Defense against Sybil Attack in the Initial Deployment Stage of Vehicular Ad hoc Network based on Roadside Unit Support

Soyoung Park[1], Baber Aslam[2], Damla Turgut[2], *Member, IEEE,* and Cliff C. Zou[2], Senior *Member, IEEE*

***Abstract*—In this paper, we propose two certificate mechanisms for preventing the Sybil attack in a vehicular ad hoc network (VANET): the timestamp series approach and the temporary certificate approach. We focus on an early-stage VANET when the number of smart vehicles is only a small fraction of the vehicles on the road and the only infrastructure components available are the roadside units (RSUs). Unlike previously proposed schemes, which require a dedicated vehicular public key infrastructure to certify individual vehicles, in our approach the RSUs are the only components issuing certificates. The vehicles can obtain certificates by simply driving by RSUs, without the need to pre-register at a certificate authority (CA).**

The timestamp series approach exploits the fact that due to the variance of the movement patterns of the vehicles, it is extremely rare that the two vehicles pass by a series of RSUs at exactly the same time points. The vehicles obtain a series of certificates signed by the RSUs, which certify their passing by at the RSU at a certain timepoint. By exploiting the spatial and temporal correlation between vehicles and RSUs, the Sybil attack can be detected by checking the similarity of timestamp series.

In the temporary certificate-based approach, an RSU issues temporary certificates valid only in a particular area for a limited time. To guarantee that each vehicle is assigned only a single certificate, at the issuance of the first certificate it is required that the RSU physically authenticate the vehicle. When driving by the subsequent RSUs, however, the certificate can be updated in chained manner. By guaranteeing that each vehicle is issued a single certificate in a single area, the Sybil attack is prevented.

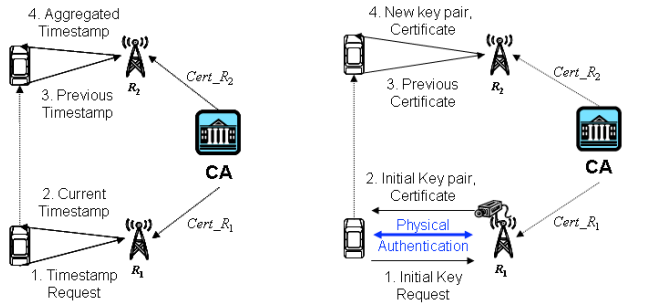
We provide mathematical analysis and simulation for the timestamp series approach. The simulation shows that it works with a small false-positive rate in simple roadway architecture.

***Index Terms*—Vehicular ad hoc network, Sybil attack, roadside Unit, temporary certificates, security**

I. Introduction

Vehicular Ad hoc NETworks (VANETs) allow vehicles on the road to exchange traffic information with each other. This can contribute to a more fluent traffic, as well as the improved road safety by advance warning about accidents and specific threats to safety. The security considerations in VANETs [14][21][22][23] are of exceptionally high importance, as a malicious attacker can trigger traffic jams or even accidents by injecting false information in the system.

In this paper, we are primarily concerned about the defense against "Sybil attack" in a VANET. Sybil attack was first described by Douceur [10] in the context of peer-to-peer networks. It allows a malicious sender to create multiple Sybil nodes (not real nodes) to impersonate other (virtual) nodes. The Sybil attack is particularly harmful in a VANET [6][11][12][18][22][27][29] because it can mislead the drivers about the current traffic situation.



(1) Timestamp series-based approach (2) Temporary certificate-based approach

Fig 1. Basic ideas of proposed approaches.

In order to protect against the Sybil attack, vehicles must be issued trusted certificates [10][24], which make impersonating impossible. In this paper, based on our preliminary work [19], we introduce two temporary certificate-based approaches. Our schemes are designed to satisfy two goals: (1) to minimize the requirements of the system architecture and the computational costs for the certificate management; and (2) the ability to deploy in an early-stage VANET.

As the deployment of VANETs will be gradual, in early-stage VANETs, only a small fraction of the vehicles on the road will be smart vehicles with the capability to participate in the network [3][20]. In such a system, neighboring-based Sybil attack defense [9][16][27] may not be applicable.

In addition, initially only the most essential infrastructure components will be present. A dedicated vehicular public key infrastructure (VPKI) for certifying individual vehicles [1][7][13][21][22][23][25] will be likely slow to emerge due

[1] The author is with the Department of Internet and Media Engineering, Konkuk University, Seoul 143-701, South Korea
(e-mail: soyoungpark@konkuk.ac.kr)

[2] The authors are with the Department of Electrical Engineering and Computer Science, University of Central Florida, Orlando, FL 32816 USA (e-mail: ababer@eecs.uf.edu; turgut@eecs.ucf.edu; czou@eecs.ucf.edu)

to the challenges of bringing the manufacturers distributed across several countries as well as the vehicle registration authorities to agree on a unique certificate model. Using of long-term certificates also involve certificate-management problems such as certificate issuing, distributing, storing, and revocation. Additionally, the use of the long-term certificates raises concerns about privacy because it allows long-term tracking and collecting information about driver behavior.

Our architecture does not require long-term certificates and only needs supports from a small number of roadside units (RSUs) [15][26][28]. The vehicles obtain temporary certificates driving by the RSUs. The vehicles do not need to be pre-registered or pre-certified by trustworthy authorities to use the temporary certificates. Only RSUs are managed by certificate authorities (CA).

The two approaches we propose differ in the type of the certificates issued by RSUs (see Fig. 1).

Series of timestamp certificates: Each RSU issues certificates which contain the current timestamp signed by the RSU. Thus, the vehicles driving by the RSUs obtain a series of timestamp certificates showing their recent driving route and the time where they passed by at the RSUs. Due to the variance of the movement patterns of the vehicles, the probability that two vehicles pass by multiple RSUs at the same time is very low. Thus, the Sybil attack can be detected if two traffic messages have similar timestamp series.

Temporary certificate: Each RSU issues temporary key pairs and certificates that are only valid for a particular local area covered by the RSU for a limited time. To obtain the first certificate, a vehicle needs to find an RSU equipped with a camera or other devices for which allow physical authentication. Once the vehicle obtains its first temporary key and certificate, it renews its key pair and certificate with next RSUs as a type of chained certificate. The certificate chain binds the current certificate with its previous certificate, which means each vehicle uses one and only temporary certificate for a single spatial interval. The uniqueness of the certificate prevents the Sybil attack.

The remainder of this paper is organized as follows. Section II reviews the related work. Our system model, assumptions and goals are specified in Section III. We describe our two approaches in Section IV, V and provide system analysis in VI. The performance evaluations of our schemes are included in Section VI.D. Finally, we conclude our paper in Section VIII.

II. Related work

Detecting and defending against the Sybil attack has been the subject of extensive research. Deploying a public key infrastructure for individual vehicles (VPKI) is the most common solution for defending against Sybil attack [22][23]. As each vehicle will be issued a single certificate, the Sybil attack can be prevented. It still remains, however the possibility that the attacker uses valid certificates, which, however, were issued to a different vehicle. This problem can be solved with a multifactor authentication scheme [18], which provides an enhanced security of certificate. The certificate contains not only the public key information but also a set of physical attribute values of a vehicle recorded by the CA: the radio frequency fingerprint, the transmitter coverage, and so on. This makes using of stolen keys and certificates of other vehicles more difficult.

Studer et al. [26] proposed the use of temporary certificates for protecting the driver's privacy. Vehicles can obtain temporary keys and certificates anonymously from RSUs. Updating temporary certificates from RSU is similar to our approaches; however, in the proposed scheme, a smart vehicle needs a long-term privacy-preserving key pre-distributed by a trusted authority to obtain the temporary anonymous certificate. Xue et al [28] has proposed privacy-preserving authentication scheme with the support of RSUs. It exploits each RSU as a group manger of vehicles in its covering area, then each RSU updates group signature of vehicles joining the area.

Zhou et al. [29] proposed a privacy preserving method for detecting the Sybil attack with trustable roadside boxes and pseudonyms. Vehicles are assigned a pool of pseudonyms from the department of motor vehicles (DMV) and use them for generating traffic messages instead of the real identities for the privacy. Since the pseudonyms belonging to a vehicle are hashed to a unique value, the vehicles cannot abuse those pseudonyms for the Sybil attack. The roadside boxes and the DMV are connected together such that any suspicious pseudonyms can be detected by their cooperation. Even though the suggested scheme provides the vehicle's privacy, it is still based on the assumption that individual vehicles are registered to and managed by trusted authorities.

Guette and Bryce [11] suggested a secure hardware-based method built on the trusted platform module (TPM). Secure information and related protocols are stored in shielded locations of the module where any forging or manufacturing of data is impossible. Since the platform credentials are trusted by car manufacturers, the communications between TPMs of vehicles are protected from the Sybil attack. But, as the TPM is an improved variation of certificate, it still needs trusted authorities for managing individual vehicles.

