DEFENSE AGAINST SYBIL ATTACK IN VEHICULAR AD HOC NETWORK BASED ON ROADSIDE UNIT SUPPORT

Soyoung Park, Baber Aslam, Damla Turgut, Cliff C. Zou
School of Electrical Engineering and Computer Science
University of Central Florida
4000 Central Florida Blvd.
Orlando, FL 32816-2362

ABSTRACT

In this paper, we propose a timestamp series approach to defend against Sybil attack in a vehicular ad hoc network (VANET) based on roadside unit support. The proposed approach targets the initial deployment stage of VANET when basic roadside unit (RSU) support infrastructure is available and a small fraction of vehicles have network communication capability. Unlike previously proposed schemes that require a dedicated vehicular public key infrastructure to certify individual vehicles, in our approach RSUs are the only components issuing the certificates. Due to the differences of moving dynamics among vehicles, it is rare to have two vehicles passing by multiple RSUs at exactly the same time. By exploiting this spatial and temporal correlation between vehicles and RSUs, two messages will be treated as Sybil attack issued by one vehicle if they have the similar timestamp series issued by RSUs. The timestamp series approach needs neither vehicular-based public-key infrastructure nor Internet accessible RSUs, which makes it an economical solution suitable for the initial stage of VANET.

1. INTRODUCTION

With the significant development of network technologies and facilities, vehicular ad hoc network (VANET) has been recently taken a growing interest as a promising technology in a ubiquitous environment. The VANET makes it possible that vehicles sense their local traffic situation and then share the traffic information quickly with each other. This means vehicles can obtain certain traffic information occurred on their driving route earlier to react against accidental events in advance.

Due to the safety requirements of VANET related applications, we have to deal with the security issues associated at the initial development stage of VANET. Among various security issues, in this paper, we focus on Sybil attack because it is the root cause of many security problems. Sybil attack was first introduced by Douceur in the context of peer-to-peer networks [4]. It allows a malicious sender to create multiple fake identities (called Sybil nodes) to impersonate as normal nodes. Most VANET based applications, such as cooperative forward collision warning, pre-crash sensing and warning, local hazard notification, enhanced route guidance and navigation, need the cooperation of vehicles: the similar view sensed by multiple distinct vehicles for a certain traffic situation can provide trustable correctness and a reliable proof about the traffic situation to other vehicles [2][5][6][9][10][12][13]. For this reason, Sybil attack is particularly harmful because it violates the fundamental assumptions of the VANET research.

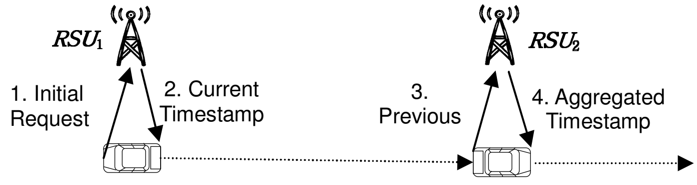

Figure 1. Illustration of the timestamp series approach

In this paper, we present a simple and economical approach called "timestamp series" to defend against the Sybil attack. The proposed work is especially suitable for the initial deployment stage of VANET when basic roadside unit (RSU) support infrastructure is available and a small fraction of vehicles have network communication capability. It uses digital certificates, but avoids using vehicular public key infrastructure (VPKI) for individual vehicles—the authors believe that although VPKI is secure and sound, it is not a realistic requirement for the initial deployment stage of VANET.

The basic idea of the proposed approach is that vehicles obtain certified timestamps signed by RSUs whenever they pass by an RSU (see Figure 1). A traffic message sent out by a vehicle has to contain a series (two or more) of most recently obtained timestamp certificates to show when it passes the last several RSUs. Our technique exploits the spatial and temporal correlation between vehicles and RSUs. Due to the differences of moving dynamics among vehicles, it is rare to have two vehicles passing by multiple RSUs at exactly the same time. Based on this phenomenon, Sybil attack can be detected when a recipient vehicle receives multiple

978-1-4244-5239-2/09/$26.00 ©2009 IEEE

messages with very similar timestamp series.

