 2 elements of $\mathbb{G}_1$. A group signature in Hybrid comprises of 2 elements of $\mathbb{G}_1$ and 5 elements of $\mathbb{Z}_q$. Meanwhile, the length of signature in our scheme composed of 2 elements of $\mathbb{G}_1$, which is similar to PPK. To provide a security level 2^{80}, we can set q to be 190-bit long and the element in $\mathbb{G}_1$ is 191-bit long by choosing an appropriate curve such as NIST curve [6]. Thus, the length of signature generated on a message is 48 bytes in our scheme and PPK. In Hybrid, the message signature is 48 bytes and the length of signature on the certified pseudonym is of 224 bytes (for a security level 2^{128}, we have $|q| = 255$ bits and $|\mathbb{G}_1|$=256 bits), which sums up to 272 bytes generated by a vehicle. This again shows that our scheme provides message signatures with length comparable to those of

existing schemes. This result is summarised in Table 3.

Communication cost. A message M in Hybrid and PPK consists of: $(\theta_{\mathtt{sk}}(\mathtt{msg}), \mathtt{msg}, \mathtt{cert}_{TA}(\mathtt{pk}), \mathtt{pk}, t)$, which denotes signature generated on an announced message, the message, its certificate which essentially is the signature of the TA on a vehicle's public key, the vehicle's public key and a timestamp respectively. To provide a security level of 2^{80}, we can take q to be 190-bit long and the element in $\mathbb{G}_1$ to be 191-bit long. According to [3] the size of safety messages is 100 bytes and we choose 8 bytes for timestamp, using the unix 64-bit timestamp. Based on the implementation in [6, 18], the size of the public key is 25 bytes, message signature is 48 bytes and TA certificate is 64 bytes. Hence the length of the message in PPK is $L = 48 + 100 + 64 + 25 + 8 = 245$ bytes. In Hybrid, the message size is $L = 48 + 100 + 224 + 25 + 8 = 405$ bytes. We note that the implementation of GS in [5] adopted by Hybrid is not available to us, hence we use the similar values in Hybrid which were calculated using the number of 32-bit word multiplications required for GS signing and verifying, extracted from [6, 18].

In our scheme, a message is composed of: $(\theta_{V_s}(\mathtt{msg}), \mathtt{msg}, \mathtt{pk}^i_{V_s}, \mathtt{pseu}^i_{V_s}, \mathtt{repscore}^i_{V_s}, t^i_{\mathtt{rs}_{V_s}}, t_b)$, where $\theta_{V_s}(\mathtt{msg}), \mathtt{msg}, \mathtt{pk}^i_{V_s}, \mathtt{pseu}^i_{V_s}, \mathtt{repscore}^i_{V_s}, t^i_{\mathtt{rs}_{V_s}}$ and t_b denotes a signature generated on an announced message, the message, a vehicle's public key, its pseudonym, its reputation score and a timestamp on the reputation score and message broadcasted respectively. The size of the public key and pseudonym is of an element $\mathbb{G}_1$ each, and reputation score of size 1 byte. Therefore the size of the message size is $L = 48 + 100 + 24 + 24 + 1 + 8 + 8 = 213$ bytes. We observed that our scheme yield the shortest message size compared to these two schemes. This result is summarised in Table 3.

Our scheme is also composed of a feedback reporting phase where a receiving vehicle V_r may choose to rate its experience with the message generator V_s. A feedback report is composed of: $((\mathtt{feedrate}, h, t_r, t_b, \theta_{V_s}, \mathtt{pk}^i_{V_s}, \mathtt{pseu}^i_{V_s}), \theta_{V_r}, \mathtt{pk}^j_{V_r})$. The size of the $\mathtt{feedrate}$ is of 1 byte, the timestamp t_r and t_b is of 8 bytes each, and hash of the announced event h is of an element q, which is 24 bytes. The pseudonym $\mathtt{pseu}^i_{V_s}$ and both public keys $\mathtt{pk}^i_{V_s}$ and $\mathtt{pk}^j_{V_r}$ is an element of $\mathbb{G}_1$ each respectively. The signature generated by V_r on the feedback is of two elements of $\mathbb{G}_1$, similar to θ_{V_s}. Hence the length of the feedback report is $F = 1 + 24 + 8 + 8 + 48 + 24 + 24 + 48 + 24 = 137$ bytes.