The previously discussed approaches all rely on a cryptographic certificate-based method. Another type of proposed defenses against the Sybil attack rely on the physical properties of the nodes.

One approach uses resource testing [10][17] based on the computational and storage capabilities, communication bandwidth, and so on. It is based on the idea of broadcasting a request whose answer requires a certain amount of resource consumption. Then, only replies given within a pre-defined time interval are accepted. Since smart vehicles are equipped with a powerful device to compute expensive operations such as encryption, digital signature and so on, this kind of approach is not adequate for detecting the Sybil attack in a VANET. J. Newsome et al. [17] proposed radio resource testing and pair-wise key based Sybil attack detection method in a static wireless sensor network. Due to the high mobility of VANET nodes and the impossibility of pre-deploying shared information among vehicles, the approach cannot be used for VANET.

Another approach uses received signal strength (RSS) [9][16][27] to deduce the inconsistency between the claimed identities. The RSS of the packets from the Sybil identities will be nearly same at arbitrary receiver. An attacker might, however, manipulate its transmission power to fake the signals as coming from different positions. Such manipulation, however, can be detected by the collaboration of multiple neighbor nodes, which compute the sender's RSS ratio at multiple receivers. Guette et al. [12] analyzed the effectiveness of the Sybil attack in various assumptions of transmission signal tuning and antenna and then showed the limitation of RSS based Sybil detection in VANET.

III. System Model

A. Assumptions

We consider an early-stage VANET, where no dedicated vehicular public key infrastructure (VPKI) exists and the penetration of smart vehicles is small. The system architecture consists of roadside units and certificate authorities (CAs) for managing the RSUs. In the following, we describe our assumptions about each of the components of the system.

- **Smart vehicle:** It has an on-board unit (OBU) for networking and computing messages, GPS for location detection, and a digital map including geographical road information. It does not have a long-term public/private key pair.
- **Roadside unit (RSU):** It has a transmitter for sending and receiving messages from the passing by vehicles. It is not required to have Internet access. It has a tamper-proof device for storing secure information and generating either certified random key pairs or certified timestamps. Each RSU has its own private-public key pair and its certificate issued by the CA. Some RSUs are equipped with means for the physical authentication of a vehicle (for instance, a camera).
- **CA:** It manages RSUs and issues certificates for the individual RSU's public key. The CA's public key is pre-installed in OBU of each vehicle.

We assume that a malicious vehicle *M* has the following abilities:

- *M* can collect any information spread over the network.
- *M* has its own manufactured computational device for creating forged GPS information, fake traffic information and authentication information such as digital signature.
- *M* has a non-standard networking device (its own manufactured device) which allows it to manipulate any networking related information.

B. Security goals

Considering an environment in which the assumptions stated above hold, we aim to develop protocols which satisfy two security goals.

- **Prevention of Sybil attack**: Any receiver node can detect the Sybil attack of a malicious vehicle.
- **Driver's privacy protection**: It is difficult to track or trace the movement of a vehicle from traffic messages in the long-term.

IV. Timestamp Series-Based Data Propagation

In this section, we describe the timestamp series-based scheme in detail. First, we suggest a basic protocol suitable for simple roadway architecture such as highways. Then, we address a couple of limitations and discuss the challenges of extending the scheme to urban environments. Unlike highways, urban roadways have a complex topology with lots of signals, intersections and obstacles. Finally, we provide an architecture that can solve those challenges.

A. Basic scheme

On multiple lane roads with no heavy traffic congestion, vehicles move with different speeds depending on the limitations of the vehicle and the preferences of the driver. Our scheme exploits the fact that it is extremely rare that two vehicles pass by several RSUs situated far apart from each other at exactly the same time. In this scheme, the RSUs issue certificates of current timestamp signed by an RSU. The traffic messages must be authenticated with these certificates; traffic messages containing series of very similar timestamps are a sign of a Sybil attack.

The timestamp certificate plays two important roles: (1) it tells the time when a vehicle passed by an RSU, and (2) it shows the recent route of the vehicle by means of the RSU information issued the timestamp certificates. The uniqueness of each vehicle is determined by a series of timestamp certificates given to a vehicle. Therefore, the timestamp certificates have to be unforgeable and non-transferable. Only RSUs should be able to create and issue the timestamp certificates. Only the owner of timestamp certificates obtained directly from RSUs can use them in its traffic messages. Vehicles that do not own the timestamp certificate should not be able to pass them as their own.

1) Timestamp certificate update

Whenever a vehicle encounters an RSU, the vehicle carries out a timestamp certificate update protocol to obtain an up-to-date timestamp according to its trajectory. In order to make a vehicle unique, each vehicle needs to keep at least two recently obtained timestamp certificates issued by distinct RSUs. Those timestamp certificates will be included in every traffic message for preventing Sybil attack.

A straightforward way of updating the timestamp certificate is for a vehicle to request an independent timestamp certificate from each RSU. However, appending these timestamp certificates to a traffic message unnecessarily increases size of the message. Since each timestamp certificate is digitally signed by the private key of the issuing RSU, a certificate about the RSU's public key is additionally required to verify the validity of the timestamp certificate. In other words, each traffic message needs not only two timestamp certificates but also two additional certificates about the public keys of the two issuing RSUs.

An additional problem is that in a traffic-congested situation, vehicles move very slowly so they could receive

similar timestamp certificates from the same RSUs located around the congested area. In this case, more than two timestamps may be required in order to differentiate individual vehicle's trajectory.

Hence, we propose an aggregated timestamp certificate in order to minimize the traffic message size. The aggregated timestamp certificate contains both the newly issued timestamp and the previous timestamp, but it is signed only by the latest RSU. Thus, the traffic message size is reduced, because each traffic message needs only a single certificate of the RSU for verifying the correctness of the timestamp certificate.

Now, we give a detail description about the timestamp certificate update protocol. Table 1 summarizes the notations used through the rest of the paper.

Suppose that a vehicle V is passing by the i-th RSU R_i, for $i \geq 1$. Figure 2 shows the timestamp certificate update protocol between V and R_i.

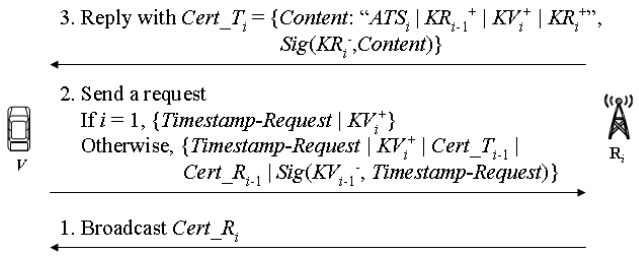


Fig 2. A timestamp certificate update protocol

1. R_i periodically broadcasts its certificate $Cert\_R_i = \{KR_i^+ \mid location, Sig(KCA^-, KR_i^+ \mid location)\}$ issued by the CA. V verifies the correctness of the certificate based on the CA's public key, which is hard-coded into each vehicle's OBU.
2. Once R_i's certificate is verified, V randomly selects its new private-public key pair (KV_i^-, KV_i^+) and generates a timestamp request. If $i = 1$, the request only includes $\{Timestamp\text{-}Request \mid KV_i^+\}$ since the vehicle has no previous timestamp. Otherwise, the request includes $\{Timestamp\text{-}Request \mid KV_i^+ \mid Cert\_T_{i-1} \mid Cert\_R_{i-1} \mid Sig(KV_{i-1}^-, Timestamp\text{-}Request)\}$. The private-public key pair is necessary to prevent any malicious vehicle from eavesdropping and stealing V's timestamp certificate. R_i can obtain the information about V's previous timestamp and RSU from $Cert\_T_{i-1}$. $Cert\_R_{i-1}$ is required for R_i to verify the correctness of $Cert\_T_{i-1}$. Lastly, the signature is a proof that V is the owner of the certificate $Cert\_T_{i-1}$.
3. For the given request, R_i first checks if $Cert\_T_{i-1}$ is issued by one of its adjacent RSUs. If so, R_i verifies the validity of $Cert\_T_{i-1}$ with $Cert\_R_{i-1}$ and KCA^+. If invalid, R_i will give no response to the request. Otherwise, R_i creates a new timestamp TS_i, and extracts the previous timestamp information "$TS_{i-1} \mid TS_{i-2} \mid \ldots$" from $Cert\_T_{i-1}$ as well. Then, R_i generates an aggregated timestamp ATS_i by concatenating TS_i to the extracted timestamps. Finally, a new timestamp certificate $Cert\_T_i = \{Content$: "$ATS_i \mid KR_{i-1}^+ \mid KV_i^+ \mid KR_i^+$", $Sig(KR_i^-, Content), Cert\_R_i\}$ is broadcasted.