The remainder of this paper is organized as follows. Section 2 reviews the related work. Our system model, assumptions, and goals are presented in Section 3. We describe our approach in Section 4. Finally, we conclude in Section 5.

2. RELATED WORK

Many algorithms on detecting and defending the Sybil attack have been proposed in various ways in different networking areas. The first approach makes use of resource testing [4][8] based on computing ability, storage ability and communication bandwidth, and so on. It broadcasts a request which requires a certain amount of resource consumption to respond and then only accepts identities that replied within a given time interval. Since smart vehicles are essentially assumed well 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 VANET. J. Newsome et al. [8] 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 the pre-deployment of the shared information among vehicles, the approach is not suitable for VANET.

Douceur [4] has proven that trusted certification is the only approach that can fully eliminate the Sybil attacks. As the most common solution for defending the Sybil attack, numerous techniques based on public key infrastructure (PKI) have been proposed. As the vehicle can be authenticated with its unique public key and certificate managed by the CA, the Sybil attack can be detected at all times.

The traditional PKI-based certificate includes only key information but not the corresponding vehicle's unique physical information. This makes it potentially vulnerable to impersonating attack because any stolen valid key pair and certificate can be used by another malicious vehicle. In multifactor authentication scheme [9], the certificate contains not only the public key information but also a set of physical attribute values of a vehicle, such as radio frequency fingerprint and transmitter coverage, and so on, recorded by the CA. Nevertheless, to establish the public key infrastructure for individual vehicles (VPKI) [10][11] takes a long time. Besides, a centralized key management and certificate authorization could not be realistic in VANET environment where numerous vehicles with different manufacturers, legal policies, or countries coexist at the same time. Most of all, the key distribution and certificate management including issuing, storing and revocation, and so on could be the main barrier to develop the VPKI. In addition, the use of a long-term key pairs and certificate can make tracking and collecting vehicles behaviors easier.

Zhou et al. [13] 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 pseudonym belonging to a vehicle is hashed to a unique value, vehicles cannot abuse those pseudonyms for the Sybil attack. The roadside boxes and the DMV are connected such that any suspicious pseudonyms can be detected through this 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 [5] 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, and the platform credentials are trusted by car manufacturers; therefore, the communications between TPMs of the vehicles are protected from the Sybil attack. However, as the TPM is an improved variation of a certificate, it still needs trusted authorities that can take the responsibility of managing individual vehicles.

Another well known approach to detect the Sybil attack without a big setup is to take use of received signal strength (RSS) [3][7][12] to detect if multiple messages with different identities are sent out by one physical device. Guette et al. [6] analyzed the effectiveness of the Sybil attack in various assumptions of transmission signal tuning and antenna then showed the limitation of RSS based Sybil detection in VANET.

3. SYSTEM MODEL AND GOAL

3.1 ASSUMPTIONS

We consider the initial deployment stage of VANET where (1) only a small fraction of vehicles on roads are equipped with wireless communication devices (we call them smart vehicles); (2) there exists no dedicated vehicular public key infrastructure (VPKI) for individual smart vehicles; (3) limited number of roadside units are set up and they may not have Internet access. In such a basic networking environment, individual vehicle does not have a permanent (or long-term) private/public key pair and certificate. The basic assumptions on vehicles and RSUs are as follows:

- **Vehicle:** It has an on-board unit (OBU) for networking and computing messages, GPS for location detection, and digital map including geographical road information.

- **Economical Roadside Unit (RSU):** It has a transmitter for sending and receiving a message via single hop, but it is not required to have Internet access. In addition, 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 their Certificate Authority (CA). The key pair and certificate are stored in its tamper-proof device.
- **Certificate Authority (CA):** It manages RSUs and issues certificates for individual RSU's public key. Every smart vehicle has pre-installed with the public key of the CA.

Now, we define a malicious vehicle for our model as follows: a malicious vehicle can (i) collect any information spread over the network and (ii) make use of its own manufactured communication device for creating forge GPS information, fake traffic information and any related authentication information such as digital signature.

