Establishing Trust on VANET Safety Messages*
(Invited Paper)

Subir Biswas[1] and Jelena Mišić[2]

[1] Dept. of Computer Science, University of Manitoba
Winnipeg MB, Canada R3T 2N2
bigstan@cs.umanitoba.ca
[2] Dept. of Computer Science, Ryerson University
Toronto ON, Canada M5B 2K3
jmisic@scs.ryerson.ca

Abstract. We introduce a new scheme for safety message authentication in VANETs. For a practical implementation of VANET, we anticipate that road side units (RSUs) are not physically protected and are prone to several different attacks including node compromise attacks. Thus, an RSU should not be automatically trusted by on road vehicles. In our proposed scheme, a road side controller (RSC) is responsible for controlling all the RSUs, and delivering messages through RSUs to vehicles in a given area, where each RSU uses a proxy signature mechanism based on Elliptic Curve Cryptography (ECC), which is a variation of known ECDS-based proxy signature schemes and modified according to the VANET's criteria and security requirements. The underlying network constraints and properties from VANET standards have been taken into consideration along with the security, reliability and other related issues. We also discuss the potential forgery and attack scenarios on our proposed scheme. The security analysis and simulation results prove the strength and adaptability of our proposed scheme in future VANETs.

Keywords: VANET, ECDSA, proxy signature, replay attack.

1 Introduction

Vehicular Ad hoc Networks (VANETs) open up a new horizon to researchers, developers, and entrepreneurs with several multipurpose applications for driving safety, navigation, and entertainment. A VANET consists of three fundamental components, namely on board unit (OBU), road side unit (RSU), and an appropriate infrastructure to coordinate with the whole system as well as the connectivity to the Internet.

Generally, an RSU is responsible for providing vehicles on road with safety information like traffic collision warning, accident notification, potential rules violation warning, changed road condition etc. It can also be used as an advertisement agent for commercial benefits. On the other hand, an OBU can communicate, exchange messages regarding traffic/road condition, destinations etc. with other vehicles on road. Apart from that, an

* This research has been partially funded by the Canadian Govt's AUTO21 project.

J. Zheng, D. Simplot-Ryl, V.C.M. Leung (Eds.): ADHOCNETS 2010, LNICST 49, pp. 314–327, 2010.

OBU periodically disseminates the vehicle's GPS location data, brake, acceleration information etc. to the other vehicles in its vicinity, and to the closest RSU.

For all these anticipated ideas, trust is a vital aspect for the elements of VANETs since a user (e.g. a driver) will be comfortable with any VANET application when he can reasonably trust the network components, and at the same time, a VANET is useful only when all its entities are trustworthy. If an adversary deliberately broadcasts false messages for traffic safety, or sends an old and expired notification to all the vehicles on road, it may misdirect the entire traffic or even impair the transportation safety. Similarly, if roadside electronic toll collection systems are not trustworthy, it may bring the whole system into a massive disarray.

As required by VANET protocols, RSUs are normally installed on road side locations where they are usually without any good physical protection and enough surveillance. Hence, an RSU is always on the brink of being compromised by an adversary for sending false, expired, and misleading messages that might bring some dangerous consequences for an accepting OBU.

To overcome this risk factor, vehicular communication must have the capability of verifying the identity of the message sender as well as maintaining the integrity of the delivered message, both of which can be accomplished through a suitable signature protocol. However, identity verification in a realtime VANET environment is not a simple task due to high and variable velocity of nodes, varying node density, and the need to operate on roads with nonuniform characteristics. Thus, the issue of scalability is of prime importance as, under heavy traffic, a single controller might need to attend to hundreds, perhaps even thousands of vehicles in a given segment of the transportation network.

Given that future VANETs will be implemented using IEEE 802.11p WAVE (Wireless Access in Vehicular Environments) protocol family [1], it is important to limit the wireless traffic intensity in order to avoid packet collisions at the Medium Access Control (MAC) layer. The presence of many messages from vehicles and RSUs on a particular road may increase the message collision rate and thus impair the performance of on-road vehicular communication. Eichler et al. [2] have shown that the WAVE standard can't deal with many high priority messages in a dense network scenario. Hence, we need a trust scheme that has low computational complexity, high scalability, as well as a reliable and quick verification mechanism.

To cope with the above requirements, we look into the issue of safety message delivery in vehicular networks with a perspective of authenticating the RSU as a valid member of the corresponding RSU group to on road OBUs, and delivering the signed safety messages to OBUs by the RSU on behalf of a road side controller (RSC). As mentioned before, RSUs may deliver several messages including routine messages on road-safety issues, accident notifications, traffic congestion alerts, and commercial messages etc. An RSU may require to keep broadcasting each message repeatedly for a particular time span.

