Security and Privacy Issues for Inter-vehicle Communcations in VANETs

Tat Wing Chim, S. M. Yiu and Lucas C. K. Hui
Department of Computer Science
The University of Hong Kong
Email: {twchim, smyiu, hui}@cs.hku.hk

Victor O. K. Li
Department of Electrical and Electronic Engineering
The University of Hong Kong
Email: vli@eee.hku.hk

***Abstract*—Vehicular ad hoc network (VANET) is an emerging type of networks to allow vehicles on roads to communicate for driving safety. An vehicle can broadcast messages (e.g. accident information) to other vehicles. These messages may have impact on other vehicles as well as the traffic control system, so all messages must be signed and authenticated. On the other hand, privacy should be enforced while the real identity of the sender should be traceable by authorized party. In this poster, we first discuss the limitations of existing solutions. In particular, we describe an impersonation attack to one of the schemes, highlight the problem of communications overhead, and effectiveness of the message verification procedure. Then, we present the main ideas of our proposed scheme which can be shown to be secure and more effective than existing schemes.**

***Index Terms*—Secure vehicular sensor network, security, privacy**

I. Setting and Basic Functions of VANETs

A vehicular ad hoc network (VANET) is also known as a vehicular sensor network by which driving safety is enhanced through inter-vehicle communications or communications with roadside infrastructure. It is an important element of the Intelligent Transportation Systems (ITSs) [1].

In a typical VANET, each vehicle is assumed to have an on-board unit (OBU) and there are road-side units (RSU) installed along the roads. A trusted authority (TA) and maybe some other application servers are installed in the backend. The OBUs and RSUs communicate using the Dedicated Short Range Communications (DSRC) protocol [2] over the wireless channel while the RSUs, TA, and the application servers communicate using a secure fixed network (e.g. the Internet).

The basic function of an VANET is to allow arbitrary vehicles to broadcast safety messages (e.g. road condition, traffic accident information) to other nearby vehicles and RSU such that other vehicles may adjust their travelling routes and RSU may inform the traffic control center to adjust traffic lights for avoiding possible traffic congestion.

II. Security and Privacy Issues

Like other communication networks, security issues have to be well-addressed. In particular, a good scheme must satisfy the following requirements:

1) Message integrity and authentication: A vehicle should be able to verify that a message is indeed sent and signed by another vehicle without being modified by anyone.
2) Identity privacy preserving: The real identity of a vehicle should be kept anonymous from other vehicles and a third-party should not be able to reveal a vehicle's real identity by analysing multiple messages sent by it.
3) Traceability: Although a vehicle's real identity should be hidden from other vehicles, if necessary, the TA should have the ability to obtain a vehicle's real identity.

III. Problems of Existing Schemes

For authentication, digital signature in conventional public key infrastructure (PKI) [3] is a well accepted approach. However, for VANETs, requiring a vehicle to verify the signatures of other vehicles may not be practical. The computation power of an OBU is not strong enough to handle all verifications in a short time. Also, for messages from an unknown vehicle, the public key certificate attached induces heavy message overhead. So, a well-accepted alternative is to let the nearby RSU to verify all messages.

Along this direction, [4] proposed the IBV protocol for vehicle-to-RSU communications (it can be easily extended for vehicle-to-vehicle communications). They also provided a batch verification process to verify a large number of signatures as a batch using three *pairing* [5] operations. However, their protocol suffers from an impersonation attack which we will explain in Section IV. Also, a vehicle's real identity can be traced by anyone, thus violating the privacy requirement. On the other hand, in their batch verification scheme, if any of the signatures in the batch is erroneous, the whole batch will be dropped. This is inefficient because most signatures in the batch may actually be valid and can be used, thus may imply a not satisfactory success rate.

In [6], the RAISE protocol was proposed for vehicle-to-vehicle communications. The protocol allows a vehicle to verify the signature of another with the aid of a nearby RSU. However, no batch verification can be done and the RSU has to verify signatures one after another. To notify other vehicles which messages are valid, hash values of individual messages need to be broadcasted. As there can be tens up to thousands of signatures within a short period of time, the notification messages may induce a heavy message overhead.

978-1-4244-3938-6/09/$25.00 (c)2009 IEEE

IV. IBV Protocol and Impersonation Attack

In this section, we describe the IBV protocol and present an impersonation attack (a vehicle can send messages on behalf of another) to the protocol.