3.2 SECURITY GOALS

Under the attack model, our protocol provides security against Sybil attacks and provides driver's privacy protection.

- **Prevention of Sybil attack**: Any receiver node can detect the Sybil attack of a malicious vehicle if any.
- **Driver's privacy protection**: It is difficult to track or trace of vehicle's movement from the traffic messages with a limited number of data collection and tracing devices.

4. TIMESTAMP SERIES-BASED DATA PROPAGATION

In this section, we first present a basic scheme suitable for simple roadway architecture such as highways. Then we address a couple of limitations and challenges for extending the approach to urban environments where the roadways have complex topology with many stop traffic signals, intersections and obstacles. Finally, we provide architecture able to solve these challenges.

4.1 BASIC SCHEME

On simple structured roadways that have multiple lanes and have no traffic congestions, vehicles move dynamically at different speeds and move *independently*. Based on this phenomenon, we discover that it would be rare for arbitrary two vehicles to pass through a few different RSUs (far apart from each other) always at the same time. Therefore, if a traffic message sent out by any vehicle contains several timestamps issued to this vehicle by the previously passed RSUs, Sybil attack can be detected if multiple traffic messages contain very similar series of timestamps. These messages can be highly suspected as Sybil messages created by a single vehicle.

This approach requires that only RSUs can issue timestamps and a vehicle cannot use a timestamp obtained by others. Therefore, in our design, (1) each timestamp is digitally signed by the issuing RSU and (2) a timestamp obtained by a vehicle contains this vehicle's self-generated public key, which cannot be used by others who do not know the corresponding private key.

A vehicle may create multiple requests to obtain multiple timestamps from a single RSU. However, multiple timestamps obtained by a single vehicle in a single transmission range of an RSU must be very close in their timestamps. As aforementioned, traffic messages with these timestamps can be easily detected as Sybil messages. We will discuss it in detail in Section 4.1.3.

4.1.1 TIMESTAMP UPDATE

In order to use the dynamic trajectories of vehicles to detect Sybil attack, every traffic message sent out by a vehicle should include at least two timestamps issued from the last two RSUs that the vehicle has passed by. A straightforward way is to attach the two recently obtained timestamps to a traffic message.

However, appending two certified timestamps can increase the size of the traffic messages. Since each timestamp is digitally signed by the issuing RSU, a traffic message must additionally include the certificates of the issuing RSUs as well. In addition, in a traffic congestion, more than two timestamps may be required to put in a traffic message in order to differentiate different vehicles' trajectories; this is because many vehicles move slowly side by side and could receive similar timestamps from RSUs located around the congested area.

To solve the above issue, we propose using an aggregated timestamp in order to minimize the security overhead. To achieve this, a vehicle needs to show its previous timestamp before it obtains a new timestamp. An RSU needs to first verify the given timestamp, and then, for a valid timestamp, creates a new aggregated timestamp that contains both the current and the previous timestamps. Hence, each traffic message needs only the newest aggregated timestamp and a single certificate of the issuing RSU for the Sybil attack detection. Table 1 summarizes all notations used in this paper.

Table 1. Notations

Notation	Description
(KCA^-, KCA^+)	A private and public key pair of the CA
(KR_i^-, KR_i^+)	A private and public key pair of the *i*-th RSU that a vehicle meets
(KV_i^-, KV_i^+)	The *i*-th private and public key pair generated by a vehicle
TS_i	A timestamp created by the *i*-th RSU that a vehicle meets
ATS_i	An aggregated timestamp of a vehicle created by the *i*-th RSU that a vehicle meets
$Cert\_R_i$	A certificate for the public key of the *i*-th RSU that a vehicle meets, which is issued by the CA
$Cert\_T_i$	A certificate for the timestamp, which is issued by the *i*-th RSU that a vehicle meets
$K(M)$	An encryption on 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 such as MD5 or SHA-1
$Sig(K^-,M)$	A digital signature for a message *M* with a private key K^-, i.e., $Sig(K^-,M) = K^-(H(M))$
Content	The basic content that each certificate contains
TM	A traffic message
Data	Traffic data created by each vehicle