We propose to exploit the features of proxy signature [3] for the RSU message delivery. The term proxy signature refers to a variation of digital signature that designates an entity (called a proxy signer) to sign a message on behalf of the original signer. We

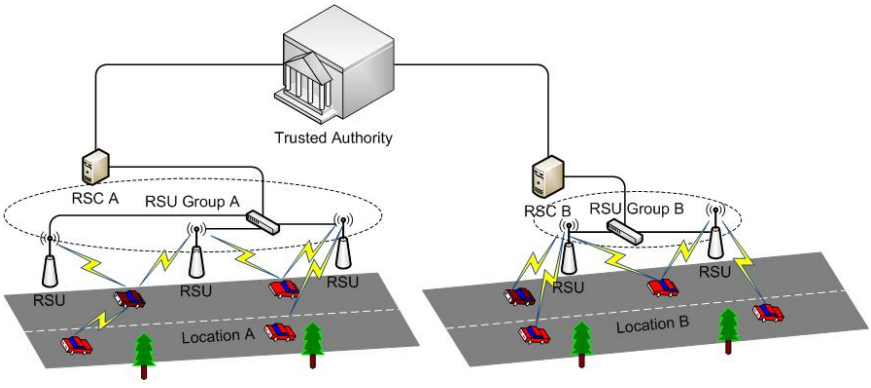

Fig. 1. The framework for trust in safety message delivery in VANETs

considered a number of signature schemes and found the Elliptic Curve Digital Signature Algorithm (ECDSA) [4] most suitable as the original signature protocol for fast and efficient signing of safety messages over the VANET. The current VANET standards for security services [5] are also suggesting the ECDSA scheme for signing the secure messages in VANETs.

In our scheme, RSUs will be the proxy signers signing safety and other application messages to the OBU recipients on behalf of the original creator of the messages- a road side controller (RSC). A recipient of the messages (OBU) can verify the identity of the original signer, and it can also verify the integrity of the contents of the received message.

Our scheme is derived from the modification of some of the ECDSA-based proxy signature protocols which are described in [6,7,8]; using which a corresponding RSU can neither replay an expired message, nor alter an original message. Therefore, the control of the message broadcast is kept with the message originator. A partial delegation mechanism [3] produces a new secret key from the original signer's secret, and the new secret is then used as the key for proxy signing.

We organize the paper in the following manner. Section 2 discusses the related work on VANET authentication and trust management together with evolution of some proxy signature schemes. A brief account of ECDSA preliminaries is given in Section 3. Section 4 illustrates the proposed mechanism for VANET safety message delivery. The security analysis is provided in Section 5, while the simulation results are presented in Section 6. Finally, Section 7 concludes the paper. all subsequent paragraphs are.

2 Related Work

Our study of related work includes the existing VANET message authentication literature, as well as the digital proxy signature approaches which we considered while proposing our scheme.

2.1 On Message Authentication in VANETs

Number of papers have been published in recent years addressing the issue of VANET authentication (e.g. [9], [10], [11], [12] and [13]), where researchers mainly focused on authenticating OBU messages to RSUs and to other OBUs in the light of vehicular anonymity and other VANET requirements while the safety messages from RSUs are mostly taken for granted, and considered trustable.

Lin et al. [14] proposed in 2007, an ID-based signature [15] scheme where the RSU location is used as the public key for the message signature. Each message sent from the RSU contains the physical location information so that once the message is received by an OBU, it can be verified based on the location information. This is how it prevents the RSU-OBU communications from a potential replication attack as messages from the RSU will be discarded if the RSU is dislocated.

In 2009, Studer et al. [16] proposed VANET Authentication using Signatures and TESLA++ (VAST). A combination of ECDSA [4] and a modified TESLA protocol [17] have been used to carry out VANET's requirement of message verification in an efficient way. Upon receiving a message from a VANET entity, the receiver would perform 2 types of verification: 1. TESLA++ verification, and 2. an ECDSA verification is used normally when non-repudiation is necessary. To prevent the major computational and memory-based DoS attacks, the received payload is verified using TESLA++ before the ECDSA verification is performed. If the initial TESLA++ verification fails due to some reasons, CPU utilization, as well as the size of the message queue are taken into consideration before switching to ECDSA mode of authentication. However, RSUs are still assumed to be trusted and the approach doesn't explicitly address the issue of replay attack and false message injection attack by a supposedly legitimate RSU. There is no clear indication on how a VANET should react upon detection of a node compromise. Moreover, Hass et al. in [18] remarked that ECDSA performs better than a TESLA implementation at longer communication distances since TESLA requires a second packet delivery for the message verification purpose.