The R_i and R_{i-1} must be different. If R_i and R_{i-1} are the same, R_i only updates the aggregated timestamp $ATS_i = TS_{i-1} \mid TS_i$ with current timestamp TS_i for the same public key KV_{i-1}.

The cryptographic hardness of the digital signature guarantees that a timestamp certificate of R_i cannot be forged by any other vehicles or RSUs that do not know about the private key of R_i. On the other hand, vehicles can easily intercept and collect timestamp certificates spread over the network. A malicious vehicle may try to use V's timestamp certificate for its traffic messages. As shown in above protocol, the timestamp certificate contains V's chosen public key. The corresponding private key is required to generate valid traffic messages of V for Sybil attack prevention. Thus, it is impossible for other vehicles that do not know about the private key of V to use V's timestamp certificate for their messages. Finally, the timestamp certificate satisfies both unforgeability and non-transferability.

2) Aggregation of timestamps

In this section, we describe the timestamp aggregation algorithm in detail. The algorithm is carried out by each RSU whenever it receives a timestamp request from a vehicle. The aggregation is done by concatenating a new timestamp to a given previous timestamps. In a normal traffic situation, the aggregated timestamp consists of at least two timestamps issued by recent two adjacent RSUs. On the other hand, in a traffic-congested situation, the aggregation may need a series of more than two timestamps in order to differentiate each vehicle. Thus, each RSU needs to decide the minimum number of timestamps for the aggregation.

The aggregation rule is as follows. Let an RSU be R, and neighboring RSUs adjacent to R be NR_j for $j \geq 1$. R maintains a list of up-to-date timestamps issued by each NR_j, which are updated whenever R receives requests.

When R receives a request, R discovers a neighboring RSU that created the timestamp certificate involved in the request. Suppose that the request contains a timestamp certificate $Cert\_T_n$ issued by NR_j. Let the timestamp values of $Cert\_T_n$ be $<TS_n \mid TS_{n-1} \mid TS_{n-2} \mid \ldots>$. Suppose that R already stores a series of timestamps $<TS_j \mid TS_{j-1} \mid TS_{j-2} \mid \ldots >$ for NR_j. The previous timestamps will be replaced with the new timestamps after responding to the request with a new aggregated timestamp certificate.

Second, in order to create an aggregated timestamp certificate for the request, R creates current timestamp TS_{n+1} for the new request, and then compares the given timestamps with the stored timestamps to decide the minimum number of timestamps required for the aggregation. R compares the similarity of the two series of timestamps by investigating the similarity of corresponding entries. Finally, R creates a new aggregated timestamp as follows:

1. R finds the first dissimilar values in the two series (the threshold ε of similarity is defined by the Sybil attack detection procedure, which will be discussed in the next section). R extracts a series of timestamps from the first value to the first dissimilar value of the given timestamps.

A new aggregated timestamp is generated by concatenating TS_{n+1} to the extracted series of timestamps. For example, if TS_n is different from TS_j then a new aggregated timestamp is $ATS_{n+1} = <TS_{n+1} \mid TS_{n,}>$. Or, if $< TS_n, TS_j >$ and $< TS_{n-1}, TS_{j-1}>$ are similar but $< TS_{n-2}, TS_{j-2} >$ are different from each other, a new aggregated timestamp is $ATS_{n+1} = <TS_{n+1} \mid TS_n \mid TS_{n-1} \mid TS_{n-2}>$.

2. If R does not find any dissimilar values in the two series, R keeps the entire series of timestamps of the given certificate. A new aggregated timestamps is generated by concatenating TS_{n+1} to the whole series, i.e., $ATS_{n+1} = <TS_{n+1} \mid TS_n \mid TS_{n-1} \mid TS_{n-2} \mid \ldots>$.

3) Use of timestamps series for Sybil attack prevention

A vehicle creates and broadcasts its traffic message about "*Data*" which the vehicle senses periodically or occasionally. "*Data*" may include traffic events, GPS information, moving direction, speed, time, and so on. After the vehicle passing the i-th RSU, for $i \geq 2$, the vehicle's current timestamp certificate includes at least two or more timestamps. A traffic message sent out by the vehicle has the following format:

$$TM = \{Data, Sig(KV_i^-, Data), Cert\_T_i, Cert\_R_i\}$$

The signature proves that the traffic data are created by a vehicle that possesses a valid timestamp certificate $Cert\_T_i$. Any receiver can verify the validity of the signed *Data* by the public key KV_i^+ contained in the certificate $Cert\_T_i$, as well as the validity of $Cert\_T_i$ itself with the RSU's certificate $Cert\_R_i$. The verification includes a check if $Cert\_T_i$ is issued by the nearest RSU from the GPS information in *Data*. Notice that the digital map of vehicles can inform about the nearest RSU. If all verification turned out true, the traffic message is valid. Otherwise, the message is ignored or discarded by a receiver.

Depending on the application, the traffic message might be propagated through multiple hops. The signed data $Sig(KV_i^-, Data)$ and the certificates $Cert\_T_i$, $Cert\_R_i$ will prevent any malicious intermediate vehicle from modifying or forging the propagated message.

On the other hand, if $i = 1$, the timestamp certificate would contain only a single timestamp. Any traffic message with such a certificate may be ignored by a receiver, because there are not enough timestamps for Sybil attack detection.

Among all the traffic messages, a Sybil attack can be detected as follows: Let arbitrary two traffic messages delivered to a receiver be $TM_1 = \{Data_i, Sig_i, Cert\_T_i, Cert\_R_i\}$ and $TM_2 = \{Data_j, Sig_j, Cert\_T_j, Cert\_R_j\}$. The receiver decides that the two messages are Sybil messages if all the following conditions are satisfied:

- The RSU information given from $Cert\_R_i$ and $Cert\_R_j$ are identical,
- $Cert\_T_i$ and $Cert\_T_j$ are issued by the same RSU specified as in $Cert\_R_i$ and $Cert\_R_j$,
- KR_{i-1}^+ in Cert_T$_i$ and KR_{j-1}^+ in $Cert\_T_j$ are identical, and
- $|TS_i - TS_j| < \varepsilon$ and $|TS_{i-1} - TS_{j-1}| < \varepsilon$, where the value of ε will be discussed in detail in Section 6.

The Sybil detection exploits the probability that arbitrary two vehicles passing by the same two or more RSUs at the very same time is low especially in a normal traffic situation. If there is traffic congestion, the aggregated timestamp will have more than two timestamp values. Even in this case, the aforementioned Sybil attack detection procedure still works except that the check of two timestamps should be extended to the check of all timestamps contained in $Cert\_T_i$ and $Cert\_T_j$.

Some vehicles may try to make multiple timestamp certificate requests to a single RSU to attempt a Sybil attack with those multiple certificates. The timestamps obtained within the transmission range of a single RSU, however, are tied together with very short time differences. Therefore, even if a malicious vehicle creates distinct messages signed by different keys, those messages are highly suspected as Sybil attack because of the similarity of timestamp certificates.

B. Challenges

The basic scheme cannot be applied in a straightforward way to an urban environment with a complex roadway architecture consisting of many signals and intersections. We briefly review the two main challenges of why the basic scheme is not suitable for urban environment.

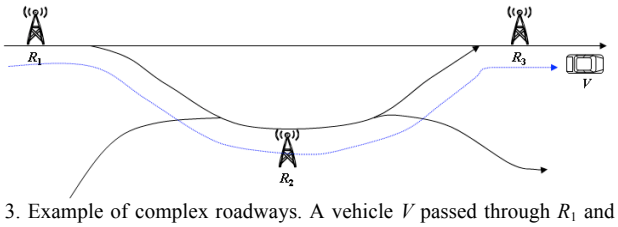


Fig 3. Example of complex roadways. A vehicle V passed through R_1 and R_2 so it can attempt to obtain at least two different certificates from RSU R_3, since both R_1 and R_2 are adjacent to R_3. One is based on the certificate from R_1 and the other is with the certificate from R_2.

1. Complex roadways: Even though it is impossible for a single vehicle to follow multiple paths concurrently, a malicious vehicle may exploit the complex road map for making the Sybil attack. Figure 3 shows an example. Suppose that the malicious vehicle V drove through RSUs denoted as R_1 and R_2, and that V attempts to make a new certificate request to R_3. Let $Cert\_T_1$ and $Cert\_T_2$ be the timestamp certificates obtained from R_1 and R_2, respectively. V can make two distinct timestamp requests by pretending to have driven through R_1 to R_3 for one request, and to have driven through R_1, R_2 and R_3 for the other. From the view of R_3, both R_1 and R_2 are adjacent to R_3 so R_3 should issue a new timestamp certificate $Cert\_T_{31}$ based on $Cert\_T_1$ and $Cert\_T_{32}$ based on $Cert\_T_2$. Therefore, V can produce two Sybil messages by using both $Cert\_T_{31}$ and $Cert\_T_{32}$ that show different driving routes. Consequently, we need a careful deployment of RSUs to prevent such an attack scenario.
2. Frequent stops at intersections: The urban traffic environment has numerous intersections with signals. Vehicles tend to stuck together at those intersections, and hence, synchronize their moving dynamics. If RSUs are located at intersections, it may make the Sybil attack detection difficult because of the synchronization. Therefore, we need to avoid deploying RSUs at

intersections.