The IBV Protocol: Before network deployment, the TA sets up the parameters using the following steps: 1) It chooses $\mathbb{G}$ and $\mathbb{G}_{\mathbb{T}}$ that satisfy the bilinear map properties. 2) It randomly picks $s_1, s_2 \in \mathbb{Z}_q$ as its master keys. These two master keys are preloaded into each vehicle's tamper-proof hardware device. 3) It computes $P_{pub_1} = s_1 P$ and $P_{pub_2} = s_2 P$ as its public keys. The parameters $\{\mathbb{G}, \mathbb{G}_{\mathbb{T}}, q, P, P_{pub_1}, P_{pub_2}\}$ are then preloaded into all RSUs and OBUs. 4) It assigns each vehicle a real identity $RID \in \mathbb{G}$ and a password PWD. The drivers are informed about them during network deployment or during vehicle first registration.

When a vehicle starts up, the driver first inputs its RID and PWD into the tamper-proof device. If they are valid, the tamper-proof device starts its role in generating pseudo identities, secret keys and message signing. Vehicle V_i's pseudo identity is generated as $ID_i = (ID_{i1}, ID_{i2})$ where $ID_{i1} = rP$ and $ID_{i2} = RID_i \oplus H(rP_{pub1})$ and r is a per-session random nonce. Its secret key is then generated as $SK_i = (SK_{i1}, SK_{i2})$ where $SK_{i1} = s_1 ID_{i1}$ and $SK_{i2} = s_2 H(ID_{i1}||ID_{i2})$. Here $H(.)$ is a MapToPoint hash function. When vehicle V_i wants to send the message M_i, it generates the signature $\sigma_i = SK_{i1} + h(M_i)SK_{i2}$ where $h(.)$ is a one-way hash function such as SHA-1. V_i then broadcasts ID_i, M_i and σ_i to the RSU.

The RSU verifies the signature σ_i by checking whether $\hat{e}(\sigma_i, P) = \hat{e}(ID_{i1}, P_{pub1})\hat{e}(h(M_i)H(ID_{i1}||ID_{i2}), P_{pub2})$.

Having the pseudo identity ID_i of vehicle V_i, the TA can trace its real identity by using the *TA RID Tracing Routine*: $ID_{i2} \oplus H(s_1 ID_{i1}) = RID_i \oplus H(rP_{pub_1}) \oplus H(s_1 rP) = RID_i$.

Impersonation Attack: Assume that at a certain instance, vehicle V_i with real identity RID_i generates its pseudo identity $ID_i = (ID_{i1}, ID_{i2})$, secret keys SK_i and signs message M_i by generating the signature σ_i as usual. While V_i is transmitting, an attacker V_a records ID_i. After some while, V_a generates the message M_a. It generates its pseudo identity as $ID_a = (ID_{a1}, ID_{a2}) = ID_i = (ID_{i1}, ID_{i2})$ and its secret keys as $SK_a = (SK_{a1}, SK_{a2})$ where $SK_{a1} = s_1 ID_{a1} = s_1 ID_{i1}$ and $SK_{a2} = s_2 H(ID_{a1}||ID_{a2}) = s_2 H(ID_{i1}||ID_{i2})$. It then signs the message M_a by generating the signature $\sigma_a = SK_{a1} + h(M_a)SK_{a2}$ and sends out ID_a, M_a and σ_a to the RSU.

Upon receiving V_a's message, the RSU can verify it successfully because $\hat{e}(\sigma_a, P) = \hat{e}(ID_{i1}, P_{pub1})\hat{e}(h(M_a)H(ID_{i1}||ID_{i2}), P_{pub2})$. Assume at a later time, V_a's message M_a causes an accident on the road. The RSU forwards V_a's pseudo identity ID_a as shown in its message to the TA and wants to reveal its real identity. After computing $ID_{a2} \oplus H(s_1 ID_{a1}) = ID_{i2} \oplus H(s_1 ID_{i1}) = RID_i \oplus H(rP_{pub_1}) \oplus H(s_1 rP) = RID_i$, both the RSU and the TA think that M_a is being sent by V_i because V_i's instead of V_a's identity is traced. Thus V_a can escape from and pass its guilty of causing the accident to V_i. In fact, the protocol also suffers from a few other security problems such as privacy violation as any other vehicle can reveal the real identity of others.

V. Our Proposed Scheme

Due to space limitation, we only list the core ideas in our scheme. For details, please refer to [7]. Our scheme is also based on bilinear map.