Wen et al. [19] exploits the spatial and temporal properties of physical layer channel responses for securing each communication pair in VANETs. The basic idea is to distinguish one sending transmitter from another based on the physical layer measurements for a series of messages. The sending entity attaches the authenticating signals which is a unique channel response along with the information payload transmission. A receiving entity can verify the legitimate transmission by comparing the authenticator signal with the estimated channel response. This approach is efficient and scalable for a physical layer approach, but again, comes with some limitations. For instance each vehicle has to be preloaded with public keys of other vehicles to be able to access the network. This will obviously pose a huge task of updating and maintaining a huge number of keys as the size of the VANET grows. Authors also haven't addressed the node compromise attack which can have a deadly consequence in the form of false message broadcast, or replaying expired safety messages.

Among the few other recent work, ASIC scheme [20] introduces a faster and efficient way of aggregated verification of signatures and certificates for VANETs using Bilinear pairing technique. This approach can verify a large number of signatures and certificates

to ensure a reliable and optimized operation of VANETs, nevertheless, establishment of trust on VANET safety messages has not been covered by this mechanism.

2.2 On Proxy Signatures

The original proxy signature scheme, proposed by Mambo et al. [3], was further extended by Kim et al. [21] who proposed two additional features – proxy signature by partial delegation with warrant and the threshold delegation based proxy signature. Further enhancements include blind proxy signature schemes [8,22,23], and [24] by which a proxy signer is made unable to manipulate the message contents (and, replay the expired messages). However, blind proxy signatures are not practical for VANET applications, since they require a new proxy tuple to be generated and delivered to the proxy signer every time there is a new message to be posted. ECDSA-based proxy signatures are comparatively new in research and [6,7,25] are among the most recent publications.

3 Preliminaries

In this section, we give a quick review on the Elliptic Curve facts that would be necessary for understanding the cryptographic materials used in ECDSA. We are going to use an elliptic curve over a finite field for our scheme.

A Finite Field. A finite field $\mathbb{F}_p$ is a finite set of of p elements along with addition and multiplication operations on F. The number of elements is denoted as the order of the finite field. There exists a finite field of order q if and only if q is a prime power, and on the other hand, if q is a prime power, then there exists only one finite field of order q denoted by $\mathbb{F}_q$.

Elliptic Curve. An Elliptic Curve E over a finite field $\mathbb{F}_p$ is defined in the form of the following equation:

$$y^2 = x^3 + ax + b, \tag{1}$$

where a prime $p > 3$; $a, b \in F_p$, and $4a^3 + 27b^2 \not\equiv 0 (mod\ p)$. The set of elements of the Elliptic Curve $E(\mathbb{F}_p)$ consists of the points (x,y) where $x \in \mathbb{F}_p$ and $y \in \mathbb{F}_p$. A point at infinity $\mathcal{O}$ together with the set of points $E(\mathbb{F}_p)$ identifies an elliptic curve. Note that the addition, multiplication, and inversion operations on an elliptic curve points are different from ordinary binary operations. Please refer to [26] for the detailed description of the above mentioned operations.

ECDSA. The domain parameters of Elliptic Curve Digital Signature Algorithm (ECDSA) require an Elliptic Curve E over a finite field of size q, and a base point $G \in (\mathbb{F}_q)$. Value q is chosen as a prime power p^t where p is a prime number, and t is a positive integer. In our case of elliptic curve, $t = 1$, thus $p = q$. Also, as indicated in (1), two field elements a and b are chosen, where, $a, b \in (\mathbb{F}_q)$. All these parameters could be shared by the entities or by some specific user depending upon the ECDSA configuration.

4 ECDSA-Based Proxy Signature for VANETs

The ECDSA-based proxy signature mechanism requires a proxy key which should be preprocessed and generated by the trusted authority. This preprocessing is done at the trusted authority and RSC during the network deployment phase, and should be kept secret from other entities. Being equipped with the proxy key, an RSU can sign on the safety message on behalf of the corresponding RSC using the proxy key and other *session components*. We assume that RSUs are independent of each other while a predefined group of RSUs in a geographical area are working under the same RSC. The RSC (and not the RSU) is physically protected, trusted entity, and responsible for issuing safety messages for OBUs. Then, the safety messages are delivered through individual RSUs in a same RSU group. Fig 1 outlines our network design. An OBU is preloaded with the public key of the trusted authority, which can verify the safety message with the signature components of the received payload. Following are the necessary steps for our scheme.