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

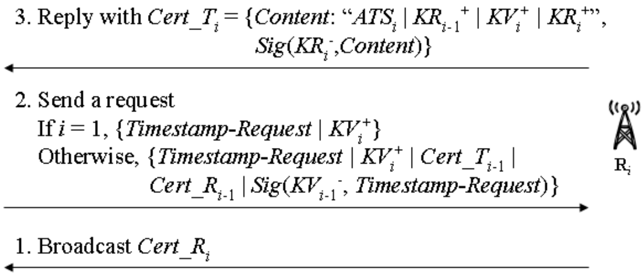
3. Reply with $Cert\_T_i = \{Content$: "$ATS_i \mid KR_{i-1}^+ \mid KV_i^+ \mid KR_i^+$", $Sig(KR_i^-,Content)\}$
2. Send a request
If $i = 1$, $\{Timestamp\text{-}Request \mid KV_i^+\}$
Otherwise, $\{Timestamp\text{-}Request \mid KV_i^+ \mid Cert\_T_{i-1} \mid Cert\_R_{i-1} \mid Sig(KV_{i-1}^-, Timestamp\text{-}Request)\}$
V
R_i
1. Broadcast $Cert\_R_i$

Figure 2. Timestamp update protocol

1. R_i periodically broadcasts its public key in the form of its certificate $Cert\_R_i = \{KR_i^+ \mid location, Sig(KCA^-, KR_i^+ \mid location)\}$ issued by the CA. Every vehicle within a single transmission range of R_i can receive R_i's certificate, and then verify the correctness of the given public key and the certificate based on the CA's public key, which is hard-coded into each vehicle's communication device.
2. After R_i's public key is verified, *V* randomly generates 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 $\{Timestamp\text{-}Request \mid KV_i^+ \mid Cert\_T_{i\text{-}1} \mid Cert\_R_{i\text{-}1} \mid Sig(KV_{i\text{-}1}^-, Timestamp\text{-}Request)\}$. The private-public key pair is used to prevent any malicious vehicle from eavesdropping and stealing *V*'s timestamp. The previous certificate $Cert\_T_{i\text{-}1}$ is required to show the previous timestamp and the issuing RSU. $Cert\_R_{i\text{-}1}$ is required for R_i to verify the correctness of $Cert\_T_{i\text{-}1}$. The signature is a proof that the vehicle is the owner of the certificate $Cert\_T_{i\text{-}1}$.
3. For the given request, R_i first checks if the given certificate $Cert\_T_{i\text{-}1}$ is issued by one of its adjacent RSUs, and verifies the validity of the certificate with $Cert\_R_{i\text{-}1}$ and the CA's public key. If invalid, R_i will not give response to the request. If valid, R_i first extracts previous timestamp information "$TS_{i\text{-}1} \mid TS_{i\text{-}2} \mid$…" from $Cert\_T_{i\text{-}1}$. Then R_i generates an aggregated timestamp ATS_i by concatenating TS_i to the extracted timestamps. Finally, a new certificate $Cert\_T_i = \{Content$: "$ATS_i \mid KR_{i\text{-}1}^+ \mid KV_i^+ \mid KR_i^+$", $Sig(KR_i^-, Content)$, $Cert\_R_i\}$ is broadcasted. Consequently, the new certificate shows a series of the most recent timestamps that V has obtained.

The only responsibility of an RSU is to provide non-malleable timestamp. The RSU does not have any responsibility for managing individual vehicle's public keys and does not need to have Internet access. Thus, a vehicle *V* may be able to create multiple timestamp requests with multiple generated keys, and an RSU will unconditionally create certified timestamps on any valid requests. However, this will not affect our Sybil attack detection because our approach relies on vehicular trajectory not the validity of certificates for the Sybil attack detection.

4.1.2 AGGREGATION OF TIMESTAMPS

The aggregation of timestamps is to concatenate a new timestamp to previous timestamps. In a traffic-congested situation, many vehicles will receive similar timestamps from the RSUs around the jam area. In this case, the aggregation may need a series of more than two timestamps given from consecutive RSUs in order to differentiate each vehicle. Thus, each RSU needs to decide the minimum number of timestamps for the aggregation.