C. Deployment of RSUs

In this section, we provide a dedicated construction that can solve the aforementioned two challenges. RSUs need to be deployed with a small restriction. Suppose a directed roadway graph, such that any merging point of distinct roadways including intersections is defined as a vertex, and individual roadway as an edge.

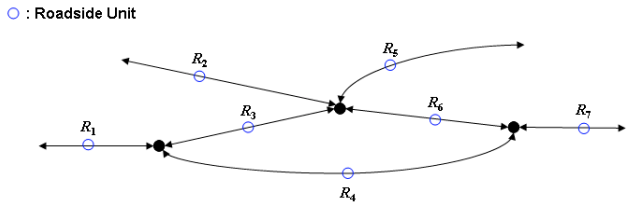


Fig 4. Deployment of RSUs on a roadway graph: RSUs are deployed on every edge of the graph.

RSUs should be located on every edge as shown in Fig. 4. With this deployment, every path has a unique combination of RSUs. Thus, a Sybil attack exploiting the RSU deployment on the complex roadway as mentioned in the previous section is not possible. The deployment will also solve the second challenge as it avoids intersections.

Excluding intersections for setting up RSUs may make it difficult to combine our system with some special intersection-related VANET applications including data synchronization at the intersection or mixing vehicles at the intersection for preserving anonymity. In such situations, additional RSUs need to be placed at the intersections for supporting other possible security issues.

V. Temporary Certificate-Based Data Propagation

In this section, we suggest another way of building trustable temporary certificates. The basic idea is to assign each vehicle a single temporary certificate with a predefined *spatial* and *temporal* constraint. To achieve this, each RSU creates a random temporary key pair and certificate valid within a particular area for a limited time. Every vehicle can obtain such a key pair and certificate by driving by the RSU. The vehicle needs to refresh its key pair and certificate at every encountered RSU. Once the key pair and its certificate are updated, the previous keys and certificates become invalid. Since a vehicle can have only one valid key pair and certificate, the Sybil attack is prevented.

The temporary certificate scheme requires two steps: (1) initialization to obtain an initial key pair and certificate, and (2) certificate update. In the initialization step, if a vehicle makes an initial certificate request to an RSU, the RSU needs to authenticate the vehicle to protect against multiple requests originating from the same vehicle. Since we do not assume to use VPKI for individual vehicles, the vehicle does not possess any trustable authentication information, such as a long-term key pair and its certificate or a certificate on its OBU, except the vehicle itself. Thus, we propose a practical way of physical authentication between a vehicle and an RSU.

Once the initial key pair and its certificate are obtained, the certificate is updated in a chain style along with the certificate update protocol.

A. Initialization of Temporary Certificate

We borrow the basic concept of "seeing-is-believing" for the initial physical authentication. To achieve this, we assume that some RSUs are equipped with a camera and an analyzer that can take a picture of license plate of a moving vehicle, then extract the license information from the picture [4]. Cameras can be easily installed at the side or top of a road, and have been already used for detecting speeding in the real world. If a vehicle meets a particular RSU equipped with a camera (knowing this RSU's specialty based on the handshake message sent back from the RSU), then it can issue a new certificate request for an initial key pair and its certificate.

Let R_1 be a camera-equipped RSU that a vehicle V meets for the first time. Fig. 5 shows the initialization protocol between V and R_1.

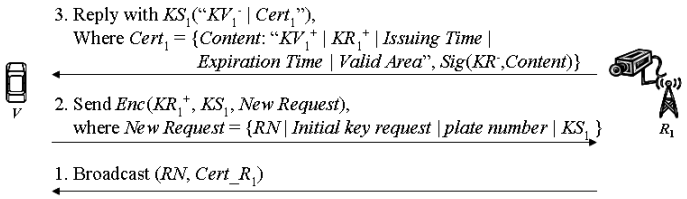


Fig 5. Initialization protocol for the temporary certificate.

1. R_1 periodically broadcasts a random nonce *RN* and its certificate *Cert_R*$_1$ (containing RSU type and its public key KR_1^+). In addition, it takes a picture of every vehicle entering into its camera zone.
2. V checks the validity of R_1's public key KR_1^+ with *Cert_R*$_1$. If the certificate is valid, V selects a random symmetric session key KS_1 and creates "*New Request*" that includes {*RN* | *Initial key request* | *plate number* | KS_1}. Then, V sends out an encrypted request *Enc*(KR_1^+, KS_1, *New Request*).
3. R_1 checks if the plate number in the request matches with one of camera-sensed values, and if the request includes a valid *RN* issued by the RSU recently. If every condition is satisfied, R_1 creates a random key pair (KV_1^-, KV_1^+) and its certificate $Cert_1$ for V. $Cert_1$ basically includes {*Content*: "KV_1^+ | KR_1^+ | *Issuing Time* | *Expiration Time* | *Valid Area*", *Sig*(KR^-,*Content*)}. The "*Valid Area*" is a single road-section defined as a spatial interval between two adjacent RSUs. Finally, R_1 sends a reply KS_1("KV_1^- | $Cert_1$") encrypted with the vehicle's symmetric key.

A camera that can precisely determine the license plate number of a vehicle requires high resolution and high demand on image processing. For cost efficiency, cheaper cameras can be used alternatively, which can detect a vehicle's physical features such as color, contour and driving lane, and so on. In this case, a combination of multiple features is required to differentiate the limited number of vehicles nearby an RSU.

B. Temporary Certificate update

Once a vehicle obtains an initial valid key pair and certificate, it can refresh them whenever it meets an RSU. The certificate update protocol is very similar to the initialization protocol except for two points. First, the update can be carried out by every RSU. Second, the update needs a right previous certificate to issue a new chained-certificate, such as a hash chain. This certificate chain is required to prevent a Sybil attack on complex roadways as shown in Fig. 3.

Suppose that a vehicle V meets the i-th RSU R_i for $i \geq 2$. Fig. 6 shows the temporary certificate update protocol between V and the R_i.

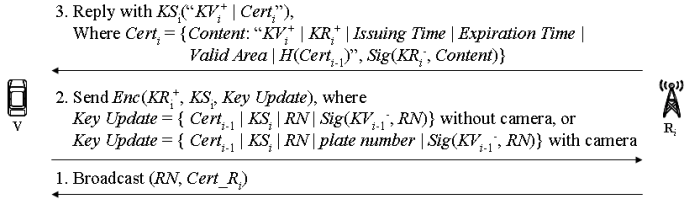


Fig 6. A temporary certificate update protocol.

1. The first step is identical to the initialization protocol.
2. V carries out the same step as the initialization except that V generates a "*Key Update*" request. The content of the request depends on the type of RSU. If R_i does not have a camera, the request includes $\{Cert_{i-1} \mid KS_i \mid RN \mid Sig(KV_{i-1}^-, RN)\}$. Otherwise, the request additionally includes V's plate number, such as $\{Cert_{i-1} \mid KS_i \mid RN \mid plate\ number \mid Sig(KV_{i-1}^-, RN)\}$. If R_i has a camera, a malicious V may try to make an additional new request along with its update request at the same time in order to get two different key pairs and certificates. Thus, for only a camera-equipped RSU, the update protocol needs the physical authentication to avoid above-mentioned attack. The signature on RN is required to show that V is the owner of its previous certificate $Cert_{i-1}$. A valid signature proves that V knows the corresponding private key of the public key written in $Cert_{i-1}$.
3. R_i first checks if $Cert_{i-1}$ is issued by one of its neighboring RSUs. If not, it rejects the request. Otherwise, it verifies the validity of the signature about RN. If the signature is valid, R_i creates a new key pair and its certificate. At this point, the new certificate $Cert_i$ additionally includes the hash value of the previous certificate $H(Cert_{i-1})$.

The update succeeds only if the request has a valid previous certificate. Therefore, V receives a single alternate. R_i declines a trial of V to update old and expired certificates. Thus, V always keeps a single valid temporary key pair and its certificate.

C. Use of the temporary certificate for Sybil attack prevention

The creation and verification of a traffic message is the same as described in section IV.A.3, except that a temporary certificate is used instead of timestamp certificate as follows:

$$TM = \{Data, Sig(KV_j^-, Data), Cert_j, Cert\_R_j\}$$

Unlike the timestamp series-based approach, a vehicle obtains one and only certificate at any time so Sybil attack is prevented.