4.1 Key Material Preprocessing

Consider a large prime q for a finite field $\mathbb{F}_q^*$ and let G be a generator over an Elliptic Curve E_q. Also, x be a random number, generated by the trusted authority (TA), and would be used as the private key. The public key for x, $V = xG$ is calculated and delivered to all concerning entities along with the certificate from the TA. TA generates a unique and distinct random secret k_o where $(1 < k_o < q)$. It then computes:

$$g = k_o G \bmod q \tag{2}$$

For each member RSU_i in VANET, TA generates a random number k_i where $(1 < k_i < q)$.

$$R_i = k_i G \tag{3}$$

$$(x_1, y_1) = k_i g \tag{4}$$

The proxy key for the RSU_i is calculated at TA using the following equation.

$$S_i = (xx_1 + k_i)k_o^{-1} \bmod q \tag{5}$$

This proxy key will be used by the RSU to sign a message payload when issued by the corresponding RSC. Hence, each RSU_i is pre-loaded with the proxy key S_i, along with g, R_i, x_1 at the end of this key material preprocessing phase.

4.2 Payload Preprocessing

Each message M includes a message payload m, and a message expiry information t_x (i.e. $M = m||t_x$). When there is a safety message M to be released, the RSC determines a new session parameter k_p using the hash over k_o, m and t_x. This session parameter is unique for each session of safety message delivery, i.e. a different message session will

have a different k_p value than the current message. We choose SHA-1 (160 bit output blocks) for the hash operation.

$$k_p = h(k_o||m||t_x) \tag{6}$$

The used k_o is the same k_o of eqn.(2). The purpose of generating this session parameter is to prevent the vehicles from the replay attacks as described in the security analysis section. The RSC further calculates:

$$(x_p, y_p) = k_p g \tag{7}$$

Then, the safety message M, session parameter k_p, and x_p are delivered to all the legitimate RSUs through a secure channel. The next step is producing the proxy signature, carried out by individual RSUs operating under the RSC.

4.3 RSU Proxy Signature

The RSU_i receives the tuple $(M||k_p||x_p)$ intact from the corresponding RSC, signs the message content on behalf of the RSC, and keeps broadcasting the signed message contents to the on road vehicles until the message is expired.

$$S_{r,i} = k_p^{-1}(h(M) + S_i x_p) mod\, q \tag{8}$$

The combination $(M||g||R_i||x_1||x_p||S_{r,i})$ is used as the broadcast payload for the safety message and transmitted to all the OBUs in the communication range of the RSU_i. We assume that the length of the prime q is 160 bits. Table 1 gives the account of the lengths of the payload components.

Table 1. Size of the safety message overheads. q is chosen of 160 bits.

Component	Size (in bits)
g	160
R_i	160
x_1	80
x_i	80
$S_{r,i}$	160
$Total$	640 (= 80 Bytes)

Message Verification by OBU. The received contents are utilized in the following verification mechanism.

$$(x_j, y_j) = (h(M)g + x_p(R_i + x_1 Q))S_{r,i}^{-1} mod\, q \tag{9}$$

If $(x_j = x_p)$ the received message is accepted, else rejected. Note that the inverse operations used in 8 and 9 (and all throughout the paper) are modular inverse operations. Refer to [26] for the details on modular inverse operation.

4.4 Enhancement

Table 2 lists the components of time complexity of our scheme. The most expensive operations here are the point multiplication operations since the mechanism of point multiplication involves both ordinary multiplications as well as modular inversion operations. In the above description, workload for signature generation in the RSU is kept low by having no point multiplications in that phase. However, we can transfer a couple of point multiplications from the verification phase to the signature generation phase. This change would be significant as verifying parties are OBUs which need to save on battery power and more importantly, an OBU is attached with a fast moving vehicle.

Table 2. Time complexity of the proposed approach. t_{pt}, t_m, t_{mi}, and t_{hf} are to mean the time taken by a point multiplication, ordinary multiplication, modular inverse, and hashing operations respectively.