Provision of security and privacy: The use of pseudo identity can help to hide the sender's real identity, thus protecting the privacy of the sender. The drawback of IBV protocol is that the pseudo identity is linked to the real identity and is supposed to be generated by the real signer. However, once the pseudo identity is known, every vehicle can produce a corresponding valid signing key. Thus, the attack is successful.

To resolve this problem, we propose to use a shared secret m_i between the vehicle and the RSU while keeping the idea of using pseudo identity to protect privacy. So, the pseudo identity is still generated based on the real identity. But then, the signing key has to be generated based on the shared secret m_i, thus even if the attacker can get hold of the pseudo identity of a vehicle, there is no way for the attacker to generate a valid signing key to sign a message.

We show the basic scheme as follows. A vehicle V_i first authenticate itself based on its real identity RID_i and password PWD_i using the conventional public key infrastructure to the TA via the closest RSU. TA authenticates V_i and generates the shared secret m_i for RSU and V_i. TA securely forwards m_i, $H(RID_i)$ and an encrypted block containing m_i and s (the system secret) for V_i to RSU. $H(.)$ is a MapToPoint hash function and the encrypted block can only be decrypted by V_i. RSU, in turn, passes that encrypted block to V_i. This basically completes the handshaking. A new shared secret between RSU and V_i will be generated when V_i meets a new RSU.

To sign a message M_i, V_i first generates a random nonce r and creates its pseudo identity ID_i as $(ID_{i1}, ID_{i2}) = (rP_{pub}, H(RID_i) \oplus H(m_i ID_{i1}))$. The signing key SK_i is $(SK_{i1}, SK_{i2}) = (sm_i ID_{i1}, sH(ID_{i2}))$. It then generates the signature as $\sigma_i = SK_{i1} + h(M_i)SK_{i2}$ where $h(.)$ is a one-way hash function such as SHA-1. Since m_i is known by V_i, RSU and TA only, no other vehicle can impersonate V_i. Note that to avoid other attacks and allow efficient batch verification of signatures, we need to enhance this basic scheme using another shared secret between the TA and a vehicle. The details can be found in [7].

Batch verification by RSU: To batch-verify the signatures, we rely on the property that $\hat{e}(\sum_{i=1}^{n} \sigma_i, P) = \hat{e}(\sum_{i=1}^{n} m_i ID_{i1} + h(M_i)H(ID_{i2}), P_{pub})$. Compared to the batch verification process proposed by the IBV protocol, ours uses one fewer pairing operation. Recall that in IBV protocol, the whole batch will be dropped if there is an invalid signature. To avoid this, we adopt binary search to extract valid signatures. We divide a batch into two halves and then perform checking on each half.

If the half still contains invalid signatures, repeat the process, otherwise notifies all signatures in the half. The searching process stops when a pre-defined level of binary search is reached or all valid ones are found.

Efficient notification by RSU: After the RSU verifies vehicle V_i's signature σ_i, it notifies all vehicles within its range the result. Instead of including the hash value of each message in the notification message, we store the hash values in bloom filters (Please refer to [8] about what a bloom filter is).

If vehicle V_i wants to know if a message it received is valid or not, it first computes the hash value of the message and checks if the hash value is in the bloom filter (signed by RSU). However, bloom filter is well-known to have false positives. To resolve this problem, we use two bloom filters with the additional bloom filter (negative filter, the other filter is referred as positive filter) for storing the hash values of those invalid signatures. In other words, if the hash value of the message is found in the positive filter, but not in the negative filter, we are sure that the message is valid. On the other hand, if the hash value is found in the negative filter, but not the positive filter, we are sure that the message is invalid. For the other unresolved cases, we use a a reconfirmation procedure to check the validity of the message. Based on our simulation, the number of these cases is very few as long as the parameters for the bloom filters (with respect to the number of hash values to be stored in the filter) are set appropriately.

VI. Simulation Results

We assume that vehicles passes through an RSU (in highway) at speeds varying from 50 km/h to 70 km/h. The RVC and the IVC ranges are set to 600m and 300m respectively. Inter-vehicle messages are sent every 500 ms from each vehicle. IEEE 802.11a is used to simulate the medium access control layer. The bandwidth of the channel is 6 Mb/s and the average length of inter-vehicle message is 200 bytes. We compute the transmission time based on the bandwidth and the length of the message. The RSU performs batch verification every 300 ms and each pairing operation takes 4.5 ms. We implement the simulation using C++ language.