Nevertheless, there could be a possibility that a vehicle can try to have two valid certificates. For a complex roadway as shown in Fig. 3, suppose that a malicious vehicle V passed through R_1 and R_2 then it makes a certificate update request to R_3. Let the certificates issued by R_1, R_2 be $Cert_1$ and $Cert_2$ respectively. V may attempt to make two distinct certificate-update-requests: one is using $Cert_1$ and the other with $Cert_2$. If both $Cert_1$ and $Cert_2$ were valid, R_3 would respond with two different key pairs and certificates, because both R_1 and R_2 are adjacent to R_3. To prevent this vulnerability, a certificate chain is used. For V to obtain $Cert_2$, V had to submit $Cert_1$ to R_2. Thus, $Cert_2$ has a hash value of $Cert_1$. If both $Cert_1$ and $Cert_2$ are submitted to R_3, then R_3 can easily detect that $Cert_1$ and $Cert_2$ belong to a same vehicle, since $Cert_2$ includes the information about $Cert_1$. R_3 will ignore and discard the request about $Cert_1$.

Using this scheme, a Sybil attack can be detected as follows: Let TM_1 and TM_2 be the arbitrary two traffic messages delivered to a receiver. $Cert_1$ and $Cert_2$ are the two certificates belonging to TM_1 and TM_2 respectively. The receiver decides that the two messages are Sybil messages if either $Cert_1$ contains the hash value of $Cert_2$, or $Cert_2$ contains the hash value of $Cert_1$.

VI. System Analysis

A. Comparison of two schemes

The advantage of the timestamp series-based approach is that it does not require RSUs with the ability to perform physical authentication. However, the approach requires a careful deployment of RSUs in urban environments. The main drawback is that the traffic message could be long and the message verification might take a long time as well. The detection of the Sybil attack relies on the probability of vehicle distribution, allowing false positives. However, the false-positive rate is low, and will be analyzed in section D.1.

On the other hand, the main advantage of the temporary certificate approach is that the Sybil attack can be always detected because of the unique assignment of temporary certificate to each vehicle. The size of traffic message is fixed and shorter than the first approach. The main drawback is the fact that this approach needs an initialization step and at least some RSUs must be equipped for the initial physical authentication between vehicles and the RSUs. The initial certificate is not issued until the vehicle meets a camera-equipped RSU. However, once the initial certificate is obtained, certificate update can be carried out at every RSU.

B. Driver's privacy protection

Let us now consider the privacy implications of our model. As the traffic messages are transmitted wirelessly and contain the certificates, an opponent can collect this information and use it to analyze and trace the movement patterns of the vehicles. This creates privacy issues. In general, we accept the fact that the current location of a vehicle is disclosed by its participation in the VANET. However, an attacker should not

be able to perform long-term tracking of the vehicle's trajectory.

The aggregated timestamp in the first approach shows two recent RSU information of a vehicle. Similarly, the temporary certificate in the second approach contains a hash value of its previous certificate. Thus, both certificates expose limited information of a vehicle's trajectory. This will facilitate the attacker in tracking a vehicle. However, if an eavesdropped sequence of certificates has any gap, the attacker cannot trace the vehicle any more. Due to the dynamic mobility of vehicles and the temporary use of certificates, the driver's privacy can be protected.

For the first approach, the timestamp update protocol can be modified slightly to provide stronger privacy protection. A simple way is to remove the previous RSU information from the certificate. However, this could create the following problem (shown in Fig. 7). If two vehicles V_1 and V_2 pass by two different RSUs R_1 and R_2 respectively almost at the same time, then both vehicles arrive at R_3 at the same time. Subsequently, both V_1 and V_2 will have a sequence of very similar timestamps. Without the previous RSU information, traffic messages with those timestamps will be treated as a Sybil attack.

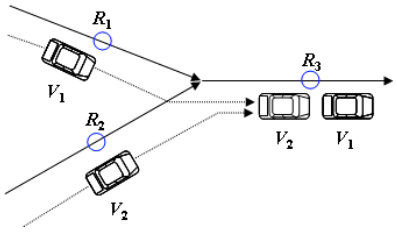


Fig 7. A special situation where two vehicles V_1 and V_2 have passed through RSU R_1 and R_2 at the same time, and then pass by R_3 at the same time as well.

An alternative way is to use pseudo-ids for previous RSUs instead of real public key information. When a vehicle makes a timestamp update request to R_3, R_3 can create pseudo-ids about its adjacent RSUs, such as PID_1 for $R1$ and PID_2 for R_2. Then, $Cert\_T_3$ for V_1 includes {*Content*: "$ATS_3 \mid PID_1 \mid KV_3^+ \mid KR_3^+$", $Sig(KR_3^-$, *Content*), $Cert\_R_3$} instead of KR_1^+. Likewise, $Cert\_T_3$' for V_2 includes {*Content*: "ATS_3' $\mid PID_2 \mid KV_3^+ \mid KR_3^+$", $Sig(KR_3^-$, *Content*), $Cert\_R_3$} instead of KR_2^+. Nobody but R_3 knows the exact RSU corresponding to the pseudo-id from the certificate. Since the previous RSU information is hidden from others, it is impossible to link any two certificates belonging to a vehicle. Besides, the modified protocol figures out the aforementioned particular situation. Even though the aggregated timestamps ATS_3 of V_1 and ATS_3' of V_2 are very similar, the certificates have different previous RSU information PID_1 and PID_2. Therefore, the certificates can avoid the misjudgment. The RSU may be able to change the pseudo-ids periodically or while there are no requests.

C. Tolerance to RSU failures

Both proposed schemes need at least two certificates issued by two recent adjacent RSUs for the certificate update and the Sybil attack detection. If an RSU is shut down by either any kind of attack or systemic errors, then vehicles that passed through the RSU will fail to update their certificates and cannot generate valid traffic messages.

We suggest two approaches to solve this problem. The first approach is based on vehicle-assisted alerts. Every vehicle that detected a broken RSU R_i gives a notice about the broken RSU to next RSUs according to each vehicle's trajectory. Subsequently, next RSU R_{i+1} will collect similar messages from various vehicles. If a certain amount of messages is collected, R_{i+1} will begin to create a certificate that contains the fact about the broken RSU R_i. Without a certificate issued from R_i, the vehicles can update their certificates at R_{i+1} with the certificate given from R_{i-1} so that those certificates will be used for the Sybil attack detection. However, there is a delay until the nearby RSU recognizes the broken RSU. Thus, the vehicles detecting the broken RSU for the first time will not obtain valid certificates and will need to restart their certificates from R_{i+1}.

The second approach assumes that RSUs have the Internet access. In this case, broken RSUs can be detected by the absence of heartbeat messages. If an RSU is broken, the nearby RSUs will create a certificate that contains the particular event. The vehicles can keep obtaining valid certificates and creating valid traffic messages without any delay.

D. Analysis of False-Positive Rate

1) Mathematical analysis

The timestamp series approach relies on the observation that it is rare for two vehicles to have the same trajectory, i.e., to pass two (or more) different RSUs at almost the same time. If such an event happens, the Sybil detection procedure will mistakenly treat messages issued by these vehicles as a Sybil attack. In this section, we provide the mathematical analysis of this approach's false-positive rate.

We assume that vehicles drive independently with each other. This assumption is reasonable when (1) no heavy congestion occurs and (2) the road has two or more driving lanes. If these two assumptions are not satisfied, multiple vehicles could move as a cluster with a similar trajectory. This will make these vehicles have a higher false-positive rate in Sybil attack detection than what we analyze here.

Let us assume that vehicles average driving speed is s, the radio transmission range of an RSU is Q, and the distance between two RSUs is D. Thus, the time for a vehicle driving through an RSU's transmission range is $T = 2Q/s$. A vehicle could receive its timestamp from an RSU at any time when it passes through the RSU's radio coverage area, thus a single vehicle could obtain multiple timestamps from an RSU within time T. For this reason, in the proposed timestamp series approach, a Sybil attack is detected if (procedure in Section IV.A.3):

$$|TS_i - TS_j| < T \text{ and } |TS_{i-1} - TS_{j-1}| < T \quad (1)$$

Therefore, the ε of the Sybil attack detection procedure described in Section IV.A.3 is defined as T.

The false-positive rate of this approach is the probability that two vehicles driving on the same road satisfy the condition in Equation (1).

When vehicles move independently, it is reasonable to assume that the arrival process of vehicles at the first RSU can be modeled as a Poisson arrival stochastic process (if we assume each lane's vehicle arrival follows Poisson process, due to independence, vehicle arrival on the whole road can also be modeled as a Poisson process with a rate equal to the summation of each lane's Poisson process rate). Suppose that the arrival rate is λ. Then the probability that two vehicles have $|TS_{i-1} - TS_{j-1}| < T$ at the first RSU is as follows:

$$\begin{aligned} &Prob(\text{Two arrival vehicles have } |TS_{i-1} - TS_{j-1}| < T) \\ &= Prob(\text{Inter-arrival time} < T) \\ &= 1 - e^{-\lambda T} \end{aligned} \quad (2)$$

Since the average speed of a vehicle is s and the distance between these two RSUs is D, it will take D/s time on average for a vehicle to arrive at the second RSU. It is reasonable to assume that each vehicle takes a normal distributed time to arrive at the second RSU, i.e., $(TS_i - TS_{i-1})$ and $(TS_j - TS_{j-1})$ follow the normal distribution with the mean value of D/s and the variance denoted by σ^2.