Type	Preprocessing	Signature	Verification
Point Multiplication	4 t_{pt}	0 t_{pt}	3 t_{pt}
Multiplication	2 t_m	2 t_m	3 t_m
Modular Inverse	1 t_{mi}	1 t_{mi}	1 t_{mi}
Hash function	1 t_{hf}	1 t_{hf}	0 t_{hf}

As the extension of our scheme, we shift the load of computation from the OBU to the RSU. The verification operation defined for an OBU (eqn. 9) is replaced by the following set of calculations to take place in the RSU:

$$\begin{aligned} u_1 &= h(M)S_{r,i}^{-1} mod\, q \\ u_2 &= x_p S_{r,i}^{-1} mod\, q \\ u_3 &= x_1 u_2 mod\, q \end{aligned} \tag{10}$$

$$d = u_1 g + u_2 R_i \tag{11}$$

An alternative message payload $(M||d||u_3||x_p)$ is then sent to the OBU. If we consider q a 160 bit prime number, d and u_3 will be of 160 bits each while the other component x_i is of 80 bits. The total size of the safety message overhead is only 50 Bytes which is much lighter than the one prior to the extension (80 Bytes). Our scheme overheads just 10 Bytes extra for proxy signature on a safety message compared to the ordinary ECDSA signature (40 Bytes) which is suggested by the VANET standards for message delivery. Therefore, our approach is not much extravagant on communication bandwidth for the safety message broadcast. The verification process at the OBU would now compute:

$$(x_j, y_j) = d + u_3 V \tag{12}$$

Again, if the relation $(x_j = x_p)$ holds, the message is a valid one and therefore accepted by the receiving OBU, else it's discarded. Fig. 2 summarizes the scheme with the help of a flow diagram.

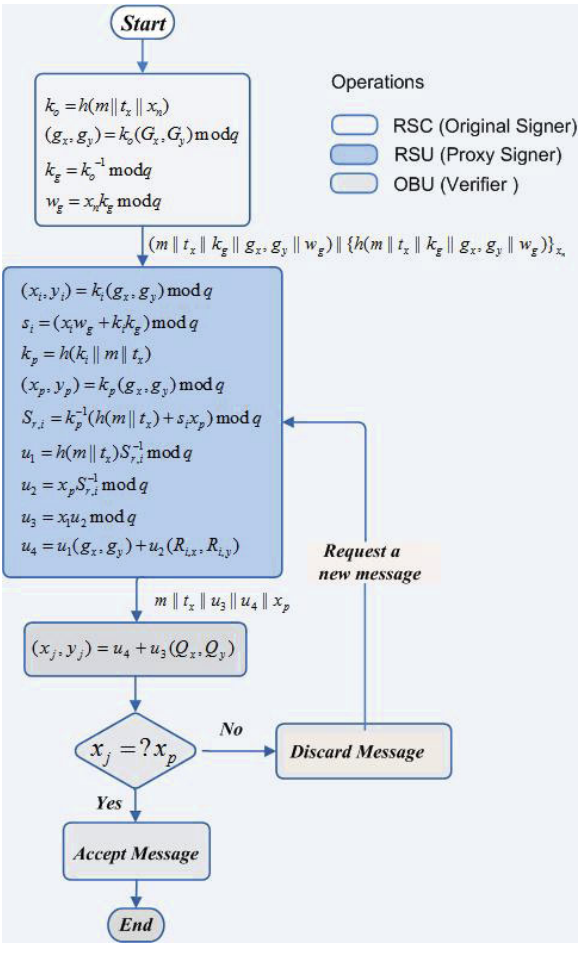

Fig. 2. Flow diagram of the ECDSA-based Proxy Signature for VANETs

5 Security Analysis

The security of our scheme depends mainly on the size of the prime q, and the hardness of solving the elliptic curve problem. We assume here that the RSC is a trusted authority, and secured against any kind of physical compromise, whereas an RSU is not trusted by the on road vehicles. An unprotected RSU can be compromised by an adversary attempting to inject false messages, or modifying the messages in order to harming the traffic system. A replay attack is possible be launched using a valid signature on an expired message. A malicious RSU might attempt to forge other RSU's signature to falsify an innocent RSU by sending a valid signature on a malicious message. This property is referred to as exculpability. Another potential misbehavior is repudiation of signed messages, which refers to a situation when a malicious RSU would deny its

involvement in producing a particular signature which has been actually generated by that RSU. In the light of the above discussion, we derive the following lemmas and proofs on our scheme.

Lemma 1. *An RSU_i can not generate a valid proxy key S_i.*

Proof. In order to generate a valid proxy key S_i, an adversary would require the Trusted Authority's secret key (x), and two other random secrets k_i, k_o as indicated in eqn.(5). The secret x is irreversible from the knowledge of the public key V, since that involves point multiplications of an elliptic curve E_q. The other two random numbers k_i and k_o are generated and stored at the TA. Assuming that both of them are 160 bit random numbers, the joined probability of a successful brute force guessing of k_i and k_o is $1/(2^{160}-1)^2$.