We propose a rule for the aggregation. Let an RSU be *R*, and each RSU adjacent to *R* be NR_j for $j \geq 1$. If *R* receives a request, it classifies the request by the issuing RSU. When *R* receives a request, it 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.

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

- *R* finds the first dissimilar values in the two series (the threshold of similarity is defined by the Sybil attack detection procedure. *R* extracts a series of timestamps from the first value to the first dissimilar value of the given timestamp. 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 $\mathrm{ATS}_{n+1} = <\mathrm{TS}_{n+1} \mid \mathrm{TS}_{n,}>$. Or, if $< TS_n, TS_j >$ and $< TS_{n-1}, TS_{j-1}>$ are very similar but $< TS_{n-2}, TS_{j-2} >$ differ from each other, a new aggregated timestamp is $ATS_{n+1} = <TS_{n+1} \mid TS_n \mid TS_{n-1} \mid TS_{n-2}>$.
- If *R* does not find any dissimilar values in the two series, *R* keeps the whole series of timestamps of the given certificate. A new aggregated timestamps is generated by concatenating TS_{n+1} to the whole series, i.e., $ATS_{n+1} = <TS_{n+1} \mid TS_n \mid TS_{n-1} \mid TS_{n-2} \mid \ldots>$.

4.1.3 USE OF TIMESTAMP SERIES FOR SYBIL ATTACK PREVENTION

Vehicle creates and broadcasts its traffic message about "*Data*" that a vehicle senses periodically or occasionally, according to the data type. "*Data*" may include traffic events, GPS information, moving direction, speed, time, and so on. After passing the *i*-th RSU, for $i \geq 2$, a current certificate of the vehicle 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 message contains *Cert_T$_i$* to prove that the vehicle really passed through the particular RSU and obtained a valid certificate/timestamp from the RSU. 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 the certificate itself with the given RSU's certificate *Cert_R$_i$*. According to the data type or applications, the traffic message can be propagated through multi 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 propagating message.

For *i*=1, the certificate contains 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.

Let arbitrary two traffic messages 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 these two messages are Sybil attack if 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,
- $|TS_i - TS_j| < \varepsilon$ and $|TS_{i-1} - TS_{j-1}| < \varepsilon$

The Sybil detection exploits that, in a normal traffic situation except the traffic jam, the probability of arbitrary two vehicles passing by the same two or more RSUs at the same time is very low. This approach has the advantage that no additional initialization step is needed to obtain the certified timestamps, and no restrictions on requesting timestamps as well. Besides, RSUs are only required to provide simple functionalities, which make them realistic and economical to be used during the initial VANET deployment stage.

If there is traffic congestion, as discussed in Section 4.1.2, the aggregated timestamp obtained by each vehicle will have more than two timestamp values. In this case, the Sybil attack detection procedures is still valid except extending the check of two timestamps to the check of all timestamps contained in *Cert_T$_i$* and *Cert_T$_j$*.

4.2 CHALLENGES

The basic scheme cannot be applied directly to urban environment with a very complex roadway architecture, many signals and intersections, and so on. We briefly review two main challenges which make the basic scheme unavailable in the urban environment.

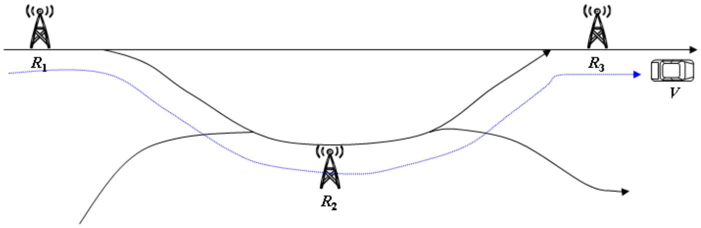

Figure 3. Complex roadway that a vehicle *V* passes through R_1 and R_2 can obtain at least two different certificates from RSU R_3 based on the certificate from R_1 and the certificate from R_2 respectively, since both R_1 and R_2 are adjacent to R_3