Storage cost. We compare the storage cost of our scheme with PPK during credential retrieval phase given the similar approach of preloading credentials onto a vehicle. For each credential retrieval period, PPK preloads a large set of key pairs and their corresponding certificate onto each vehicle, for its usage over a long period of time (i.e. a year). The next retrieval may occur during periodical vehicle maintenance visits, for instance. The public and private key is 25 bytes and 24 bytes respectively, using ECDSA-192 and a TA certificate is 64 bytes using ECDSA-256. This sums up to storage space of $25 + 24 + 64 = 113$ bytes per key. In [26], they assumed an average a driver uses his car is 2 hours per day, where the lifetime of each key is one minute.

Then the number of required keys per year is approximately 43800, which amounts to storage space of 4.95 Mbytes on each vehicle.

In our scheme, a vehicle retrieves from the RS a set of credentials $\texttt{Cre}_V = (\texttt{pseu}_V^i, \texttt{repscore}_V^i, t_V^i, x_V^i)$. The partial key x_V^i and pseudonym $\texttt{pseu}_V^i$ is an element of $\mathbb{G}_1$ each, hence x_V^i and $\texttt{pseu}_V^i$ is of 24 bytes each for $|\mathbb{G}_1| = 191$ bits. The sum of storage space is of $8+1+24+24 = 57$ bytes per key, which is more efficient as it is about half of the PPK key size. The retrieval period in our scheme may be shorter and flexible, depending on whether a vehicle would like to obtain its latest reputation score or when it runs out of credentials.

Scheme	Msg signature Sign	Msg signature Verify	Group signature Sign	Group signature Verify
PPK	$2 \cdot \mathbb{G}_1$	$4 \cdot \mathbb{G}_1$	N/A	N/A
Hybrid	$2 \cdot \mathbb{G}_1$	$4 \cdot \mathbb{G}_1$	$8 \cdot \mathbb{G}_1 + 1 \cdot P$	$5 \cdot \mathbb{G}_1 + 3 \cdot P$
Ours	$3 \cdot \mathbb{G}_1$	$4 \cdot P$	N/A	N/A

Here $n \cdot \mathbb{G}_1$ denotes n scalar multiplications and $n \cdot P$ denotes n pairing operations.

Table 2: Comparison of computational cost

Scheme	Signature length (bytes)	Communication cost	Storage cost
PPK	$2\|\mathbb{G}_1\|$ (48)	245 bytes	113 bytes/key
Hybrid	(Group sig) $2\|\mathbb{G}_1\| + 5\|q\|$ (224) (Msg sig) $2 \cdot \mathbb{G}_1$ (48)	405 bytes	- -
Ours	$2 \cdot \mathbb{G}_1$ (48)	213 bytes	57 bytes/key

Table 3: Comparison of communication and storage cost ($l = 80$)

6.3 Effect of temporary unavailability of RS and AP

The effect of the temporary unavailability of RS and AP can be measured in terms of message drop rates: the average rate that reliable messages are rejected by a receiving vehicle due to low reputation scores of the broadcasting vehicles. Since the reputation system we use here is the same as that of [20] the simulation results on message drop rates will also apply to our scheme. We summarise the results here and refer the reader to [20] for details.

The simulations are carried out with conditions in line with other studies in the literature. If RS and AP are all functioning continuously, the message drop rate depends (obviously) on the density of the vehicles and the density of APs. It appears that the message drop rates decreases drammatically when the density of AP is increased from very low. Subsequent increases have much smaller effect. For example, if there are 2 APs per km^2, the message drop rate is 0.1 if the density of vehicles is 500

per $10km^2$, while the message drop rate is less than 0.05 if there are 4 APs per km^2. This also confirms that even with initial reputation score of 0, a new vehicle will be able to establish its own reputation fairly reasonably. This is because a receiving vehicle may still provide a feedback even if it considered a message unreliable due to low reputation.