Lemma 2. *An RSU_i can not launch a false message attack and a replay attack.*

Proof. While signing a message payload M on behalf of the RSC, RSU_i uses k_p (in eqn.(6)) which has been derived by hashing the k_o value and the payload M. The value k_o is a secret kept at the TA, and any changes on M would result into a different k_p value for which the signature would be different. Note that the payload M contains the main message m, as well as the message expiry information t_x (i.e. $M = m||t_x$). Therefore, for a modified t_x, k_p would be changed with a consequence of an invalid signature from the RSU_i. If we deploy SHA-1 for hashing in eqn.(6), the probability that an adversary or a malicious RSU_i would be successful in brute forcing the hash function is $1/(2^{160}-1)$.

Lemma 3. *A proxy signature by an RSU_i is non-repudiable and exculpable.*

Proof. The random number k_i is a non-overlapping one which makes a proxy key S_i unique for a particular RSU_i (refer to eqn.(5)). We assume that the number of RSUs working under a given RSC is much less than q. Thus, an adversary can not reproduce the S_i value with an acceptable probability. Hence, a proxy signature for a given message $S_{r,i}$ can only be created by the RSU_i, and therefore, non-repudiable.

RSC stores all individual proxy keys (S_i) along with the corresponding k_i, k_p, and x_p values. If there is any dispute, the RSC can reproduce the signature using the credentials of the disputed RSU. Therefore, an RSU_i can not sign a message that would appear as signed by a different RSU. That preserves the exculpability property.

Note that the RSC can always reproduce d and u_3 for a given payload M. Since reproducing them would require $S_{r,i}$ and R_i, which are private parameters and only accessible to the RSC and RSU_i, we can argue that the enhancement of our protocol is non-repudiable and exculpable.

6 Simulation

We evaluate our scheme for VANET in NS2 simulation over a 8 lane straight highway (4 lanes in either direction) of length 1.5 km. Vehicles with varying traffic densities and speeds are moving in both directions. We assume that all the vehicles in the simulation scenario are always in the communication range of a RSU group. A safety message

is signed using our scheme by an RSU and broadcast to OBUs. We don't simulate RSCs here assuming that the RSC has already delivered the safety message and other credentials for proxy signature using some secure means. Also, we don't simulate the initial wireless association and certificate verification process to keep things simple and straight.

We configure each RSU and OBU with communication range set to 300m while all the devices are equipped with an omni-directional antenna. The Two-Ray Ground propagation model is used on top of the Nakagami model at the physical layer. All other PHY and MAC layer parameters are set as suggested by NS2 $Mac/802\_11Ext$ and $Phy/WirelessPhyExt$ for IEEE802.11p. The bandwidth of the channel is fixed at 6Mbps. The interface queue length for each device is set to 20 for OBUs and RSUs in the simulation.

Each OBU periodically (every 100 ms) sends a heartbeat message that contains its current address, location, velocity information along with the size of the packet, broadcast address, and authentication data. The size of the packet delivered by the RSU is considered 500 Bytes including 50 Bytes of signature overhead from our scheme. An RSU broadcasts the signed safety message periodically (every 100 ms) until the message gets expired.

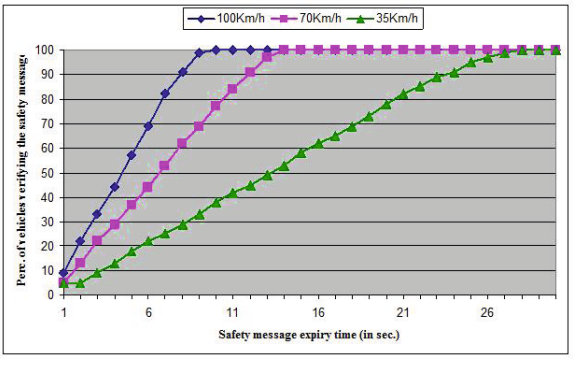

Fig. 3. Percentage of on-road vehicles verifying the safety message for different message expiry time (t_x)

Fig. 3 shows the impact of safety messages' expiry time on the proportion of vehicles that would be able to receive and verify them for a traffic scenario of 200 vehicles moving at three different speed levels. For a faster moving traffic (speeding at least 70Km/h), it takes between 10 to 15 seconds to get the message verified by all the vehicles in the communication range. Slower traffic on the other hand, would require a longer broadcast session in order for an OBU to receive and verify the same message.

Fig. 4 reveals that the percentage of vehicles verifying the signed safety messages is almost constant with the different size of the traffic, while $t_x > 10$ sec enables 100% of vehicles in the communication range, verifying safety messages within the given time period.