Comparing with D/s, the time T in Equation (2) is usually negligible when RSUs are not densely distributed. Thus those two vehicles satisfying Equation (2) can be treated as leaving the first RSU at the same time to move towards the second RSU; and hence, the variable

$$TS_i - TS_j \approx (TS_i - TS_{i-1}) - (TS_j - TS_{j-1}) \quad (3)$$

also follows the normal distribution with mean 0 and variance $2\sigma^2$ [5]. Therefore, we can derive:

$$\begin{aligned} &Prob\ (|TS_i - TS_j| < T \text{ given } |TS_{i-1} - TS_{j-1}| < T) \\ &= \frac{1}{\sqrt{4\pi}\sigma}\int_{-T}^{T} e^{-x^2/4\sigma^2} dx \end{aligned} \quad (4)$$

Based on Equation (2), (4), and the formula of conditional probability [5], we can derive the false-positive rate of our approach as:

$$\begin{aligned} &Prob(|TS_i - TS_j| < T \text{ and } |TS_{i-1} - TS_{j-1}| < T) \\ &= Prob(|TS_{i-1} - TS_{j-1}| < T) \times \\ &\quad Prob(|TS_i - TS_j| < T \text{ given } |TS_{i-1} - TS_{j-1}| < T) \\ &= (1 - e^{-\lambda T}) \cdot \frac{1}{\sqrt{4\pi}\sigma}\int_{-T}^{T} e^{-x^2/4\sigma^2} dx \end{aligned} \quad (5)$$

As the distance between two RSUs becomes larger, $(TS_i - TS_{i-1})$, $(TS_j - TS_{j-1})$ and the variance σ^2 will get bigger. This will result in a smaller false-positive rate as shown in Equation (5). In addition, if an RSU has smaller transmission range Q, or vehicles drive at higher speed s, T becomes smaller. Consequently, the false-positive rate gets smaller by Equation (5).

If we rely on more than two consecutive timestamps to make the Sybil attack decision, we will have smaller false-positive rate, which can be derived in a similar way to the above analysis.

2) Simulation Verification

We simulated the timestamp series-based data propagation to study its false-positive rate by using the same assumptions as in the above mathematical analysis. We assumed that vehicles arrive at the first RSU following the Poisson arrival process. Each vehicle takes a normal distributed time to reach at the second RSU (average time is D/s). If we assume that the density of vehicles, denoted by *Density*, is known, then vehicle Poisson arrival process should have a rate of $\lambda = Density \times s$.

The simulation carried out with various s, Q, D and *Density*. We conduct 1000 simulation runs for each set of simulation setting in order to obtain the average false-positive rate.

With $Q=150$m, s=80km/hr and the stand deviation $\sigma = 0.2\times$ D/s, Fig. 8 shows the comparison of simulation results and the analytical results based on Equation (5) for two different vehicles' density scenarios. This figure shows that our analysis, Equation (5), is accurate. As the distance between RSUs increases, vehicles will have less chance to move with the same dynamics and the false-positive rate drops quickly to around 3%.

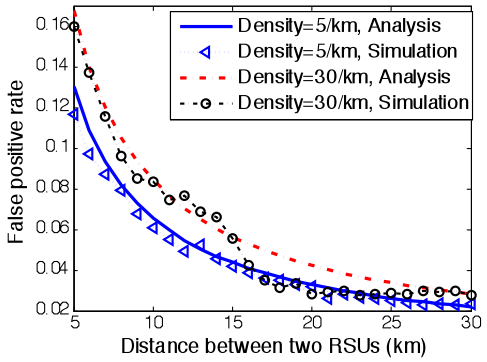


Fig 8. False positive rate for the time series approach for different vehicular density (Q=150m, s=80km/hr, $\sigma = 0.2\times D/s$)

Fig. 9 shows the results under different vehicular moving variability. If vehicles have less variant moving speeds, i.e., smaller value of the standard deviation σ, the false-positive rate of the approach will increase since many vehicles will obtain similar timestamps from RSUs.

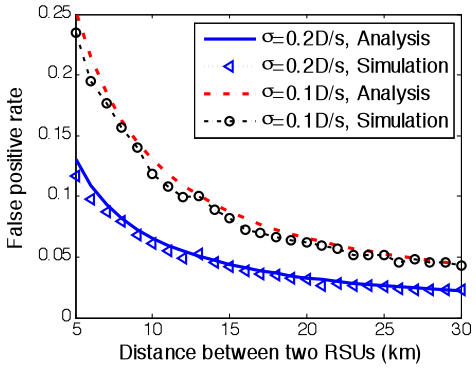


Fig 9. False positive rate for the timestamp series approach for different vehicular moving variability (Q=150m, s=80km/hr, *Density*=5/km)

VII. Simulation and Performance Evaluation

In this section, we present a qualitative evaluation of the performance for issuing and updating certificates in diverse traffic situations. The simulation scenario includes multiple vehicles and a single RSU. We have examined the performance in terms of four matrices including scalability, packet overhead, processing time, and latency. They are

defined as follows:

- Scalability: it measures how much the network performance changes as the number of vehicles increases. We show the scalability of our approaches by carrying out simulation with numbers of vehicles of 10, 30, 50, 100 and 200 within a range of 500 meters.
- Packet Overhead: it measures how many bytes of additional data appended to each traffic data message in order to provide the required security in our approaches.
- Processing time: it measures the computational resources required for the computing and processing the certificate update in our approaches.
- Latency: it is the time delay for updating timestamp certificate between vehicles and an RSU according to various densities of vehicles with different levels of speed. The simulation was carried out with different vehicular speeds of 10km/h, 30km/h, 70km/h and 100km/h.

Simulation experiments were carried out in NS-2 simulator using the MAC 802.11a module. In order to reduce the packet overhead, we assume elliptic curve cryptosystem for our basic signature scheme. Table 2 shows the summary of our simulation environment.

The key length of Elliptic Curve Digital Signature Algorithm (ECDSA) is 163 bits (21 bytes), and the corresponding signature size is 28 bytes [8]. The estimated signature generation time is 36.82ms and the verification requires 38.05ms [1]. SHA-1 generates a hash value of 20 bytes long.

A. Packet Overhead

As shown in Fig. 2, the timestamp certificate *Cert_T* is *Cert_T*$_i$ = {Content: "ATS_i | $KR_{i\text{-}1}^{+}$ | KV_i^{+} | KR_i^{+}", Sig(KR_i, Content)}, includes an aggregated timestamp, two public keys of adjacent RSUs, one public key of a vehicle and one digital signature about the message. Since every vehicle keeps only two adjacent timestamps in a normal traffic situation, the length of the aggregated timestamp is usually twice as long as a single timestamp. Let *len*() denote the length of an argument in bytes. Thus the total packet overhead for a single timestamp certificate is as follows:

$$\begin{aligned} len(Cert\_T) &= 2\times len(\text{Timestamp}) + 3\times len(\text{Key}) + len(\text{Sig}) \\ &= 2\times 8\text{bytes} + 3\times 21\text{bytes} + 28\text{bytes} \\ &= 107\text{bytes}. \end{aligned}$$

In traffic jammed situation, described in Section IV.A.2, as the length of jammed road section increases, the length of aggregated timestamp will increase proportionally to the number of RSUs belonging to the jammed section. For example, if a jammed road section covers three adjacent RSUs, the size of the aggregated timestamp will be 4×*len*(Timestamp). In such a situation, only the length of timestamps increases, and the additional overhead will be 16bytes. Due to the long distance between two RSUs, the aggregated timestamps will not have more than four timestamps.