First, lets discuss the complex roadways. Even though it is impossible for a single vehicle to drive multiple roads concurrently, a malicious vehicle may exploit the complex roadways for introducing the Sybil attack. Figure 3 shows such 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 timestamps obtained from R_1 and R_2. V can make two distinct timestamp requests based on $Cert\_T_1$ and $Cert\_T_2$, respectively. From the view of R_3, both R_1 and R_2 are adjacent to R_3 so R_3 should issue timestamp $Cert\_T_{31}$ based on $Cert\_T_1$ and $Cert\_T_{32}$ based on $Cert\_T_2$. Therefore, V can produce two Sybil messages by using both certificates $Cert\_T_{31}$ and $Cert\_T_{32}$. For this reason, we need a careful deployment of RSUs in order to prevent such an attack scenario.

Second, lets focus on the frequent stops and slow-down. 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 at intersections. In other words, it may make the Sybil attack easier: since our approach does not prevent a single vehicle from obtaining multiple timestamps from an RSU, a malicious vehicle V can stop nearby an RSU at intersection and gather multiple timestamps that have a long time difference. Consequently, V can send Sybil messages because these timestamps are dissimilar with each other. Therefore, we need to avoid deploying RSUs at the intersections.

4.3 DEPLOYMENT OF RSUs

In this section, we provide a dedicated construction that can solve the aforementioned two challenges. In order to make the approach available at all times in any complex roadway in the real world, we deploy RSUs on roadways with a small restriction. From graph theory point of view, roadways can be expressed as a directed graph where any merging point of distinct roadways including intersections is defined as a vertex, and individual roadway as an edge.

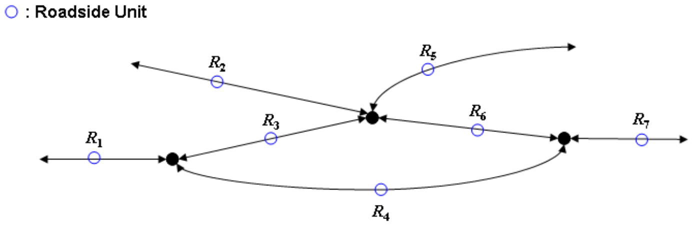

Figure 4. Deployment of RSUs on a roadway graph: RSUs are deployed on every edge of the graph

Our design allows RSUs to locate on every edge as shown in Figure 4. Since different RSUs exist on distinct paths reaching to a certain RSU, the first challenge shown in Figure 3 will be solved. Notice that the digital map of vehicles can inform about the nearest RSU from the vehicle location. If a malicious vehicle attempts to make Sybil message with old certificates not issued by the nearest RSU, it can be easily detected and ignored by every receiver. The deployment will also solve the second challenge as it avoids intersections.

4.4 DRIVER'S PRIVACY PROTECTION

We address the driver's privacy protection under our VANET model. It is a security concern because any spying watcher can attempt to collect, analyze and trace the moving patterns of certain vehicles based on the wireless traffic messages sent out by these vehicles.

We assume that an attacker can obtain any information on the network at any desired place by using his own manufactured devices, but he has a limited number of devices.

The aggregated timestamp includes the previous RSUs' information. Thus, the timestamp reveals information of the trajectory of a vehicle. This will facilitate attackers in tracking a vehicle. However, if the eavesdropped sequence of timestamp chains have any gap, attackers cannot trace a vehicle any more. Because no long-term ID or certificate is used in traffic messages, driver's privacy can be protected.

In order to give a stronger privacy protection, the timestamp updating protocol can be slightly modified. We may simply remove the previous RSU information from the certificate. However, it may have a problem, as shown in Figure 5, when arbitrary two vehicles V_1 and V_2 pass by two different RSUs R_1 and R_2 at the same time and then pass by R_3 again at the same time. In such a case, both V_1 and V_2 have a sequence of very similar timestamps. Without the previous RSU information, traffic messages with those timestamps will be treated as the Sybil messages.