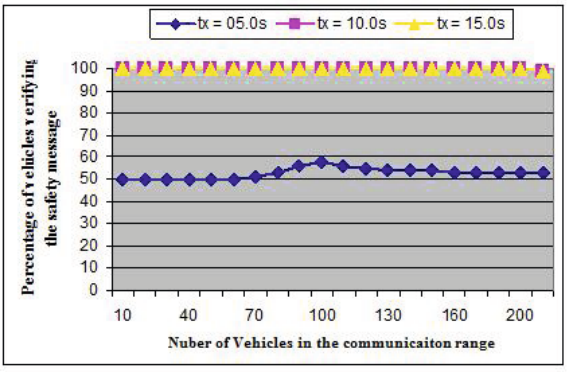

Fig. 4. Percentage of OBUs verifying a safety message for different number of vehicles in the communication range

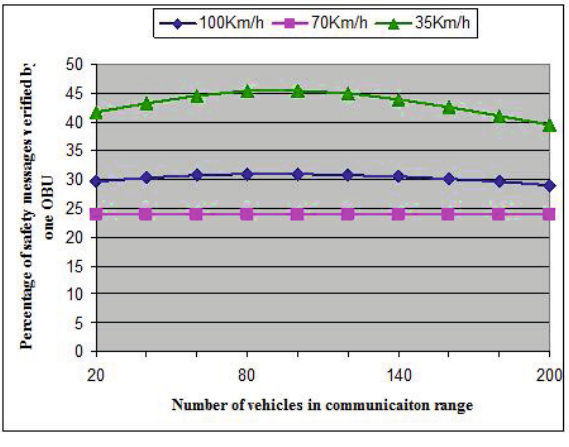

Fig. 5. Percentage of safety messages verified by one OBU for different number of vehicles in the communication range

As indicated in Fig. 4, for $t_x = 10$ to 15, almost all the vehicles receive and verifies a safety message, while for a shorter expiry time, the percentage of vehicles is significantly less.

The proportion of safety messages being verified by one OBU depends on the total number of vehicles in the communication range of the RSU group as indicated in Fig. 5. It shows for three different speed levels of the traffic, only a portion of total delivered safety messages are received and verified by an OBU. Since our scheme considers rebroadcasting of a safety message until its expiry (t_x), an OBU doesn't require to receive and verify all the safety messages in a particular session.

7 Conclusion and Future Direction

We presented a new trust scheme for VANETs' safety message delivery system, using ECDSA-based proxy signature. This mechanism would help a VANET user to authenticate the safety message, as well as the message originator along with the transmitting RSUs. In our scheme, an adversary can not forge an RSU, or a compromised RSU can not broadcast false, altered, and expired safety message. Thus, it makes the VANET safety message more reliable, trustworthy and acceptable to a user. The security analysis proved the strength of the scheme by analyzing the potential attack scenarios, while the simulation results support the usability of the scheme in the real-world VANETs. The approach has low communication overhead, which is compliant to the IEEE802.11p/WAVE standards as it uses the basic ECDSA signature scheme for the proxy signature in RSUs.

Our future work includes study and research on security and trust management for vehicle to vehicle (V2V), and vehicle to infrastructure (V2I) safety message forwarding mechanisms with vehicle privacy. In order to make it energy efficient and bandwidth friendly, our research will have a focus on low power cryptographic primitives. For the experimental evaluation of our scheme, and comparison with its counterparts, we'll be implementing a more realistic and dynamic VANET simulation that'll provide a number of real-world traffic scenarios.

References

1. Draft amendment for wireless access in vehicular environments (WAVE). IEEE, New York, IEEE Draft 802.11p (July 2007)
2. Eichler, S.: Performance evaluation of the ieee 802.11p wave communication standard. In: IEEE 66th Vehicular Technology Conference, VTC-2007, 30 2007-October 3, pp. 2199–2203 (Fall 2007)
3. Mambo, M., Usuda, K., Okamoto, E.: Proxy signatures for delegating signing operation. In: CCS 1996: Proceedings of the 3rd ACM Conference on Computer and Communications Security, pp. 48–57. ACM, New York (1996)
4. Johnson, D.B., Menezes, A.J.: Elliptic curve dsa (ecsda): an enhanced dsa. In: SSYM 1998: Proceedings of the 7th Conference on USENIX Security Symposium, p. 13. USENIX Association, Berkeley (1998)
5. IEEE trial-use standard for wireless access in vehicular environments (wave)- security services for applications and management messages. IEEE, New York (July 2006) IEEE Std 1609.2
6. Chang, M.-H., Chen, I.-T., Chen, M.-T.: Design of proxy signature in ecdsa. In: ISDA 2008: Proceedings of the 2008 Eighth International Conference on Intelligent Systems Design and Applications, pp. 17–22. IEEE Computer Society, Washington (2008)
7. Sun, X., Xia, M.: An improved proxy signature scheme based on elliptic curve cryptography. In: International Conference on Computer and Communications Security, pp. 88–91 (2009)
8. Qi, C., Wang, Y.: An improved proxy blind signature scheme based on factoring and ecdlp. In: International Conference on Computational Intelligence and Software Engineering, CiSE 2009, Wuhan, China, pp. 1–4 (2009)
9. Raya, M., Papadimitratos, P., Hubaux, J.-P.: Securing vehicular communications. Wireless Communications, IEEE 13(5), 8–15 (2006)
10. Guo, J., Baugh, J., Wang, S.: A group signature based secure and privacy-preserving vehicular communication framework, pp. 103–108 (May 2007)