Whenever a vehicle meets an RSU, it requests certificate update. As shown in Fig. 2, the request message contains {*Timestamp-Request* | KV_i+ | *Cert_T*$_{i\text{-}1}$ | *Cert_R*$_{i\text{-}1}$ | *Sig* (*Timestamp-Request*)}. The certificate of RSU *Cert_R* is a regular certificate to certify the public key of the RSU. It basically includes {KR^{+} | *Sig*(KR+)}. Thus, the minimum size of the certificate is 21bytes + 28bytes = 49bytes. However, it may contain additional RSU information including the RSU ID (8 bytes), timestamp (8 bytes) and valid period (5 bytes), and so on. Thus, we estimated the maximum size of certificate as 70 bytes. Consequently, the packet overhead of the request can be obtained as follows:

$$\begin{aligned} len(\text{Request}) &= len(KV^{+}) + len(Cert\_T) + len(Cert\_R) \\ &\quad + len(Sig) \\ &= 21\text{bytes} + 107\text{bytes} + 70\text{bytes} + 28\text{bytes} \\ &= 226\text{bytes} \end{aligned}$$

Next, we analyze the packet overhead in the temporary certificate approach. We only consider the temporary certificate update protocol, which happens at the RSU without camera since most RSUs are not equipped with camera. Unlike the timestamp certificate approach, the temporary certificate has a fixed size of length regardless of traffic situation. As shown in Fig. 6, the temporary certificate is {Content: "KV_i^{+} | KR_i^{+} | *Issuing Time* | *Expiration Time* | *Valid Area* | $H(Cert_{i\text{-}1})$", *Sig*(Content)}. Thus the certificate size is:

$$\begin{aligned} len(Cert) &= 2\times len(\text{Key}) + 2\times len(\text{Timestamp}) + len(\text{valid area}) \\ &\quad + \text{len(hash)} + len(\text{Sig}) \\ &= 2\times 8\text{bytes} + 2\times 21\text{bytes} + 8\text{bytes} + 28\text{bytes} + 20\text{bytes} \\ &= 114\text{bytes} \end{aligned}$$

The certificate update request message *Key Update* issued by a vehicle in Fig. 6 is {$Cert_{i\text{-}1}$ | KS_i | RN | *Sig*($KV_{i\text{-}1}^{-}$, RN). Then, the request size is:

$$\begin{aligned} len(\textit{Key Update}) &= len(Cert) + len(\text{Symmetric Session Key}) \\ &\quad + len(RN) + len(Sig) \\ &= 114\text{bytes} + 16\text{bytes} + 4\text{bytes} + 28\text{bytes} \\ &= 162\text{bytes} \end{aligned}$$

B. Processing Time

In the certificate update protocol in both proposed approaches, the most time-consuming processes are asymmetric cryptographic computations including asymmetric encryption/decryption and signature generation/verification. We only consider such heavy computations for estimating the processing time of our proposed schemes.

In the timestamp certificate updating, generating a request by a vehicle requires only one signature generation (36.82ms) [1]. On the other hand, creating a new timestamp certificate by an RSU needs two signature verifications and one signature generation. The total processing time of RSU is 2×38.05ms + 36.82ms = 112.92ms.

The temporary certificate updating protocol performs encrypted message communication. For the consistency with signature, we adopt Elliptic Curve Cryptography (ECC) encryption and decryption with a key of 163bits for our asymmetric algorithm. The estimated encryption and decryption time are 2.65ms and 1.31ms, respectively [1]. Generating an update request by a vehicle requires one signature generation (36.82ms), one asymmetric encryption (2.65ms) and one symmetric encryption (ignored). So the total processing time is 39.47ms.

Issuing a new certificate by an RSU requires one asymmetric decryption (1.31ms), one symmetric decryption (ignored), two signature verifications (76.1ms), one hash function (ignored), one signature generation (36.82ms) and one symmetric encryption (ignored). Therefore, the estimated total processing time is equal to 114.23ms.

Consequently, both approaches require similar processing time. Table 3 shows the detailed comparison of our two approaches in packet size and processing time.

C. Latency

Finally, we analyze the transmission delay and the transmission accuracy in the certificate update protocols in different traffic situations. For this simulation, a single RSU broadcasts its certificate periodically with an interval of 500ms. Once a vehicle catches the RSU broadcasting, it generates its certificate update request and broadcasts it. Then, the RSU issues a certificate for a given request and broadcasts it back. We define the transmission delay as a round-trip delay of messages. It computes the total time difference from the creation time of the request by a vehicle to the arrival time of the corresponding certificate at the vehicle as follows:

$T(\text{Delay}) = T(\text{Request}) + T(\text{Certificate}) + 2 \times T(\text{Transmission})$,

where $T()$ means the processing time for its argument.

The simulation is implemented from 10 to 200 vehicles randomly distributed within the range of 500m. In addition, for each density, the simulation is repeated with the same distribution of vehicles for average moving speeds of 30km/h, 70km/h and 100km/h. However, we excluded some unrealistic situations such as 200 vehicles within a range of 500m at a speed of 100km/h. We have run the simulation 10 times for each case to obtain correct average values.

Figure 10 shows the simulation results of the total transmission delay. 10 vehicles drive with an average speed of 30km/h generates the average transmission delay of 155.173ms. On the other hand, when 100 vehicles move at a speed of 100km/h, the average transmission delay is 159.665ms. When a small number of vehicles drive slowly, the transmission delay is almost the same as the processing time for generating request and issuing certificate. The transmission delay is highly affected by the density of vehicles and the moving speed of vehicles. As the density increases, due to the bottle neck at the RSU by lots of update requests and certificate responses, the transmission delay takes longer. However, the difference is not that significant since the maximal transmission delay only takes about 9ms longer than the minimum transmission delay.

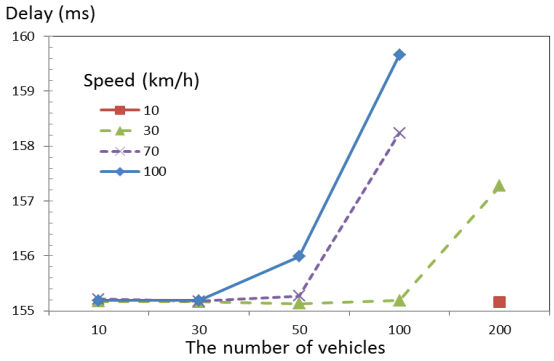


Fig 10. Transmission delay in updating certificate for various numbers of vehicles with different moving speeds (Q=150m, packet size=226bytes)

When a vehicle does not get the response from RSU for its first request within a predefined timeout period, it must retransmit the request, which will create additional delay. In order to know how often this retransmission happens, we come up with a metric "transmission accuracy", defined as the success probability of the RSU receiving the first request from a vehicle. Figure 11 shows the simulation results of the transmission accuracy. In the case of 30 vehicles, the transmission accuracy is 100%. When 200 vehicles drive at a speed of 30km/h, the accuracy is 99%. In a case of 50 vehicles moving at 100km/h, the accuracy is 98%. The worst case of accuracy of 96% is observed when 100 vehicles move with the speed of 100km/h.

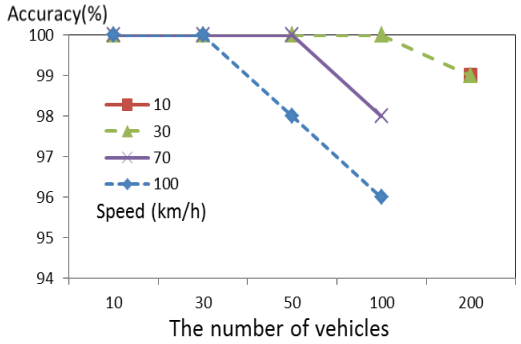


Fig 11. Transmission accuracy in updating certificate for various numbers of vehicles with different moving speeds (Q=150m, packet size=226bytes)

VIII. Conclusion

We proposed two practical ways of building trustable temporary certificates for defending against the Sybil attack in an early-deployment stage VANET environment. Our schemes require only RSUs on the roadway for issuing the certificates without any dedicated vehicular public key infrastructure for individual vehicle. Vehicles can obtain the certificates by driving by the RSUs. In the timestamp series-based data propagation, vehicles obtain non-transferable timestamps from RSUs. The aggregated timestamp given to a vehicle shows the vehicle's most recent trajectory and the time at particular locations. Due to the variance in the movement pattern of vehicles, a Sybil attack can be detected by the similarity of the

timestamps of the impersonated vehicles.

In the temporary certificate-based data propagation, a single temporary key pair and its certificate valid in a particular area for a limited time are assigned to a single vehicle. In order to satisfy the goal of assigning only one temporary certificate to one vehicle, two methods of physical authentication and certificate chain were used. Each vehicle obtains its initial key pair and certificate from a camera-equipped RSU by the physical authentication. After the initialization, a vehicle updates its key pair and certificate at every RSU. Because of the uniqueness of certificate, Sybil attack can be prevented all the time.

Since the Sybil attack detection mechanism of the timestamp series-based approach is based on the probability of vehicle distribution, we analyzed the false-positive decision rate mathematically. We also simulated it for simple roadway architecture. The result shows that the false-positive rate is less than 5% under the following conditions: (1) vehicle's average speed is 80km/h, (2) RSU's transmission range is150m, and (3) the distance between RSUs is 20km.

Finally, we analyzed the performance of our approaches in terms of packet size, processing time, transmission delay, and transmission accuracy for different traffic situations. The simulation results show that our schemes are practical and suitable for large scale of VANET applications. For the future work, we plan to expand our techniques suitable for large scale of more complex VANET environments.

ACKNOWLEDGEMENTS

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (No. 2012-0004160).