Temporary unavailability of RS. With 2 to 5 APs per km^2, the message drop rates increases proportionally as the length of unavailability of RS, until a certain point where the message drop rate is 1. This point is dependent upon the time discount parameter Ψ_{TD} and the reputation threshold $\Psi_{\mathtt{RS}}$. If Ψ_{TD} and $\Psi_{\mathtt{RS}}$ are set conservatively to 1 hour and 0.8 respectively (and the experiment time is only 30 minutes) then the message drop rate reaches 1 in 12 minutes. It is expected that in a real-world implementation with a much longer Ψ_{TD} the time to reach message drop rate 1 would be much longer.

Temporary unavailability of APs. The simulation result shows that for 5 APs per km^2 with the density of 500 vehicles per $10km^2$, even the unavailability of up to 50% of APs for 25 minutes contribute only slightly to the increase in message drop rate. Again, this is not unexpected, since a vehicle can always retrieve its reputation score and report feedback when it comes across another functioning AP.

7 Conclusion and Future Work

We have presented a novel privacy-preserving authentication protocol for VANETs based on certificateless signature and reputation systems. To our knowledge, this is the first certificateless announcement scheme for VANETs that has been proposed in the literature. We have shown that our scheme is efficient and robust, and achieves the desirable properties of a reliable, anonymous and accountable announcement scheme without introducing the problems of certificate management and overhead.

Some questions remain open for future research:

- In our scheme, an announced event is only utilised by its neighbouring vehicles. It might be of interest to extend the current scheme to where a message can be utilised by vehicles in a greater area. How this may be done without compromising the security against reputation manipulation attacks is the subject of future research.

- In this paper, we present a simple feedback aggregation algorithm based on binary feedback ratings. It might be of interest to investigate alternative approaches which allow continuous feedback ratings and thus provide richer results.

References

[1] Sattam S. Al-Riyami and Kenneth G. Paterson. Certificateless Public Key Cryptography. In *ASIACRYPT*, volume 2894 of *Lecture Notes in Computer Science*, pages 452–473. Springer, 2003.

[2] Tim Finin Anand Patwardhan, Anupam Joshi and Yelena Yesha. A Data Intensive Reputation Management Scheme for Vehicular Ad Hoc Networks. In *Proceedings of the 3rd Annual International Conference on Mobile and Ubiquitous Systems*, pages 1–8, 2006.

[3] ASTM E2213-03. Standard Specification for Telecommunications and Information Exchange Between Roadside and Vehicle Systems - 5 GHz Band Dedicated Short Range Communications (DSRC) Medium Access Control (MAC) and Physical Layer (PHY) Specification, 2003.

[4] Dan Boneh and Matthew K. Franklin. Identity-Based Encryption from the Weil Pairing. In *CRYPTO*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer, 2001.

[5] Dan Boneh and Hovav Shacham. Group Signatures with Verifier-Local Revocation. In *ACM Conference on Computer and Communications Security*, pages 168–177, 2004.

[6] Michael Brown, Darrel Hankerson, Julio López, and Alfred Menezes. Software Implementation of the NIST Elliptic Curves Over Prime Fields. In *CT-RSA*, Lecture Notes in Computer Science, pages 250–265. Springer, 2001.

[7] Giorgio Calandriello, Panos Papadimitratos, Jean-Pierre Hubaux, and Antonio Lioy. Efficient and Robust Pseudonymous Authentication in VANET. In *Proceedings of the 4th ACM International Workshop on Vehicular Ad Hoc Networks*, pages 19–28. ACM, 2007.

[8] Liqun Chen, Paul Morrissey, and Nigel P. Smart. DAA: Fixing the Pairing-Based Protocols. *IACR Cryptology ePrint Archive*, page 198, 2009. Available at http://eprint.iacr.org/2009/198.

[9] Liqun Chen, Siaw-Lynn Ng, and Guilin Wang. Threshold Anonymous Announcement in VANETs. *IEEE Journal on Selected Areas in Communications*, 29:605–615, 2011.

[10] Vanesa Daza, Josep Domingo-Ferrer, Francesc Sebé, and Alexandre Viejo. Trustworthy Privacy-Preserving Car Generated Announcements in Vehicular Ad Hoc Networks. *IEEE Transaction on Vehicular Technology*, 58(4):1876 – 1886, 2009.