The simulation runs for 1000 s. We vary the inter-vehicle message signature error rate from 1% to 10% to interpret its impact on the performance of our schemes. For each configuration, we compute the average of 5 different random scenario. To measure the successful rate, we only consider the batch with invalid signatures (invalid batch). We extend the definition of the successful rate (only on invalid batches) in [6] as $IBSR = \frac{1}{N}\sum_{i=1}^{N}\frac{M_{app}^i}{M_{mac}^i}$, where M_{app}^i is the total number of messages that are successfully verfied by the RSU and are consumed by vehicle V_i in the application layer before V_i leaves RSU's RVC range.

The success rate and delay performance of our scheme are shown in Figure 1 and Figure 2 respectively. In the figures, SPECS (BS0) is our scheme without using binary search in batch verification while SPECS (BS2) is our scheme using 2 levels of binary search in batch verification. It can be seen that with binary search, a large portion of valid signatures in invalid batches can still be used while the delay overhead induced is only marginal. Note that in Figure 1, the success rate of IBV and SPECS(BS0) are the same as both will drop the whole batch if there is an invalid signature inside. In Figure 2, we remark that our SPECS(BS0) is slightly better than IBV as we improve the batch verification process by using one fewer pairing operation.

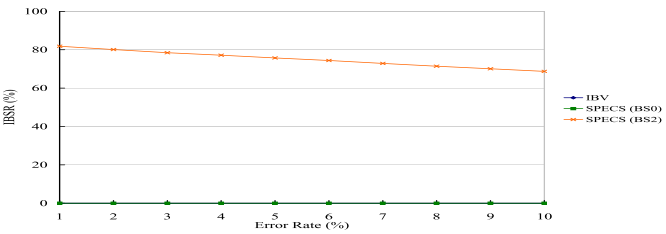

Fig. 1. Invalid Batch Success Rate vs. Error Rate

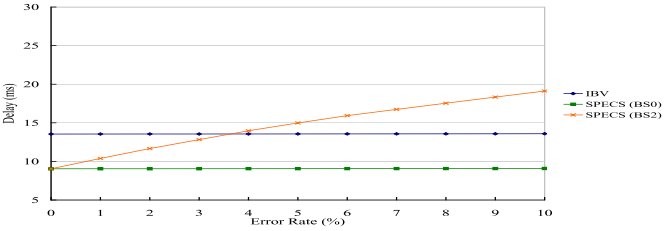

Fig. 2. Delay vs. Error Rate

VII. Conclusions

In this poster, we discuss the security and privacy concerns for vehicle-to-vehicle communications in VANETs. We highlight the problems of existing solutions and present the core ideas of our proposed scheme. We show that our scheme is more effective than existing schemes based on simulation. The security analysis of our scheme is given in the full paper.

References

[1] F. Wang, D. Zeng, and L. Yang, "Smart Cars on Smart Roads: an IEEE Intelligent Transportation Systems Society Update," *IEEE Pervasive Computing, Vol. 5, No. 4*, pp. 68 – 69, 2006.

[2] H. Oh, C. Yae, D. Ahn, and H. Cho, "5.8 GHz DSRC Packet Communication System for ITS Services," in *IEEE Proceedings of the VTC '99*, Sept. 1999, pp. 2223 – 2227.

[3] R. Housley, W. Ford, W. Polk, and D. Solo, "Internet X.509 Public Key Infrastructure Certificate and CRL Profile," *IETF RFC2459*, 1999.

[4] C. Zhang, R. Lu, X. Lin, P. H. Ho, and X. Shen, "An Efficient Identity-based Batch Verification Scheme for Vehicular Sensor Networks," in *IEEE Proceedings of the INFOCOM '08*, Apr. 2008, pp. 816 – 824.

[5] A. Menezes, "An Introduction to Pairing-Based Cryptography," in *1991 Mathematics Subject Classification, Primary 94A60*, 1991.

[6] C. Zhang, X. Lin, R. Lu, and P. H. Ho, "RAISE: An Efficient RSU-aided Message Authentication Scheme in Vehicular Communication Networks," in *IEEE Proceedings of the ICC '08*, May 2008, pp. 1451 – 1457.

[7] T. W. Chim, S. M. Yiu, L. C. K. Hui, and V. O. K Li, "SPECS: Secure and Privacy Enhancing Communications for VANET," 2009, manuscript.

[8] Y. Dong, T. W. Chim, Victor O. K. Li, S. M. Yiu, and Lucas C. K. Hui, "ARMR: Anonymous Routing protocol with Multiple Routes for Communications in Mobile Ad Hoc Networks," *Ad Hoc Networks (to appear)*, 2009.