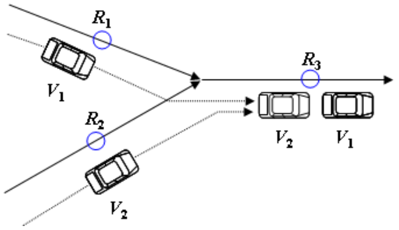

Figure 5. A special situation where two vehicles V_1 and V_2 pass through RSU R_1 and R_2 at the same time, and pass by R_3 at the same time

An alternative way is to use pseudo-ids for previous RSUs. When a vehicle makes a timestamp updating request to R_3, R_3 can create pseudo-ids about its adjacent

RSUs, such as PID_1 for *R*1 and PID_2 for R_2. Then, $Cert\_T_3$ for V_1 includes {*Content*: "ATS_3 | PID_1 | KV_3^+ | KR_3^+", $Sig(KR_3^-$, *Content*), $Cert\_R_3$} instead of KR_1^+, and $Cert\_T_3$' for V_2 includes {*Content*: "ATS_3' | PID_2 | KV_3^+'| 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, because vehicles that pass by the same RSU will obtain certificates with the same pseudo-id. With this design, the certificate shows each vehicle's recent trajectory and corresponding time without revealing the real RSU's information. RSUs can change their pseudo-ids periodically or during the time when there are no requests.

4.5. FAULT TOLERANCE OF RSU FAILURES

The proposed scheme needs at least two certificates issued by two recent adjacent RSUs for the certificate update and the Sybil attack detection. If an RSU is down due to attack or systemic error, vehicles that pass 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 detected a broken RSU R_i gives a notice about the broken RSU to next RSUs according to each vehicle's trajectory. Subsequently, each 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. Then, 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 and admits the fact of 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. Then vehicles can keep obtaining valid certificates and creating valid traffic messages without any delay.

5. CONCLUSION

In this paper, we proposed a practical ways of defending against Sybil attack in a VANET environment, which require neither a dedicated vehicular public key infrastructure for individual vehicles, nor additional setup, but only basic roadside units. Due to the dynamic mobility of vehicles, the Sybil attack can be easily detected if traffic messages have similar timestamps, since the aggregated timestamp shows the most recent trajectory and time of each vehicle. We analyzed our approach for various traffic situations, such as traffic congestion, complex roadways, and so on. We then suggested improved approaches and alternative ways to resolve the challenges posed by these situations.

REFERENCES

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

[2] J. Blum and A. Eskandarian, "The Threat of Intelligent Collisions," IT Professional, vol. 6, no. 1, pp. 24–29, 2004.

[3] 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

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

[5] G. Guette and C. Bryce, "Using TPMs to Secure Vehicular Ad-Hoc Networks (VANETs)," Proc. of WISTP 08, LNCS 5019, pp. 106-116, 2008.

[6] G. Guette and B. Ducourthial, "On the Sybil attack detection in VANET," Proc. of IEEE International Conference on Mobile Adhoc and Sensor Systems, 2007.

[7] 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 '08), pp. 442-446, 2008

[8] 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.

[9] S. Pal, AK. Mukhopadhyay and PP. Bhattacharya, "Defending Mechanisms Against Sybil Attack in Next Generation Mobile Ad Hoc Networks," IETE Technical Review, vol 25, no 4, pp. 209-214, 2008

[10] M. Raya and JP. Hubaux, "Securing Vehicular Ad Hoc Networks," Journal of Computer Security, Special Issue on Security of Ad Hoc and Sensor Networks, vol. 15, no. 1, pp. 39–68, 2007.

[11] M. Raya, P. Papadimitratos, and JP. Hubaux, "Securing Vehicular Communications," IEEE Wireless Communications Magazine, Special Issue on Inter-Vehicular Communications, vol. 13, no. 5, pp. 8–15, 2006.

[12] B. Xiao, Bo Yu and C Gao, "Detection and Localization of Sybil Nodes in VANETs," Proc. of the 2006 workshop on Dependability issues in wireless ad hoc networks and sensor networks, pp. 1-8, 2006

[13] 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 2007, pp. 1-8, 2007