[11] Josep Domingo-Ferrer, Qianhong Wu, and Úrsula González-Nicolás. *Balanced Trustworthiness, Safety and Privacy in Vehicle-to-Vehicle Communications*,

2010. To be published in IEEE Transactions on Vehicular Technology. DOI: 10.1109/TVT.2009.2034669.

[12] Florian Dötzer, Lars Fischer, and Przemyslaw Magiera. VARS: A Vehicle Ad Hoc Network Reputation System. In *Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks*, volume 1, pages 454–456, 2005.

[13] John R. Douceur. The Sybil Attack. In Peter Druschel and M. Frans Kaashoek and Antony I. T. Rowstron, editor, *IPTPS*, volume 2429 of *Lecture Notes in Computer Science*, pages 251–260. Springer, 2002.

[14] Philippe Golle, Daniel H. Greene, and Jessica Staddon. Detecting and Correcting Malicious Data in VANETs. In *Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks*, pages 29–37. ACM, 2004.

[15] Peter Gutmann. PKI: It's Not Dead, Just Resting. *IEEE Computer*, 35(8):41–49, 2002.

[16] IEEE 1363a. IEEE Standard Specifications for Public Key Cryptography - Amendment 1: Additional Technique, 2004.

[17] Don Johnson, Alfred Menezes, and Scott A. Vanstone. The Elliptic Curve Digital Signature Algorithm (ECDSA). *International Journal of Information Security*, 1(1):36–63, 2001.

[18] Neal Koblitz and Alfred Menezes. Pairing-Based Cryptography at High Security Levels. In *IMA Int. Conf.*, pages 13–36, 2005.

[19] Gina Kounga, Thomas Walter, and Sven Lachmund. Proving Reliability of Anonymous Information in VANETs. *IEEE Transactions on Vehicular Technology*, 58(6):2977–2989, 2009.

[20] Qin Li, Amizah Malip, Keith M. Martin, Siaw-Lynn Ng, and Jie Zhang. Reputation-based Announcement Scheme for VANETs. *IEEE Transactions on Vehicular Technology*, 61(9):4095–4108, 2012.

[21] Xiaodong Lin, Xiaoting Sun, and Pin-Han Ho. GSIS: Secure Vehicular Communications with Privacy Preserving. In *IEEE Transactions on Vechicular Technology*, volume 56, pages 3442–3456, 2007.

[22] Umar Farooq Minhas, Jie Zhang, Thomas Tran, and Robin Cohen. Towards Expanded Trust Management for Agents in Vehicular Ad Hoc Networks. In *International Journal of Computational Intelligence Theory and Practice (IJCITP)*, pages 3–15, 2010.

[23] P. Papadimitratos, L. Buttyan, T. Holczer, E. Schoch, J. Freudiger, M. Raya, Z. Ma, F. Kargl, A. Kung, and J.-P. Hubaux. Secure Vehicular Communication Systems: Design and Architecture. *IEEE Communcations magazine*, pages 100–109, 2008.

[24] Bryan Parno and Adrian Perrig. Challenges in Securing Vehicular Networks. In *Proceedings of Workshop on Hot Topics in Networks (HotNets-IV)*, 2005.

[25] Maxim Raya, Adel Aziz, and Jean-Pierre Hubaux. Efficient Secure Aggregation in VANETs. In *Proceedings of the 3rd International Workshop on Vehicular Ad Hoc Networks*, pages 67–75. ACM, 2006.

[26] Maxim Raya and Jean-Pierre Hubaux. The Security of Vehicular Ad Hoc Networks. In *Proceeding of SASN*, pages 11–21. ACM, 2005.

[27] Maxim Raya and Jean-Pierre Hubaux. Securing Vehicular Ad Hoc Networks. *Journal of Computer Security*, 15(1):39–68, 2007.

[28] Adi Shamir. Identity-Based Cryptosystems and Signature Schemes. In *CRYPTO*, pages 47–53, 1984.

[29] Zhenfeng Zhang, Duncan S. Wong, Jing Xu, and Dengguo Feng. Certificateless Public-Key Signature: Security Model and Efficient Construction. In *ACNS*, volume 3989 of *Lecture Notes in Computer Science*, pages 293–308, 2006.