11. Sha, K., Xi, Y., Shi, W., Schwiebert, L., Zhang, T.: Adaptive privacy-preserving authentication in vehicular networks. In: Proceedings of the First International Conference on Communications and Networking in China, ChinaCom 2006, pp. 1–8 (October 2006)
12. Calandriello, G., Papadimitratos, P., Hubaux, J.-P., Lioy, A.: Efficient and robust pseudonymous authentication in vanet. In: VANET 2007: Proceedings of the Fourth ACM International Workshop on Vehicular Ad hoc Networks, pp. 19–28. ACM, New York (2007)
13. Xi, Y., Sha, K., Shi, W., Schwiebert, L., Zhang, T.: Enforcing privacy using symmetric random key-set in vehicular networks, pp. 344–351 (March 2007)
14. Lin, X., Sun, X., Ho, P.-H., Shen, X.: Gsis: A secure and privacy-preserving protocol for vehicular communications. IEEE Transactions on Vehicular Technology 56(6), 3442–3456 (2007)
15. Barreto, P.S.L.M., Libert, B., McCullagh, N., Quisquater, J.-J.: Efficient and provably-secure identity-based signatures and signcryption from bilinear maps. In: Roy, B. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 515–532. Springer, Heidelberg (2005)
16. Studer, A., Bai, F., Bellur, B., Perrig, A.: Flexible, extensible, and efficient vanet authentication. Journal of Communications and Networks 11(6), 574–588 (2009)
17. Perrig, A., Canetti, R., Tygar, J.D., Song, D.: The tesla broadcast authentication protocol. RSA CryptoBytes 5(2), 2–13 (2002)
18. Haas, J.J., Hu, Y.-C., Laberteaux, K.P.: Real-world vanet security protocol performance. In: IEEE GLOBECOM, pp. 1–7 (2009)
19. Wen, H., Ho, P.-H., Gong, G.: A novel framework for message authentication in vehicular communication networks. In: Global Telecommunications Conference, IEEE GLOBECOM 2009, pp. 1–6 (2009)
20. Wasef, A., Shen, X.: ASIC: Aggregate signatures and certificates verification scheme for vehicular networks. Engine (2009), https://129.97.58.88/ojs-2.2/index.php/pptvt/article/view/487
21. Kim, S., Park, S., Won, D.: Proxy signatures, revisited. In: ICICS 1997: Proceedings of the First International Conference on Information and Communication Security, pp. 223–232. Springer, London (1997)
22. Park, J.-H., Kim, Y.-S., Chang, J.H.: A proxy blind signature scheme with proxy revocation. In: International Conference on Computational Intelligence and Security Workshops, pp. 761–764 (2007)
23. Wei-min, L., Zong-kai, Y., Wen-qing, C.: A new id-based proxy blind signature scheme. Wuhan University Journal of Natural Sciences 10(3), 555–558 (2005)
24. Cai, M., Kang, L., Jia, J.: A multiple grade blind proxy signature scheme. In: International Conference on Intelligent Information Hiding and Multimedia Signal Processing, vol. 2, pp. 130–133 (2007)
25. Xue, Q., Li, F., Zhou, Y., Zhang, J., Cao, Z., Qian, H.: An ecdlp-based threshold proxy signature scheme using self-certified public key system. In: MobiSec 2009, pp. 58–70 (2009)
26. Johnson, D., Menezes, A.: The elliptic curve digital signature algorithm (ecdsa). Certicom Research, Canada; and Dept. of Combinatorics and Optimization, University of Waterloo, Canada, Tech. Rep. (1999)