REFERENCES

[1] T. Abdurahmonov, E. Yeoh and H. M. Hussain, "Improving Smart Card Security using Elliptic Curve Cryptography over Prime Field (F_p), Studies in Computational Intelligence, Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, Springer, vol. 368, pp. 127-140, 2011.

[2] F. Armknecht, A. Festag, D. Westhoff, and K. Zeng, "Cross-Layer Privacy Enhancement and Non-repudiation in Vehicular Communication," Proc. of Workshop on Mobile Ad-hoc Networks (WMAN), pp. 1-12, February-March 2007.

[3] B. Aslam, S. Park, C. Zou, and D. Turgut, "Secure Traffic Data Propagation in Vehicular Ad hoc Networks," International Journal of Ad Hoc and Ubiquitous Computing (IJAHUC), 6(1):24–39, 2010.

[4] C.-N.E. Anagnostopoulos, I.E. Anagnostopoulos, I.D. Psoroulas, V. Loumos, E. Kayafas, " License Plate Recognition From Still Images and Video Sequences: A Survey," IEEE Trans. on Intelligent Transportation Systems, 9(3): 377-391, 2008.

[5] D.P. Bertsekas, J.N. Tsitsiklis. Introduction to Probability. Athena Scientific; 2nd edition, 2008.

[6] J. Blum and A. Eskandarian, "The Threat of Intelligent Collisions," IT Professional, 6(1):24–29, 2004.

[7] G. Calandriello, P. Papadimitratos, A. Lloy, and J.-P.Hubaux, "Efficient and Robust Pseudonymous Authentication in VANET," Proc. of the Workshop on Vehicular Ad Hoc Networks (VANET), 2007.

[8] E. Coronado and S. Cherkaoui, "An AAA Study for Service Provisioning in Vehicular Networks", Proc. of IEEE LCN, pp. 669-676, October 2007.

[9] M. Demirbas and Y. Song, "An RSSI-based Scheme for Sybil Attack Detection in Wireless Sensor Networks," Proc. of International Symposium on World of Wireless, Mobile and Multimedia Networks, pp. 564 – 570, 2006.

[10] J. Douceur, "The Sybil Attack," Proc. of International Workshop on Peer-to-Peer Systems, pp. 251–260, 2002.

[11] G. Guette and C. Bryce, "Using TPMs to Secure Vehicular Ad-Hoc Networks (VANETs)," Proc. of WISTP, pp. 106-116, 2008.

[12] G. Guette and B. Ducourthial, "On the Sybil attack detection in VANET," Proc. of IEEE International Conference on Mobile Ad hoc and Sensor Systems (MASS), pp. 1-6, October 2007.

[13] J.-P. Hubaux, S. Capkun, and J. Luo, "The Security and Privacy of Smart Vehicles," IEEE Security & Privacy magazine, 2(3):49–55, 2004.

[14] G. Karagiannis, O. Altintas, E. Ekici, G. Heijenk, B. Jarupan, K. Lin, and T. Weil, "Vehicular Networking: A Survey and Tutorial on Requirements, Architectures, Challenges, Standards and Solutions," IEEE Communications Surveys & Tutorials, Fourth Quarter, 13(4): 584-616, 2011.

[15] X. Lin and H.-H. Chen, ", A secure and efficient RSU-aided bundle forwarding protocol for vehicular delay tolerant networks," Special Issue on Network Security and Digital Forensics in Next Generation Communications of Wiley's Wireless Communications and Mobile Computing Journal, 11(2):187-195, February 2011.

[16] S. Lv, X. Wang, X Zhao and X Zhou, "Detecting the Sybil Attack Cooperatively in Wireless Sensor Networks," Proc. of International Conference on Computational Intelligence and Security (CIS), pp. 442-446, 2008.

[17] J. Newsome, E. Shi, D. Song, and A. Perrig, "The Sybil Attack in Sensor Networks: Analysis & Defenses." Proc. of International Symposium on Information Processing in Sensor Networks, pp 259–268, 2004.

[18] S. Pal, A.K. Mukhopadhyay and P.P. Bhattacharya, "Defending Mechanisms Against Sybil Attack in Next Generation Mobile Ad Hoc Networks," IETE Technical Review, 25(4): 209-214, 2008.

[19] S. Park, B. Aslam, D. Turgut, C. C. Zou. "Defense against Sybil Attack in Vehicular Ad-Hoc Network based On Roadside Units Support", Proc. of IEEE Military Communications Conference (MILCOM), pp. 1-7, October 2009.

[20] S. Park, C. Zou, and D. Turgut, "Reliable Traffic Information Propagation in Vehicular Ad hoc Networks," In R. Beyah, J. McNair, and and C. Corbett, editors, Security in Ad-hoc and Sensor Networks, pp. 261–291, World Scientific Press, 2009.

[21] M. Raya and J.-P. Hubaux, "The Security of Vehicular Ad Hoc Networks," Proc. of the ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN), pp. 11-21, November 2005.

[22] M. Raya and JP. Hubaux, "Securing Vehicular Ad Hoc Networks," Journal of Computer Security, Special Issue on Security of Ad Hoc and Sensor Networks, 15(1): 39–68, 2007.

[23] M. Raya, P. Papadimitratos, and JP. Hubaux, "Securing Vehicular Communications," IEEE Wireless Communications Magazine, Special Issue on Inter-Vehicular Communications, 13(5): 8–15, 2006.

[24] M. Riley, K. Akkaya and K. Fong, "A survey of authentication schemes for vehicular ad hoc networks," Wiley's Security and Communication Networks Journal, 4(10): 1137-1152, 2011.

[25] K. Sampigethaya, L. Huang, M. Li, R. Poovendran,K. Matsuura, and K. Sezaki, "CARAVAN: Providing Location Privacy for VANET," Proc. of Embedded Security in Cars (ESCAR), Nov. 2005

[26] A. Studer, E. Shi, F. Bai and A. Perrig, "Tacking Together Efficient Authentication, Revocation and Privacy in VANETs," Proc. of SECON pp. 1-9, June 2009.

[27] B. Xiao, B. Yu and C Gao, "Detection and Localization of Sybil Nodes in VANETs," Proc. of Workshop on Dependability issues in Wireless Ad hoc Networks and Sensor networks, pp. 1-8, 2006.

[28] X. Xue and J. Ding, "LPA: a new location-based privacy-preserving authentication protocol in VANET," Special Issue on Focus on Security and Privacy in Emerging Information Technologies of Wiley's Security and Communication Networks Journal, 5(1): 69–78, January 2012.

[29] T. Zhou, R.R. Choudhury, P. Ning and K. Chakrabarty, "Privacy-Preserving Detection of Sybil Attacks in Vehicular Ad Hoc Networks," Proc. of International Conference on MobiQuitous, pp. 1-8, 2007.

TABLE 1. NOTATIONS

Notation	Description
R_i	The *i*-th RSU encountered by the vehicle
KCA^-, KCA^+	A private and public key pair of the CA
KR_i^-, KR_i^+	A private and public key pair of R_i
KV_i^-, KV_i^+	The *i*-th private and public key pair generated by a vehicle
KS_i	The *i*-th random symmetric session key of a vehicle
$K(M)$	An encryption algorithm about a message M with a key K. Both public key encryption and symmetric key encryption are available according to the key type.
$H()$	A cryptographic one-way hash function
$Sig(K^-,M)$	A digital signature for a message M with a private key K^-. Defined as $Sig(K^-,M) = K^-(H(M))$
$Enc(K^+, KS, M)$	A hybrid encryption for a message M with a public key K^+ and a session key KS. Defined as $Enc(K^+, KS, M) = \{K^+(KS), KS(M)\}$
TS_i	A timestamp created by R_i
ATS_i	An aggregated timestamp created by R_i
$Cert\_R_i$	A certificate for the public key of R_i, which is issued by the CA
$Cert\_T_i$	A timestamp certificate issued by R_i
$Cert_i$	A temporary certificate issued by R_i
Content	The basic content of each certificate
TM	A traffic message
Data	Traffic data created by each vehicle

TABLE 2. SIMULATION ENVIRONMENTS

Item	Values
Simulation distance	500m
The number of vehicles	10, 30, 50, 100, 200
Average driving speed	10km/h, 30km/h, 70km/h, 100km/h
Signature algorithm	ECDSA with a key of 163 bits
Asymmetric Encryption	ECC with a key of 163 bits
Hash algorithm	SHA-1
Transmission range	150m
Bandwidth	2MB
The length of timestamp	64bits
Packet size	107bytes, 226bytes

TABLE 3. COMPARISON OF PACKET SIZE AND PROCESSING TIME

Items	Approaches	Timestamp Certificate	Temporary Certificate
Packet Size	Certificate	107 bytes	114bytes
	Request	226bytes	162bytes
Processing Time	Certificate	112.92ms	114.23ms
	Request	36.82ms	39.47ms